ABSTRACT

Title of dissertation: Extending the Scope of Provable Adversarial Robustness in Machine Learning

Aounon Kumar
Doctor of Philosophy, 2023

Dissertation directed by: Professor Soheil Feizi
Department of Computer Science

The study of provable defenses against adversarial attacks in machine learning has mostly been limited to classification tasks and static one-step adversaries. Robustness certificates are designed with a fixed adversarial budget for each input instance and with the assumption that inputs are sampled independently. The goal of this work is to expand the set of provable robustness techniques to cover more general real-world settings such as adaptive multi-step adversaries (e.g., reinforcement learning), distribution shifts (e.g., color shifts in images) and models with structured outputs (e.g., images, sets, and segmentation masks). Each setting presents unique challenges which require special proof techniques designed specifically to tackle them. For instance, an adversarial attack on a reinforcement learning agent at a given time step can affect its performance in future time steps. Thus, certified robustness methods developed for the static setting cannot provide guarantees in a dynamic environment that evolves over time. Similarly, tasks like image segmentation and text generation cannot be modeled as a classification problem as their outputs cannot be treated as discrete class labels in a meaningful way.

First, we present a robustness certificate for bounded Wasserstein shifts of the input distribution. We show that a simple procedure that randomizes the input of the model within a transformation space is provably robust to distributional shifts under that transformation. Our framework allows the datum-specific perturbation size to vary across different points in the input distribution and is general enough to include fixed-sized perturbations as well. Our certificates produce guaranteed lower bounds on the performance of the model for any (natural or adversarial) shift of the input distribution within a Wasserstein ball around the original distribution. We apply our technique to (i) certify robustness against natural (non-adversarial) transformations of images such as color shifts, hue shifts and changes in brightness and saturation, (ii) certify robustness against adversarial shifts of the input distribution, and (iii) show provable lower bounds (hardness results) on the performance of models trained on so-called "unlearnable" datasets that have been poisoned to interfere with model training.

Next, we present certifiable robustness in the setting of reinforcement learning where the adversary is allowed to track the states, actions and observations generated in previous time-steps and adapt its attack. We prove robustness guarantees for an agent following a Gaussian-smoothed policy. The goal here is to certify that the expected total reward obtained by the robust policy remains above a certain threshold under a norm-bounded adaptive adversary. Our main theoretical contribution is to prove an adaptive version of the Neyman-Pearson Lemma – a key lemma for smoothing-based certificates – where the adversarial perturbation at a particular time-step is allowed to be a stochastic function of previous observations, states and actions. Our approach differs from existing techniques as it can generate certificates for an entire episode instead of certifying predictions

at individual time-steps.

We then develop a randomized smoothing-based algorithm to produce certifiably robust models for problems with structured outputs. Many machine learning problems like image segmentation, object detection, image/audio-to-text systems, etc., fall under this category. Our procedure works by evaluating the base model on a collection of noisy versions of the input point and aggregating the predictions by computing the center of the smallest ball that covers at least half of the output points. It can produce robustness certificates under a wide range of similarity (or distance) metrics in the output space such as perceptual distance, intersection over union and cosine distance. These certificates guarantee that the change in the output as measured by the distance metric remains bounded for an adversarial perturbation of the input.

We also study some limitations of randomized smoothing when used to defend against ℓ_p-norm bounded adversaries for $p > 2$, especially for $p = \infty$. We show that this technique suffers from the curse of dimensionality when the smoothing distribution is independent and identical in each input dimension. The size of the certificates decreases with an increase in the dimensionality of the input space. Thus, for high-dimensional inputs such as images, randomized smoothing does not yield meaningful certificates against an ℓ_∞-norm bounded adversary.

We also design a method to certify confidence scores for neural network predictions under adversarial perturbations of the input. Conventional classification networks with a softmax layer output a confidence score that can be interpreted as the degree of certainty the network has about the class label. In applications like credit scoring and disease diagnosis systems where reliability is key, it is important to know how sure a model is

about its predictions so that a human expert can take over if the model's confidence is low. Our procedure uses the distribution of the confidence scores under randomized smoothing to generate stronger certificates than a naive approach that ignores the distributional information.

Finally, we present a certifiable defense for streaming models. In many deep learning applications such as online content recommendation and stock market analysis, models use historical data to make predictions. Robustness certificates based on the assumption of independent input samples are not directly applicable in such scenarios. We study provable robustness of machine learning models in the context of data streams, where inputs are presented as a sequence of potentially correlated items. We derive robustness certificates for models that use a fixed-size sliding window over the input stream. Our guarantees hold for the average model performance across the entire stream and are independent of stream size, making them suitable for large data streams.

Extending the Scope of Provable Adversarial Robustness in Machine Learning

by

Aounon Kumar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:

Professor Soheil Feizi, Chair and Advisor
Professor Tom Goldstein, Co-Advisor
Professor John P Dickerson
Professor Ming C. Lin
Professor Behtash Babadi

Acknowledgments

I wish to extend my profound gratitude to my advisor, Professor Soheil Feizi. His support and mentorship have been instrumental in the successful completion of this dissertation. His availability for counsel and guidance throughout this journey has been invaluable. It has been a great pleasure to work with and learn from such an extraordinary individual.

I would also like to thank my co-advisor, Professor Tom Goldstein. His insightful feedback and guidance over the course of my studies have been instrumental.

I would also like to thank my colleagues Alexander Levine and Vinu Sankar Sadasivan for their invaluable collaborations. Alex has made significant contributions to chapters 2, 3, 5 and 6, and Vinu has contributed to chapter 7.

Table of Contents

Chapter 1: Introduction

1.1 Adversarial Attacks

Machine learning (ML) models, especially deep neural networks (DNN), are prone to attacks where an adversary adds a tiny perturbation to the input and completely alters the prediction of the model [1, 2, 3, 4]. Such attacks can significantly degrade the performance of a model, like an image classifier, and make it output any class label of the attacker's choice. Apart from static tasks like classification, adversarial attacks also exist for dynamic tasks like reinforcement learning where the adversary is capable of adapting its strategy based on the observations to become more effective [5, 6, 7, 8]. Such attacks also exist for models with more complex outputs than class labels, such as image captioning [9], speech-to-text systems [10], image reconstruction [11, 12, 13, 14], generative models [15], super-resolution [16, 17], etc. Such widespread presence of adversarial attacks is concerning as it poses serious risks in the use of deep neural networks for safety-critical applications, such as autonomous vehicles and medical diagnosis, where robustness and reliability are of utmost importance.

1.2 Provable Robustness

Over time, several heuristic approaches have been proposed to detect and defend against adversarial attacks [18, 19, 20, 21, 22, 23]. Unfortunately, such defenses have been broken by stronger attacks [4, 24, 25, 26]. A defense that achieves good empirical performance against a particular attack might remain vulnerable to newer and stronger attacks, making it difficult to determine the true robustness of a model. This necessitates the study of provable adversarial robustness which seeks to design ML models with mathematically verifiable guarantees on their performance against adversarial attacks. Provable defenses are of special importance in the study of adversarial robustness as their robustness guarantees continue to hold regardless of improvements in attack strategies. Several provable defenses have been proposed in the literature, with the notable ones being based on convex-relaxation [27, 28, 29, 30, 31], interval-bound propagation [32, 33, 34, 35], and randomized smoothing [36, 37, 38, 39]. Out of these techniques, randomized smoothing has been shown to scale up to high-dimensional inputs like ImageNet images [40].

The existing literature on provable robustness focuses mostly on static machine-learning tasks like classification. However, practical machine-learning applications frequently diverge from the canonical classification setting. Machine learning models are often expected to operate in dynamic and adaptive environments, like in robotics and self-driving. They may produce structured outputs such as images, segmentation masks, and language, which are difficult to model as discrete class labels in a meaningful way. Furthermore, models also encounter distribution shifts when deployed in the real world

which can significantly deteriorate their performance [41, 42, 43, 44, 45, 46]. Our objective is to design provable robustness techniques for real-world machine-learning scenarios such as those mentioned above. Each setting presents unique challenges which require us to develop innovative proof techniques for addressing them. For instance, the robustness certificates designed for the static setting of classification cannot be used against a dynamic and adaptive adversary in reinforcement learning.

1.3 Randomized Smoothing

Among the notable provable defenses, randomized smoothing scales up to high-dimensional inputs like images. Given an input image, this procedure evaluates a classifier on several noisy versions of the image. The class label that gets predicted with the highest probability is returned as the output of the robust model [36, 37, 38, 39]. Cohen et al. [36] proved tight certificates for the robust model under the ℓ_2 adversarial threat model when smoothed using the Gaussian distribution.

If the most-likely class gets predicted with probability p under Gaussian perturbations of variance σ^2, then the output of the robust model is guaranteed to remain unchanged within an ℓ_2 ball of the following radius around the input image:

$$\epsilon = \sigma \Phi^{-1}(p),$$

where Φ is the cumulative distribution function of the standard normal distribution. Since the probability p cannot be computed exactly for most conventional neural network-based classifiers, a high-confidence (say 99.9%) lower-bound $\underline{p}$ is obtained using a large number of samples of the smoothing distribution.

1.4 Outline

The following chapters study provable robustness in various different settings. Chapter 2 presents robustness certificates against input distribution shifts. The goal is to certify the accuracy of a model under bounded Wasserstein shifts of the input distribution. The contents of this chapter have been published in [47].

Chapter 3 studies provable robustness in reinforcement learning. The goal here is to certify the total reward obtained by an agent under an adversarial attack. This work has been published in [48].

Chapter 4 presents robustness certificates for models with structured outputs like images and segmentation masks. The goal is to guarantee that the changes in the output are small, as measured by a distance or similarity metric in the output space, for small perturbations in the input of bounded ℓ_2 length. This work has been published in [49].

Chapter 5 studies the limitations of randomized smoothing-based approaches for certifying the ℓ_∞-treat model. The best possible ℓ_∞ certificates decrease rapidly with the dimensionality of the problem. The contents of this chapter have been published in [50].

Chapter 6 presents certified guarantees on the confidence of a neural network in its predictions. It uses the cumulative distribution function of the confidence scores under the smoothing distribution to obtain better certificates. This work has been published in [51].

Chapter 7 studies provable robustness in the context of data streams. The goal is to certify the average performance of a model on a sequence of potentially correlated input items. The robustness certificate is independent of the stream size and is applicable to potentially infinite streams. This work has been published in [52].

Chapter 2: Robustness to Distribution Shifts

2.1 Introduction

Machine learning models often suffer significant performance loss under minor shifts in the data distribution that do not affect a human's ability to perform the same task– e.g., input noise [53, 54], image scaling, shifting and translation [41], spatial [42] and geometric transformations [43, 44], blurring [45, 46], acoustic corruptions [55] and adversarial perturbations [1, 2, 3, 24, 56]. Overcoming such robustness challenges is a major hurdle for deploying these models in safety-critical applications where reliability is paramount. Several techniques have been developed to improve the empirical robustness of a model to data shifts, e.g., diversifying datasets [57], training with natural corruptions [58], data augmentations [59], contrastive learning [60, 61, 62] and adversarial training [2, 3, 63, 64, 65]. Empirical robustness techniques are designed to protect a model against a particular type of shift or adversary (e.g., by introducing similar shifts during training) and may not be effective against new ones. For instance, adversarial defenses have been shown to break down under newer attacks [4, 24, 25, 26, 66].

Certifiable robustness methods, on the other hand, seek to produce provable guarantees on the robustness of a model which hold for any perturbation within a certain neighborhood of the input instance regardless of the strategy used to generate this perturbation. A

robustness certificate produces a verifiable lower bound on the size of the perturbation required to fool a model. Apart from being a guarantee on the robust performance, these certificates may also serve as a metric to compare the robustness of different models that is independent of the mechanism producing the input perturbations. However, the study of provable robustness has mostly focused on perturbations with a fixed size budget (e.g., an ℓ_p-ball of same size) for all input points [27, 28, 29, 31, 32, 33, 36, 37, 38, 39, 67, 68, 69]. Among provable robustness methods, randomized smoothing based procedures have been able to successfully scale up to high-dimensional problems [36, 37, 38, 39] and adapted effectively to other domains such as reinforcement learning [70, 71] and models with structured outputs [49] as in segmentation tasks and generative modeling. However, these techniques cannot be extended to certify under distribution shifts as the perturbation size for each instance in the input distribution need not have a fixed bound. For example, stochastic changes in the input images of a vision model caused by lighting and weather conditions may vary across time and location. Even adversarial attacks may choose to adjust the perturbation size depending on the input instance.

A standard way of describing a distribution shift is to constrain the Wasserstein distance between the original distribution $\mathcal{D}$ and the shifted distribution $\tilde{\mathcal{D}}$ to be bounded by a certain amount ϵ, i.e., $W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \epsilon$, for an appropriate distance function d. The Wasserstein distance is the minimum expectation of the distance function d over all possible joint distributions with marginals $\mathcal{D}$ and $\tilde{\mathcal{D}}$. Wasserstein distance is a standard similarity measure for probability distributions and has been extensively used to study distribution shifts [72, 73, 74, 75]. Certifiable robustness against Wasserstein shifts is an interesting problem to study in its own right and a useful tool to have in the arsenal of

provable robustness techniques in machine learning.

In this work, we design robustness certificates for distribution shifts bounded by a Wasserstein distance of ϵ. We show that by simply randomizing the input in a transformation space, it is possible to bound the difference between the accuracy of the robust model under the original distribution $\mathcal{D}$ and the shifted distribution $\tilde{\mathcal{D}}$ as a function of their Wasserstein distance ϵ under that transformation. Given a base model μ, we define a robust model $\bar{\mu}$ which replaces the input of μ with a randomized version sampled from a "smoothing" distribution around the original input. Let $\bar{h}$ be a function denoting the performance of the robust model $\bar{\mu}$ on an input-output pair (x, y) (see Section 2.3 for a formal definition). Then, our main theoretical result in Theorem 1 shows that

$$\left|\mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)]\right| \leq \psi(\epsilon),$$

where ψ is a concave function that bounds the total variation between the smoothing distributions at two input points as a function of the distance between them (condition (2.3) in Section 2.3). Such an upper bound always exists for any smoothing distribution as the total variation remains between zero and one as the distance between the two distributions increases. We discuss how to find the appropriate ψ for different smoothing distributions in Appendix G.

We apply our result to certify model performance for families of parameterized distribution shifts which include shifts in the RBG color balance of an image, the hue/saturation balance, the brightness/contrast, and more. Our method does not make any assumptions on the model and applies to both natural and adversarial shifts of the distribution. It does

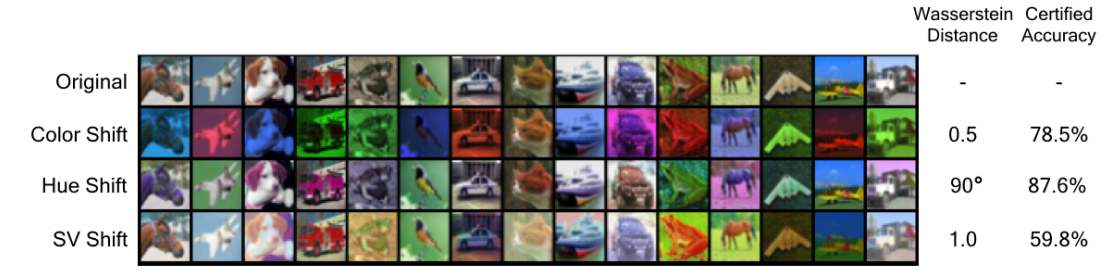


Figure 2.1: Certified accuracies obtained for different natural transformations of CIFAR-10 images such as color shifts, hue shifts and changes in brightness and saturation. The Wasserstein distance of each distribution shift from the original distribution is defined with respect to the corresponding distance function.

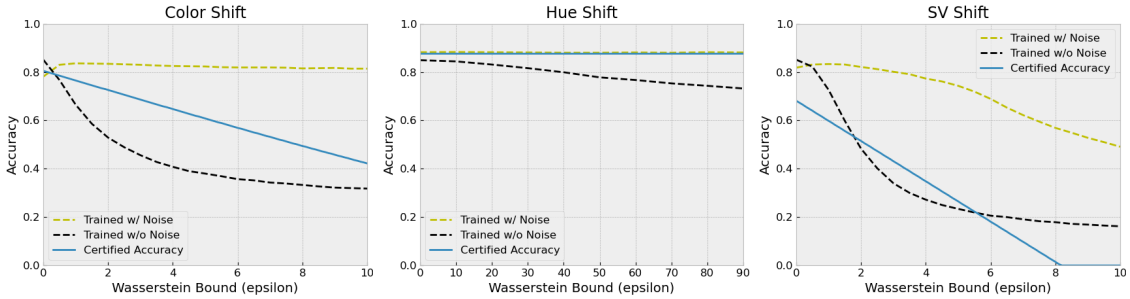


Figure 2.2: Comparison between the empirical performance (dashed lines) of two base models (trained on CIFAR-10 images with and without noise in transformation space) and the certified accuracy (solid line) of a robust model (noise-trained model smoothed using input randomization) under distribution shifts. The certified accuracy often outperforms the undefended model and remains reasonably close (almost overlaps for hue shift) to the model trained under noise for small shifts in the distribution.

not increase the computational requirements of the base model as it only samples one randomized input per robust prediction, making it scalable to high-dimensional problems that require conventional deep neural network architectures. The sample complexity for generating the Wasserstein certificates over the entire distribution is roughly the same as obtaining adversarial certificates for a single input instance using existing randomized smoothing based techniques [36, 39].

Robustness under distribution shifts is a fundamental problem in several areas of machine learning and our certificates could be applicable to a multitude of learning tasks. We demonstrate the usefulness of our main theoretical result (Theorem 1) in the following

domains:

(i) Certifying model accuracy under natural shifts (Section 2.5): We consider three image transformations: color shift, hue shift and changes in brightness and saturation (SV shift). Figure (2.1) visualizes CIFAR-10 [76] images under each of these transformations and reports the corresponding certified accuracies obtained by our method. Figure (2.2) plots the accuracy of two base models (trained on CIFAR-10 images with and without noise in the transformation space) under a shifted distribution and compares it with the certified accuracy of a robust model (noise-trained model smoothed using input randomization). These results demonstrate that our certificates are significant and non-vacuous (see appendix I for more details). In figures (2.5) and (2.8), we plot the certified accuracies for different values of training and smoothing noise – first for the CIFAR-10 dataset and then confirm our results on the SVHN dataset [77].

(ii) Certifying population level robustness against adversarial attacks (Section 2.6): The distribution of instances generated by an adversarial attack can also be viewed as a shift in the input distribution within a Wasserstein bound. Unlike existing certification techniques which assume a fixed perturbation budget across all inputs [36, 37, 38, 39], our guarantees work for a more general threat model where the adversary is allowed to choose the perturbation size for each input instance as long as it respects the constraint on the average perturbation size over the entire data distribution. Also, our procedure only requires *one* sample from the smoothing distribution per input instance which makes computing population level certificates significantly more efficient than existing techniques. The certified accuracy we obtain significantly outperforms the base model under attack (figure 2.11).

(iii) Hardness results for generating "unlearnable" datasets (Section 2.7): Huang et al. [78] proposed a method to make regular datasets unusable for modern deep learning models by poisoning them with adversarial perturbations to interfere with the training of the model. The intended purpose is to increase privacy for sensitive data such as personal images uploaded to social media sites. The dataset is poisoned in such a way that a model that minimizes the loss on this data distribution will have low accuracy on clean test samples. We show that our framework can obtain verifiable lower bounds on the performance of a model trained on such unlearnable datasets. Our certificates guarantee that the performance of the robust model (using input randomization) will remain above a certain threshold on the test distribution even when the base model is trained on the poisoned dataset with a smoothing noise of suitable magnitude. This demonstrates a fundamental limitation in producing unlearnable datasets.

2.2 Related Work

Several methods for introducing corruptions during training have been shown to improve the empirical robustness of machine learning models [2, 3, 58, 59]. Training with input transformations, such as blurring, cropping and rotations, can improve test accuracy against these corruptions. However, these methods do not produce any guarantees on the performance of the model with respect to the amount of shift added to the distribution. Our method applies random input transformations during inference to make the model provably robust against any distribution shift within a certain Wasserstein distance. It is independent of the model architecture and training procedure, and can be coupled with

robust training techniques, such as noise or adversarial training, to improve the certified performance.

Randomized smoothing based approaches that aggregate model predictions over a large number of noised samples of the input [36, 37, 38, 39] and that use input randomization [79] have been studied in the context of certified adversarial robustness. Provable robustness for parameterized transformations on images also exist [80]. These techniques produce instance-wise fixed-budget certificates and do not generate robustness guarantees over the entire data distribution or allow varying perturbation sizes for different instances. Our work also differs from instance-wise adversarial attacks and defenses [81, 82] that use the Wasserstein distance (instead of conventional ℓ_p distances) to measure difference between an image and its perturbed version. In contrast, our certificates consider the Wasserstein distance between data distributions from which images themselves are sampled.

Robustness bounds on the population loss against Wasserstein shifts under the ℓ_2-distance [83, 84] have been derived assuming Lipschitz-continuity of the base model. These bounds depend on the Lipschitz constant for the underlying model, which can grow rapidly for deep neural networks. We produce guarantees on the accuracy of an arbitrary model without requiring any restrictive assumptions or a global Lipschitz bound. Additionally, our approach can certify robustness against non-ℓ_p changes, such as visible color shifts, for which the ℓ_2-norm of the perturbation in the image space will be very large. Another line of work proves generalization bounds with for divergence-based measures of distribution shift [85, 86, 87, 88] like KL-divergence, total variation distance and Hellinger distance. Divergence measures between two distributions become arbitrarily large (e.g. KL-divergence becomes infinity) or attain their maximal value (e.g. total

variation and Hellinger distances become equal to one) when their supports do not coincide. This drawback makes them unsuitable for measuring out-of-distribution data shifts which by definition have non-overlapping support. Wasserstein distance, on the other hand, captures the spatial separation of two distributions and produces a more meaningful measure of the distance even when their supports are disjoint.

2.3 Preliminaries and Notations

Let $\mathcal{D}$ be the data distribution representing a machine learning task over an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. We define a distribution shift as a covariate shift that only changes the distribution of the input element in samples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ drawn from $\mathcal{D}$ and leaves the output element unchanged, i.e., (x, y) changes to $(\tilde{x}, y)$ under the shift. Given a distance function $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ over the input space, we define the following distance function between two tuples $\tau_1 = (x_1, y_1)$ and $\tau_2 = (x_2, y_2)$ to capture the above shift:

$$d(\tau_1, \tau_2) = \begin{cases} d_{\mathcal{X}}(x_1, x_2) & \text{if } y_1 = y_2 \\ \infty & \text{otherwise.} \end{cases} \tag{2.1}$$

Let $\tilde{\mathcal{D}}$ denote a shift in the original data distribution $\mathcal{D}$ such that the Wasserstein distance under d between $\mathcal{D}$ and $\tilde{\mathcal{D}}$ is bounded by ϵ (i.e., $W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \epsilon$). Define the set of all joint probability distributions with marginals $\mu_{\mathcal{D}}$ and $\mu_{\tilde{\mathcal{D}}}$ as follows:

$$\Gamma(\mathcal{D}, \tilde{\mathcal{D}}) = \left\{ \gamma \text{ s.t. } \int_{\mathcal{X}\times\mathcal{Y}} \gamma(\tau_1, \tau_2) d\tau_2 = \mu_{\mathcal{D}}(\tau_1) \text{ and } \int_{\mathcal{X}\times\mathcal{Y}} \gamma(\tau_1, \tau_2) d\tau_1 = \mu_{\tilde{\mathcal{D}}}(\tau_2) \right\}.$$

The Wasserstein bound implies that there exists an element $\gamma^* \in \Gamma(\mathcal{D}, \tilde{\mathcal{D}})$ such that

$$\mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[d(\tau_1, \tau_2)] \leq \epsilon. \tag{2.2}$$

Let $\mathcal{S} : \mathcal{X} \to \Delta(\mathcal{X})$ be a function mapping each element $x \in \mathcal{X}$ to a smoothing distribution $\mathcal{S}(x)$, where $\Delta(\mathcal{X})$ is the set of all probability distributions over $\mathcal{X}$. For example, smoothing with an isometric Gaussian noise distribution with variance σ^2 can be denoted as $\mathcal{S}(x) = \mathcal{N}(x, \sigma^2 I)$. Let the total variation between the smoothing distributions at two points x_1 and x_2 be bounded by a concave increasing function ψ of the distance between them, i.e.,

$$\mathsf{TV}(\mathcal{S}(x_1), \mathcal{S}(x_2)) \leq \psi(d_{\mathcal{X}}(x_1, x_2)). \tag{2.3}$$

For example, when the distance function d is the ℓ_2-norm of the difference of x_1 and x_2, and the smoothing distribution is an isometric Gaussian $\mathcal{N}(0, \sigma^2 I)$ with variance σ^2, $\psi(\cdot) = \mathrm{erf}(\cdot/2\sqrt{2}\sigma)$ is a valid upper bound on the above total variation that is concave in the positive domain (see Appendix G for more examples).

Consider a function $h : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ that represents the performance (e.g., accuracy) of a model μ over all possible input-output pairs. For example, in the case of a classifier $\mu : \mathcal{X} \to \mathcal{Y}$ that maps inputs from space $\mathcal{X}$ to a class label in $\mathcal{Y}$, $h(x, y) \coloneqq \mathbf{1}\{\mu(x) = y\}$ could indicate whether the prediction of μ on x matches the desired output label y or not. Another example could be that of segmentation/detection tasks, where y represents a region on an input image x. Then, $h(x, y) \coloneqq \mathrm{IoU}(\mu(x), y)$[1] could represent the overlap between the predicted regions $\mu(x)$ and the ground truth y. The overall

[1] IoU stands for Intersection over Union.

accuracy of the model μ under $\mathcal{D}$ is then given by $\mathbb{E}_{(x,y)\in\mathcal{D}}[h(x,y)]$. Now, define a robust model $\bar{\mu}(x) = \mu(x')$ where $x' \sim \mathcal{S}(x)$ which simply applies the base model μ on a randomized version of the input x sampled from a smoothing distribution $\mathcal{S}(x)$. Our goal is to bound the difference in the expected performance of the robust model between the original distribution $\mathcal{D}$ and the shifted distribution $\tilde{\mathcal{D}}$. Let $\bar{h}$ be the performance function for the robust model $\bar{\mu}$ defined as

$$\bar{h}(x,y) = \mathbb{E}_{x'\sim\mathcal{S}(x)}[h(x',y)]. \tag{2.4}$$

Then, the accuracy of the robust model $\bar{\mu}$ under $\mathcal{D}$ is given by $\mathbb{E}_{(x,y)\in\mathcal{D}}[\bar{h}(x,y)]$. Our result in Theorem 1 bounds the difference between the expectation of $\bar{h}$ under $\mathcal{D}$ and $\tilde{\mathcal{D}}$ with $\psi(\epsilon)$.

2.3.1 Parameterized Transformations

We apply our distributional certificates to produce guarantees on the accuracy of an image classifier under natural transformations such as color shifts, hue shifts and changes in brightness and saturation. We model each transformation as a function $\mathcal{T} : \mathcal{X} \times P \to \mathcal{X}$ over the image space $\mathcal{X}$ and a parameter space P. It takes an image $x \in \mathcal{X}$ and a parameter vector $\theta \in P$ as inputs and outputs a transformed image $x' = \mathcal{T}(x,\theta) \in \mathcal{X}$. An example of such a transformation could be a color shift in an RGB image produced by scaling the intensities in the red, green and blue channels $x = (\{x^R_{ij}\}, \{x^G_{ij}\}, \{x^B_{ij}\})$ defined as $\mathsf{CS}(x,\theta) = (2^{\theta_R}\{x^R_{ij}\}, 2^{\theta_G}\{x^G_{ij}\}, 2^{\theta_B}\{x^B_{ij}\})/\mathsf{MAX}$ for a tuple $\theta = (\theta_R, \theta_G, \theta_B)$, where MAX is the maximum of all the RGB values after scaling. Additive perturbations in

the input space can also be captured as parameterized transformations, e.g., $\mathsf{VT}(x, \theta) = x + \theta$. We assume that the transformation returns x if the parameters are all zero, i.e., $\mathcal{T}(x, 0) = x$ and that the composition of two transformations with parameters θ_1 and θ_2 is a transformation with parameters $\theta_1 + \theta_2$ (additive composability), i.e.,

$$\mathcal{T}(\mathcal{T}(x, \theta_1), \theta_2) = \mathcal{T}(x, \theta_1 + \theta_2). \tag{2.5}$$

Given a norm $\|\cdot\|$ in the parameter space P, we define a distance function in the input space $\mathcal{X}$ as follows:

$$d_{\mathcal{T}}(x_1, x_2) = \begin{cases} \min\{\|\theta\| \mid \mathcal{T}(x_1, \theta) = x_2\} & \text{if } \exists\theta \text{ s.t. } \mathcal{T}(x_1, \theta) = x_2 \\ \infty & \text{otherwise.} \end{cases} \tag{2.6}$$

Now, define a smoothing distribution $\mathcal{S}(x) = \mathcal{T}(x, \mathcal{Q}(0))$ for some distribution $\mathcal{Q}$ in the parameter space of $\mathcal{T}$ such that $\forall\theta \in P$, $\mathcal{Q}(\theta) = \theta + \mathcal{Q}(0)$ is the distribution of $\theta + \delta$ where $\delta \sim \mathcal{Q}(0)$, and $\mathsf{TV}(\mathcal{Q}(0), \mathcal{Q}(\theta)) \leq \psi(\|\theta\|)$ for a concave function ψ. For example, $\mathcal{Q}(\cdot) = \mathcal{N}(\cdot, \sigma^2 I)$ satisfies these properties for $\psi(\cdot) = \mathrm{erf}(\cdot/2\sqrt{2}\sigma)$. Then, the following lemma holds (proof in Appendix B):

Lemma 1. *For two points $x_1, x_2 \in \mathcal{X}$ such that $d_{\mathcal{T}}(x_1, x_2)$ is finite,*

$$\mathsf{TV}(\mathcal{S}(x_1), \mathcal{S}(x_2)) \leq \psi(d_{\mathcal{T}}(x_1, x_2)).$$

2.4 Certified Distributional Robustness

In this section, we state our main theoretical result which shows that the difference in the expectation of the performance function $\bar{h}$ of the robust model (equation (2.4)) under the original distribution $\mathcal{D}$ and any shifted distribution $\tilde{\mathcal{D}}$ within a Wasserstein distance of ϵ from $\mathcal{D}$ is bounded by $\psi(\epsilon)$, where ψ is the concave upper bound on the total variation between the smoothing distributions at two points x_1 and x_2 as defined in condition (2.3).

Theorem 1. *Given a function $h : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, define its smoothed version as $\bar{h}(x, y) = \mathbb{E}_{x' \sim \mathcal{S}(x)}[h(x', y)]$. Then,*

$$\forall\ \tilde{\mathcal{D}}\ \textit{s.t.}\ W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \epsilon, \quad \left| \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)] \right| \leq \psi(\epsilon).$$

We defer the proof to Appendix A. Note that this certificate does not require us to compute the Wasserstein distance between $\mathcal{D}$ and $\tilde{\mathcal{D}}$. Given a value for ϵ, it holds for *all* distributions $\tilde{\mathcal{D}}$ such that $W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \epsilon$. Our certified guarantees hold for the entire input distribution (potentially continuous) and not just for a finite set of samples. The intuition behind the above bound is that if the overlap between the smoothing distributions between two individual points does not decrease rapidly with the distance between them, then the overlap between $\mathcal{D}$ and $\tilde{\mathcal{D}}$ augmented with the smoothing distribution is high when the Wasserstein distance between them is small. The key observation here is that the total variation of the individual smoothing distributions can be upper bounded by a convex function ψ and this upper bound can then be generalised over the entire distribution using

Jensen's inequality. The above guarantee implies that for any distribution $\tilde{\mathcal{D}}$ that is within a Wasserstein distance of ϵ from the original distribution $\mathcal{D}$, the accuracy of the model under $\tilde{\mathcal{D}}$ can be bounded as $\mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2,y_2)] \geq \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1,y_1)] - \psi(\epsilon)$.

2.4.1 Computing the Certificate and Empirical Evaluations

Given a target Wasserstein bound ϵ and an appropriate function ψ, we simply need to calculate the expected performance of the robust model over the original distribution $\mathcal{D}$, i.e., $\mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1,y_1)]$. Since we only have sample access to the original distribution $\mathcal{D}$, we estimate the expected performance on $\mathcal{D}$, i.e. $\mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1,y_1)]$, using a finite number of samples. In our experiments, we compute a high-confidence lower bound of this quantity using the Clopper-Pearson method [89] that holds with $1-\alpha$ probability, for some $\alpha > 0$ (usually 0.001). Note that although we calculate the bound with a finite number of samples from the distribution $\mathcal{D}$, this lower bound holds for the expectation over the *entire* distribution and not just for the samples. See Appendix C for pseudocodes of the prediction and certification steps.

To compare our certified guarantees against the empirical performance of an undefended model under distribution shifts, we design shifted distributions using natural and adversarial transformations on the original distribution. We ensure that the constructed distribution shift is within the desired Wasserstein distance using two methods:

1. By construction: We analytically guarantee beforehand that the applied transformation does not exceed the Wasserstein bound. For example, in Figure 2.2, we report the empirical performance of the base models under distribution shifts constructed by

adding a noise vector from a fixed distribution like a Gaussian distribution of a certain variance in the transformation space (see Appendix I).

2. By estimation: We compute a high-confidence bound on the average perturbation added to a finite number of samples to bound the Wasserstein distance. For example, in Section 2.6, when reporting the undefended baseline performance, we measure $\mathbb{E}[\|\text{Adv}(x) - x\|_2]$ on the test set, and use Hoeffding's inequality to derive from this a 99% confidence upper bound on the true, population expectation $\mathbb{E}_{x\sim\mathcal{D}}[\|\text{Adv}(x) - x\|_2]$. By Equation 2.7, this is a (high-probability) upper bound on the Wasserstein distance of the distribution shift.

In the following sections, we apply our main theoretical result to obtain certified robustness guarantees against several different distribution shifts – natural shifts, unlearnable distributions and adversarial shifts. We experiment on two image classification datasets, namely CIFAR-10 [76] and SVHN [77], and observe that the our certificates can obtain meaningful performance guarantees and exhibit similar trends for both datasets.

2.5 Certified Accuracy against Natural Transformations

We certify the accuracy of a ResNet-110 model and a ResNet-20 model trained on CIFAR-10 and SVHN images respectively under three types of transformations: color shifts, hue shifts and variation in brightness and saturation (SV shift). We train our models with varying levels of noise in the transformation space and evaluate their certified performance using smoothing distributions of different standard deviations. For color and SV shifts, we show how the certified accuracy varies as a function of the Wasserstein

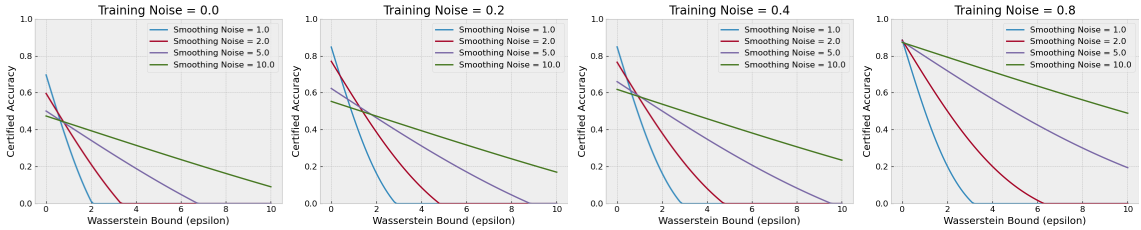


Figure 2.3: Color Shift – CIFAR-10

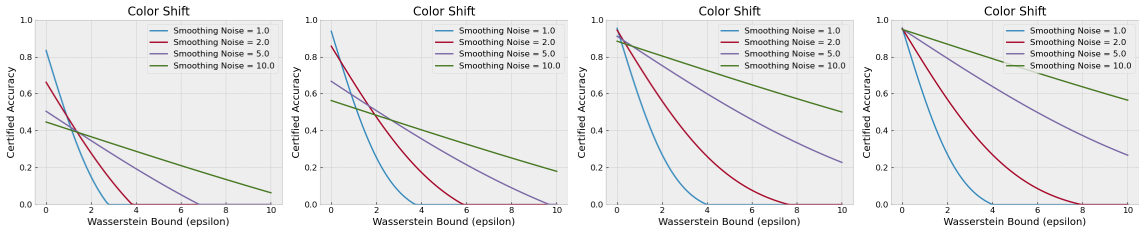


Figure 2.4: Color Shift – SVHN

Figure 2.5: Certified accuracy under color shifts for (a) CIFAR-10 and (b) SVHN. Each plot corresponds to a particular training noise and each curve corresponds to a particular smoothing noise.

distance as we change the training and smoothing noise. For hue shift, we use a smoothing distribution (with fixed noise level) that is invariant to rotations in hue space because of which the certified accuracy remains constant with respect to the corresponding Wasserstein distance. We train the ResNet-110 models for 90 epochs which takes a few hours on a single NVIDIA GeForce RTX 2080 Ti GPU and the ResNet-20 models for 40 epochs which takes around twenty minute on the same GPU. Once the models have been trained, computing the distribution level Wasserstein certificates using 10^5 samples with 99.9% confidence takes only about 25 seconds for each model.

2.5.1 Color Shifts

Denote an RGB image x as an $H \times W$ array of pixels where the red, green and blue components of the pixel in the ith row and jth column are given by the tuple $x_{ij} = (r, g, b)_{ij}$. Let $r_{\max}, g_{\max}$ and $b_{\max}$ be the maximum values of the red, green and blue

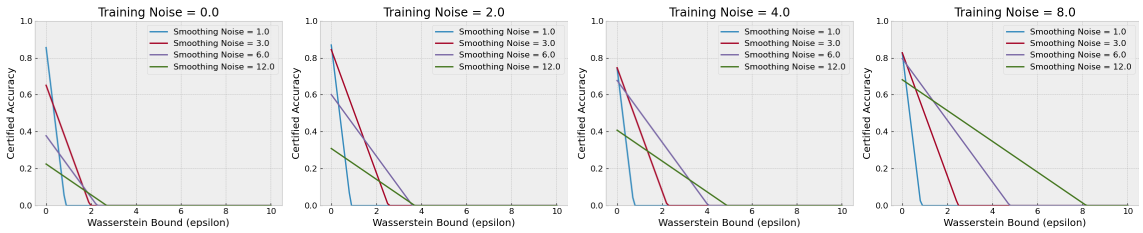


Figure 2.6: Brightness and Saturation Shift – CIFAR-10

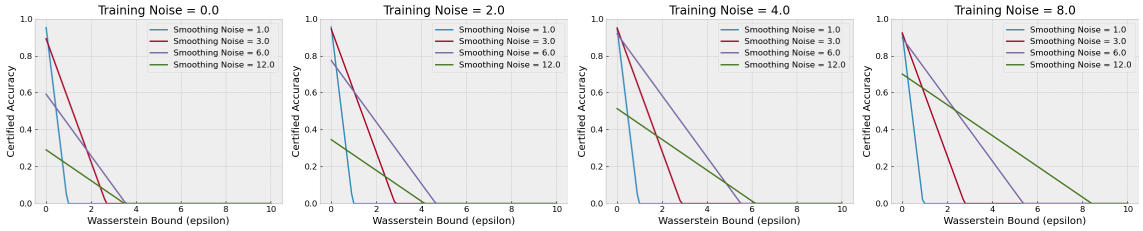


Figure 2.7: Brightness and Saturation Shift – SVHN

Figure 2.8: Certified accuracy under brightness and saturation changes for (a) CIFAR-10 and (b) SVHN images. Each plot corresponds to a particular training noise and each curve corresponds to a particular smoothing noise.

channels, respectively. Assume that the RGB values are in the interval $[0, 1]$ normalized such that the maximum over all intensity values is one, i.e., $\max(r_{\max}, g_{\max}, b_{\max}) = 1$. Define a color shift of the image x for a parameter vector $\theta \in \mathbb{R}^3$ as

$$\mathsf{CS}(x, \theta) = \left\{ \frac{(2^{\theta_R} r, 2^{\theta_G} g, 2^{\theta_B} b)_{ij}}{\max(2^{\theta_R} r_{\max}, 2^{\theta_G} g_{\max}, 2^{\theta_B} b_{\max})} \right\}^{H \times W}$$

which scales the intensities of each channel by the corresponding component of θ raised to the power of two and then normalizes the scaled image so that the maximum intensity is one. For example, $\theta = (1, -1, 0)$ would first double all the red intensities, halve the green intensities and leave the blue intensities unchanged, and then, normalize the image so that the maximum intensity value over all the channels is equal to one. The above transformation can be shown to satisfy the additive composability property in condition (2.5). See Appendix H for a proof.

Given an image x, we define a smoothing distribution around x in the parameter

space as $\mathsf{CS}(x, \delta)$ where $\delta \sim \mathcal{N}(0, \sigma^2 I_{3\times 3})$. Define the distance function d_{CS} as described in (2.6) using the ℓ_2-norm in the parameter space. For a distribution $\tilde{\mathcal{D}}$ within a Wasserstein distance of ϵ from the original distribution $\mathcal{D}$, the performance of the smoothed model on $\tilde{\mathcal{D}}$ can be bounded as $\mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)] \geq \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)] - \text{erf}(\epsilon/2\sqrt{2}\sigma)$. Figure 2.5 plots the certified accuracy under color shift with respect to the Wasserstein bound ϵ for different values of training and smoothing noise. In Appendix K, we consider a smoothing distribution that randomly picks one color channel achieving a constant certified accuracy of 87.1% with respect to ϵ.

2.5.2 Brightness and Saturation Changes

Define the following transformation in the HSV space of an image that shifts the mean of the saturation (S) and brightness (V) values for each pixel by a certain amount:

$$\mathsf{SV}(x, \theta) = \left\{ \left(h, \frac{s + (2^{\theta_S} - 1)s_{\text{mean}}}{\mathsf{MAX}}, \frac{v + (2^{\theta_V} - 1)v_{\text{mean}}}{\mathsf{MAX}} \right)_{ij} \right\}^{H\times W}$$

where $s_{\text{mean}}, s_{\text{max}}, v_{\text{mean}}$ and v_{max} are the means and maximums of the saturation and brightness values respectively before the shift is applied and $\mathsf{MAX} = \max(s_{\text{max}} + (2^{\theta_S} - 1)s_{\text{mean}}, v_{\text{max}} + (2^{\theta_V} - 1)v_{\text{mean}})$ is the maximum of the brightness and saturation values after the shift. Similar to color shift , the SV transformation can also be shown to satisfy additive composability (Appendix H). Figure 2.8 plots the certified accuracy under saturation and brightness changes with respect to ϵ for different values of training and smoothing noise. The smoothing distribution is uniform in the range $[0, a]^2$ in the parameter space, the distance function is the ℓ_1-norm and $\psi(\epsilon) = \min(\epsilon/a, 1)$.

2.6 Population-Level Certificates against Adversarial Attacks

In this section, we consider the ℓ_2-distance in the image space to measure the Wasserstein distance instead of a parameterized transformation (see Appendix D for a detailed version). We use a pixel-space Gaussian smoothing distribution $\mathcal{S}(x) = \mathcal{N}(x, \sigma^2 I)$ to obtain robustness guarantees under this metric. To motivate this, consider an adversarial attacker $\text{Adv} : \mathcal{X} \to \mathcal{X}$, which takes an image x and computes perturbation $\text{Adv}(x)$ to try and fool a model into misclassifying the input. If $(x, y) \sim \mathcal{D}$, define $\tilde{\mathcal{D}}$ to be the distribution of the tuples $(\text{Adv}(x), y)$. Defining d in 2.1 using $d_{\mathcal{X}} = \ell_2$, it is easy to show that:

$$W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \mathbb{E}_{x \sim \mathcal{D}}[\|\text{Adv}(x) - x\|_2] \tag{2.7}$$

Results on CIFAR-10 are presented in Figure 2.9 and results on SVHN are available in Appendix D. For CIFAR-10, we use ResNet-110 models trained under noise from Cohen et al. [36]. The solid lines represent certified accuracies for different smoothing noises and the black dashed line represents the empirical performance of an undefended model under attack. For the undefended baseline, we give the performance of an undefended model against a *strategic* attacker, which first finds a minimal ℓ_2 attack for each sample via [24]. If this attack is too large in magnitude ($\ell_2 >$ a threshold γ), it instead chooses not to attack the sample. This "saves" the attack budget (i.e., the average

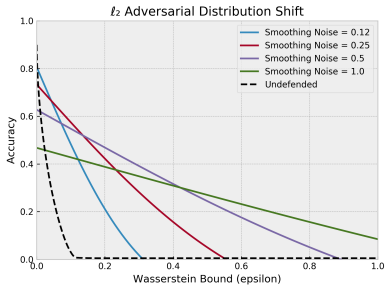


Figure 2.9: Distributional certificates against adversarial attacks on CIFAR-10.

attack magnitude and therefore the Wasserstein shift) for easier samples. The size of the Wasserstein shift can be adjusted by varying γ.

2.7 Hardness Results on Unlearnability

In this section, we show that the pixel-space ℓ_2-Wasserstein distributional robustness certificate shown above can also be applied to establish a hardness result in creating provably "unlearnable" datasets [78]. These datasets contain "poisoned" samples which make any classifier trained on the released data achieve a high training and validation accuracy, but a low test accuracy on non-poisoned samples from the original data distribution. This technique has legitimate applications, such as protecting privacy by preventing one's personal data from being learned, but may also have malicious uses (e.g., a malicious actor could sell a useless classifier that nevertheless has good performance on a provided validation set.) We can view the "clean" data distribution as $\mathcal{D}$, and the distribution of the poisoned samples (i.e., the unlearnable distribution) as $\tilde{\mathcal{D}}$. If the magnitude of the perturbations is limited, Theorem 1 implies that the accuracy on $\mathcal{D}$ and $\tilde{\mathcal{D}}$ must be similar, implying that our algorithm is provably resistant to unlearnablility attacks, effectively establishing provable hardness results to create unlearnable datasets.

In order to apply our guarantees, we make a few modifications to the attack proposed in Huang et al. [78]. First, we bound each poisoning perturbation on the released dataset to within an ϵ-radius ℓ_2 ball, rather than an ℓ_∞ ball. From Equation 2.7, this ensures that $W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \epsilon$. Second, we consider an "offline" version of the attack. In the original attack [78], perturbations for the entire dataset are optimized simultaneously with a proxy classifier model in an iterative manner. This makes the perturbations applied to each

sample non-I.I.D., (they may depend on each other through proxy-model parameters) which makes deriving generalizable guarantees for it difficult.

However, this simultaneous proxy-model training and poisoning may not always represent a realistic threat model. In particular, an actor releasing "unlearnable" data at scale may not be able to constantly update the proxy model being used. For example, consider an "unlearnability" module in a camera, which would make photos unusable as training data. Because the camera itself has access to only a small number of photographs, such a module would likely rely on a fixed, pre-trained proxy classifier model to create the poisoning perturbations. To model this, we consider a threat model where the proxy classifier is first optimized using an unreleased dataset: the released "unlearnable" samples are then perturbed independently using this fixed proxy model. We see in Figure 2.10 that our modified attack is still highly effective at making data unlearnable, as shown by the high validation and low test accuracy of the undefended baseline.

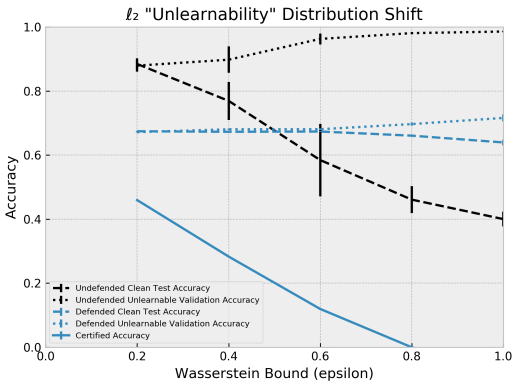


Figure 2.10: Distributional certificates for unlearnable datasets on CIFAR-10. The smoothing noise used is 0.4. Results for other values are reported in the appendix.

2.8 Conclusion

We show that it is possible to certify the distributional robustness of a general deep neural network without increasing its computational requirements. We obtain robustness guarantees with respect to the Wasserstein distance of the distribution shift which is a more suitable metric for out-of-distribution shifts than divergence measures such as KL-divergence and total variation. We only consider predefined distance functions in this work which may not be suitable for capturing more sophisticated distribution shifts such as perceptual changes. A future direction of research could be to adapt our certificates for learnable transformations for domain generalization and adaptation.

2.9 Appendices

A Proof of Theorem 1

Statement: *Given a function* $h : \mathcal{X} \times \mathcal{Y} \to [0, 1]$*, define its smoothed version as* $\bar{h}(x, y) = \mathbb{E}_{x' \sim \mathcal{S}(x)}[h(x', y)]$*. Then,*

$$\left| \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)] \right| \leq \psi(\epsilon).$$

Proof. Let $\tau_1 = (x_1, y_1)$ and $\tau_2 = (x_2, y_2)$ denote the input-output tuples sampled from $\mathcal{D}$ and $\tilde{\mathcal{D}}$ respectively. Then, for the joint distribution $\gamma^* \in \Gamma(\mathcal{D}, \tilde{\mathcal{D}})$ in (2.2), we have

$$\mathbb{E}_{\tau_1\sim\mathcal{D}}[\bar{h}(\tau_1)] = \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[\bar{h}(\tau_1)] \quad \text{and} \quad \mathbb{E}_{\tau_2\sim\tilde{\mathcal{D}}}[\bar{h}(\tau_2)] = \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[\bar{h}(\tau_2)].$$

This is because when (τ_1, τ_2) is sampled from the joint distribution γ^*, τ_1 and τ_2 individually have distributions $\mathcal{D}$ and $\tilde{\mathcal{D}}$ respectively. Also, since the expected distance between $\tau_1 = (x_1, y_1)$ and $\tau_2 = (x_2, y_2)$ is finite, the output elements of the sampled tuples must be the same, i.e. $y_1 = y_2 = y$ (say). See lemma 2 below. Then,

$$\begin{aligned}
&\left| \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)] \right| \\
&\qquad = \left| \mathbb{E}_{\tau_1\sim\mathcal{D}}[\bar{h}(\tau_1)] - \mathbb{E}_{\tau_2\sim\tilde{\mathcal{D}}}[\bar{h}(\tau_2)] \right| \\
&\qquad = \left| \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[\bar{h}(\tau_1) - \bar{h}(\tau_2)] \right| \\
&\qquad \leq \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[|\bar{h}(\tau_1) - \bar{h}(\tau_2)|].
\end{aligned}$$

Now, from definition (2.4) and for $i = 1$ and 2,

$$\bar{h}(\tau_i) = \bar{h}(x_i, y) = \mathbb{E}_{x_i' \sim \mathcal{S}(x_i)}[h(x_i', y)] = \mathbb{E}_{x_i' \sim \mathcal{S}(x_i)}[g(x_i')]$$

can be expressed as the expected value of a function $g : \mathcal{X} \to [0, 1]$ under distribution $\mathcal{S}(x_i)$. Without loss of generality, assume $\mathbb{E}_{x_1' \sim \mathcal{S}(x_1)}[g(x_1')] \geq \mathbb{E}_{x_2' \sim \mathcal{S}(x_2)}[g(x_2')]$. Then,

$$\begin{aligned}
&\left|\mathbb{E}_{x_1' \sim \mathcal{S}(x_1)}[g(x_1')] - \mathbb{E}_{x_2' \sim \mathcal{S}(x_2)}[g(x_2')]\right| \\
&\qquad = \int_{\mathcal{X}} g(x)\mu_1(x)dx - \int_{\mathcal{X}} g(x)\mu_2(x)dx \\
&\qquad\qquad\qquad\qquad (\mu_1 \text{ and } \mu_2 \text{ are the PDFs of } \mathcal{S}(x_1) \text{ and } \mathcal{S}(x_1)) \\
&\qquad = \int_{\mathcal{X}} g(x)(\mu_1(x) - \mu_2(x))dx \\
&\qquad = \int_{\mu_1 > \mu_2} g(x)(\mu_1(x) - \mu_2(x))dx - \int_{\mu_2 > \mu_1} g(x)(\mu_2(x) - \mu_1(x))dx \\
&\qquad \leq \int_{\mu_1 > \mu_2} \max_{x' \in \mathcal{X}} g(x')(\mu_1(x) - \mu_2(x))dx - \int_{\mu_2 > \mu_1} \min_{x' \in \mathcal{X}} g(x')(\mu_2(x) - \mu_1(x))dx \\
&\qquad \leq \int_{\mu_1 > \mu_2} (\mu_1(x) - \mu_2(x))dz \\
&\qquad\qquad\qquad\qquad (\text{since } \max_{x' \in \mathcal{X}} g(x') \leq 1 \text{ and } \min_{x' \in \mathcal{X}} g(x') \geq 0) \\
&\qquad = \frac{1}{2}\int_{\mathcal{X}} |\mu_1(x) - \mu_2(x)|dx = \mathsf{TV}(\mathcal{S}(x_1), \mathcal{S}(x_2)).
\end{aligned}$$

(since $\int_{\mu_1 > \mu_2}(\mu_1(x) - \mu_2(x))dx = \int_{\mu_2 > \mu_1}(\mu_2(x) - \mu_1(x))dx = \frac{1}{2}\int_{\mathcal{X}} |\mu_1(x) - \mu_2(x)|dx)$

Thus, from (2.1) and (2.3), we have $|\bar{h}(\tau_1) - \bar{h}(\tau_2)| \leq \psi(d_{\mathcal{X}}(x_1, x_2)) = \psi(d(\tau_1, \tau_2))$, and

therefore,

$$\begin{aligned}
&\left|\mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1,y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2,y_2)]\right| \\
&\qquad\qquad \leq \mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[\psi(d(\tau_1,\tau_2))] \\
&\qquad\qquad \leq \psi\left(\mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[d(\tau_1,\tau_2)]\right). \quad (\psi \text{ is concave, Jensen's inequality})
\end{aligned}$$

Hence, from (2.2) and since ψ is non-decreasing, we have

$$\left|\mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1,y_1)] - \mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2,y_2)]\right| \leq \psi(\epsilon).$$

□

Lemma 2. *Let* $\Omega = \{(\tau_1,\tau_2)$ *s.t.* $y_1 \neq y_2$ *where* $\tau_1 = (x_1,y_1)$ *and* $\tau_2 = (x_2,y_2)\}$. *Then*

$$\mathbb{P}_{(\tau_1,\tau_2)\sim\gamma^*}[(\tau_1,\tau_2) \in \Omega] = 0.$$

Proof. Assume, for the sake of contradiction, that

$$\mathbb{P}_{(\tau_1,\tau_2)\sim\gamma^*}[(\tau_1,\tau_2) \in \Omega] \geq p$$

for some $p > 0$. From condition (2.2), we have

$$\mathbb{E}_{(\tau_1,\tau_2)\sim\gamma^*}[d(\tau_1,\tau_2)] \leq \epsilon.$$

By the law of total expectation

$$\begin{aligned}\mathbb{E}_{\gamma^*}[d(\tau_1,\tau_2)] =& \mathbb{E}_{\gamma^*}[d(\tau_1,\tau_2) \mid (\tau_1,\tau_2) \in \Omega]\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] \\ &+ \mathbb{E}_{\gamma^*}[d(\tau_1,\tau_2) \mid (\tau_1,\tau_2) \notin \Omega]\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \notin \Omega].\end{aligned}$$

We replace $(\tau_1, \tau_2) \sim \gamma^*$ with just γ^* in the subscripts for brevity. Since both summands are non-negative,

$$\mathbb{E}_{\gamma^*}[d(\tau_1,\tau_2) \mid (\tau_1,\tau_2) \in \Omega]\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] \leq \epsilon.$$

Consider a real number $l > \epsilon/p$. Then, for any $(\tau_1, \tau_2) \in \Omega$, from definition (2.1) and because $y_1 \neq y_2$, $d(\tau_1, \tau_2) \geq l$. Therefore, $\mathbb{E}_{\gamma^*}[d(\tau_1, \tau_2) \mid (\tau_1, \tau_2) \in \Omega] \geq l$ and

$$\begin{aligned} l\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] &\leq \mathbb{E}_{\gamma^*}[d(\tau_1,\tau_2) \mid (\tau_1,\tau_2) \in \Omega]\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] \\ l\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] &\leq \epsilon \\ \mathbb{P}_{\gamma^*}[(\tau_1,\tau_2) \in \Omega] &\leq \epsilon/l < p, \end{aligned}$$

which contradicts our initial assumption. □

B Proof of Lemma 1

Statement: *For two points $x_1, x_2 \in \mathcal{X}$ such that $d_{\mathcal{T}}(x_1, x_2)$ is finite,*

$$\mathsf{TV}(\mathcal{S}(x_1), \mathcal{S}(x_2)) \leq \psi(d_{\mathcal{T}}(x_1, x_2)).$$

Proof. Consider the θ for which $d_{\mathcal{T}}(x_1, x_2) = \|\theta\|$. Then, $\mathcal{T}(x_1, \theta) = x_2$.

$$\begin{aligned}
\mathsf{TV}(\mathcal{S}(x), \mathcal{S}(x_2)) &= \mathsf{TV}(\mathcal{T}(x, \mathcal{Q}(0)), \mathcal{T}(z, \mathcal{Q}(0))) \\
&= \mathsf{TV}(\mathcal{T}(x, \mathcal{Q}(0)), \mathcal{T}(\mathcal{T}(x, \theta), \mathcal{Q}(0))) \\
&= \mathsf{TV}(\mathcal{T}(x, \mathcal{Q}(0)), \mathcal{T}(x, \theta + \mathcal{Q}(0))) \\
&\qquad\qquad \text{(additive composability, equation (2.5))} \\
&= \mathsf{TV}(\mathcal{T}(x, \mathcal{Q}(0)), \mathcal{T}(x, \mathcal{Q}(\theta))). \qquad \text{(definition of } \mathcal{Q})
\end{aligned}$$

Let A be the event in the space M that maximizes the difference in the probabilities assigned to A by $\mathcal{T}(x, \mathcal{Q}(0))$ and $\mathcal{T}(x, \mathcal{Q}(\theta))$. Let $u : P \to [0, 1]$ be a function that returns the probability (over the randomness of $\mathcal{T}$) of any parameter $\eta \in P$ being mapped to a point in A, i.e., $u(\eta) = \mathbb{P}\{\mathcal{T}(x, \eta) \in A\}$. For a deterministic transformation $\mathcal{T}$, u is a 0/1 function. Then, the probabilities assigned by $\mathcal{T}(x, \mathcal{Q}(0))$ and $\mathcal{T}(x, \mathcal{Q}(\theta))$ to A is equal to $\mathbb{E}_{\eta \sim \mathcal{Q}(0)}[u(\eta)]$ and $\mathbb{E}_{\eta \sim \mathcal{Q}(\theta)}[u(\eta)]$. Therefore,

$$\begin{aligned}
\mathsf{TV}(\mathcal{S}(x), \mathcal{S}(x_2)) &= |\mathbb{E}_{\eta \sim \mathcal{Q}(0)}[u(\eta)] - \mathbb{E}_{\eta \sim \mathcal{Q}(\theta)}[u(\eta)]| \\
&\leq \mathsf{TV}(\mathcal{Q}(0), \mathcal{Q}(\theta)) \\
&\leq \psi(\|\theta\|) = \psi(d_{\mathcal{T}}(x_1, x_2)). \qquad \text{(definition of } \mathcal{Q} \text{ and } d_{\mathcal{T}})
\end{aligned}$$

□

Algorithm 1: Prediction

Input: Model μ, input instance x.
Output: Robust prediction y.
Randomize input: $x' \sim \mathcal{S}(x)$.
Evaluate model: $y = \mu(x')$.
Return y.

Algorithm 2: Certification

Input: Accuracy function h, data distribution $\mathcal{D}$, Wasserstein bound ϵ, integer n and $\alpha > 0$.
Output: Certified accuracy for bound ϵ.
$\mathsf{sum} = 0$.
for i in $1 \dots n$ **do**
  Sample $(x, y) \sim \mathcal{D}$.
  Sample $x' \sim \mathcal{S}(x)$.
  Compute $h(x', y)$.
  $\mathsf{sum} = \mathsf{sum} + h(x', y)$
end for
Compute $1 - \alpha$ confidence lower-bound $\underline{h}$ of $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\bar{h}(x, y)]$ using sum and n.
Return $\underline{h} - \psi(\epsilon)$.

C Pseudocode for Prediction and Certification

Algorithm 1 and Algorithm 2 describe the prediction and certification steps of our method.

D Population-Level Certificates against Adversarial Attacks

In this section, we consider the ℓ_2-distance in the image space to measure the Wasserstein distance instead of a parameterized transformation. We use a pixel-space Gaussian smoothing distribution $\mathcal{S}(x) = \mathcal{N}(x, \sigma^2 I)$ to obtain robustness guarantees under this metric. To motivate this, consider an adversarial attacker $\mathrm{Adv} : \mathcal{X} \to \mathcal{X}$, which takes an image x and computes perturbation $\mathrm{Adv}(x)$ to try and fool a model into misclassifying the input. If $(x, y) \sim \mathcal{D}$, define $\tilde{\mathcal{D}}$ to be the distribution of the tuples

$(\text{Adv}(x), y)$. Defining d in 2.1 using $d_{\mathcal{X}} = \ell_2$, it is easy to show that:

$$W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) \leq \mathbb{E}_{x \sim \mathcal{D}}[\|\text{Adv}(x) - x\|_2] \tag{2.8}$$

So, if the *average* magnitude of perturbations induced by Adv is less than ϵ (i.e., $[\|\text{Adv}(x) - x\|_2] < \epsilon$), then $W_1^d(\mathcal{D}, \tilde{\mathcal{D}}) < \epsilon$ which means that we can apply Theorem 1: the gap in the expected accuracy between $x \sim \mathcal{D}$ and $\text{Adv}(x) \sim \tilde{\mathcal{D}}$ will be at most $\psi(\epsilon)$. Note that, under this threat model, Adv can be strategic in its use of the average perturbation "budget": if a certain point x would require a very large perturbation to be misclassified, or is already misclassified, then $\text{Adv}(x)$ can save the budget by simply returning x and use it to attack a greater number of more vulnerable samples.

Note that our method differs from *sample-wise* certificates against ℓ_2 adversarial attacks which use randomized smoothing, such as Cohen et al. [36]. Specifically, we use only one smoothing perturbation (and therefore only one forward pass) per sample. Our guarantees are on the overall accuracy of the classifier, not on the stability of any particular prediction. Finally, as discussed, our threat model is different, because we allow the adversary to strategically choose which samples to attack, with the certificate dependent on the *Wasserstein* magnitude of the *distributional* attack.

Results on CIFAR-10 and SVHN are presented in Figure 2.11. For CIFAR-10, we use ResNet-110 models trained under noise from Cohen et al. [36]. For SVHN, we train our own models using the same training schedule as used for CIFAR-10 by [36], but we use ResNet-20 in place of ResNet-110. The solid lines represent certified accuracies for different smoothing noises and the black dashed line represents the empirical performance

of an undefended model under attack. For the undefended baseline (on an undefended classifier g), we first apply a Carlini and Wagner ℓ_2 attack to each sample x [24], generating adversarial examples x'. Define this attack as the function $CW(\cdot)$, such that $x' = CW(x, y; g)$, where y is the ground-truth label. (If the attack fails, $CW(x, y; g) = x$). We then define a *strategic* adversary Adv_γ that returns $CW(x, y; g)$ if $\|CW(x, y; g) - x\|_2 < \gamma$, otherwise it returns x.

By not attacking samples which would require the largest ℓ_2 perturbations to cause misclassification, this attack efficiently balances maximizing misclassification rate with minimizing the Wasserstein distance between $\mathcal{D}$ and $\tilde{\mathcal{D}}$. The threshold parameter γ controls the tradeoff between misclassifcation rate and the Wasserstein perturbation magnitude. Note that our attacker here is strategic in a way that takes more advantage of the distributional threat model than simply finding the minimal perturbation for each sample: by choosing to *not attack at all* on robust samples, it can successfully attack a larger number of more vulnerable samples. The 'Undefended' baseline in Figure 2.11 plots the accuracy on attacked test samples under adversary Adv_γ, for a sweep of values of γ, against an upper bound on the Wasserstein distance, given by $\mathbb{E}_{x\sim\mathcal{D}}[\|\text{Adv}_\gamma(x) - x\|_2]$. (In order to estimate $\mathbb{E}_{x\sim\mathcal{D}}[\|\text{Adv}_\gamma(x) - x\|_2]$, we compute the average perturbation size over the test set and use Hoeffding inequality to upper-bound the population expectation with 99% confidence.) We can observe a large gap between this undefended model performance under attack, and the certified robustness of our model, showing that our certificate is highly nonvacuous. In Appendix E, we include results to show the empirical robustness of the smoothed classifiers under an "adaptive" attack, based on the attack on sample-wise ℓ_2 smoothing proposed by Salman et al. [39]. We also test an alternate form strategic attacker

on the baseline model that does not requires us to estimate the average perturbation size empirically (Appendix F).

E Empirical Attacks on ℓ_2-distributional robustness.

In this section, we describe an empirical attack on ℓ_2-distributional smoothing. Our attack is based on the attack from Salman et al. [39], and we use the code for PGD attack against smoothed classifiers from that work as a base, but there are a few considerations we must make.

First, while the goal of the attacker in Salman et al. [39] is to change the output of a classifier that uses the *expected* logits, the goal in our case is to instead reduce the average classification accuracy of *each noise instance*. Concretely, Salman et al. [39] uses an attacker loss function for each sample x, y of the following form:

$$\max_{\epsilon} \mathcal{L}_{\text{Cross Ent.}} \left(\mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} [\tilde{f}_\theta(x + \epsilon + \delta)], y \right) \tag{2.9}$$

Where we use $\tilde{f}$ to represent the SoftMax-ed logit function. However, because in our case, the classifier under attack is *not* $\mathbb{E}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}[\tilde{f}_\theta(x + \epsilon + \delta)]$, but rather $\tilde{f}_\theta(x + \epsilon + \delta)$ itself, we instead considered the loss function:

$$\max_{\epsilon} \mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} \left[\mathcal{L}_{\text{Cross Ent.}} \left(\tilde{f}_\theta(x + \epsilon + \delta), y \right) \right] \tag{2.10}$$

Empirically, we find the choice of loss function to make very little difference: see Figures 2.14 and 2.15.

We also must consider how to correctly make the attacker "strategic": that is, how to allocate attack magnitude so as to attack most effectively while minimizing Wasserstein distance. This is more difficult than in the undefended case, because it is no longer true that for each sample x, we can identify the magnitude $\|CW(x, y; g) - x\|_2$ such that an attack of this magnitude is guaranteed to be successful, while a smaller attack is unsuccessful and hence is not attempted. Rather, for a given attack magnitude, there is instead a *probability of success*, over the distribution of δ.

In order to deal with this, we perform PGD at a range of attack magnitudes, specifically $E = \{i/8 | i \in \{1, ..., 16\}\}$. Let $PGD_e(x, y; g)$ be the result of the attack at magnitude $e \in E$. We then define the adaptive attacker as:

$$\text{Adv}_\gamma(x) := PGD_{e*}(x, y; g) \tag{2.11}$$

Where:

$$\begin{aligned} &e* := \max e \in E \text{ such that} \\ &\frac{\mathbb{E}_\delta\left[\mathcal{L}_{0/1}\left(\tilde{f}_\theta(PGD_e(x, y; g) + \delta), y\right)\right] - \mathbb{E}_\delta\left[\mathcal{L}_{0/1}\left(\tilde{f}_\theta(x + \delta), y\right)\right]}{e} > \gamma \end{aligned} \tag{2.12}$$

In other words, we use the largest attack such that the *increase in misclassification rate per unit attack magnitude* is above the threshold γ. If this is not the case for any $e \in E$, we elect not to attack, and set $\text{Adv}_\gamma(x) := x$. As was described in the main text for the baseline case, we sweep over a range of threshold values γ when reporting results. When evaluating the expectations in Equation 2.12, we use a sample of 100 noise instances. However, once $e*$ is identified, we then use a *different* sample of 100 noise instances per

training sample x when reporting the final accuracy: this is to de-correlate the attack generation of $\text{Adv}_\gamma(x)$ with the evaluation of the attack. (However, noise instances are kept constant over the sweep of γ). When reporting results (the upper bounds of the Wasserstein distances), we use $e*$ as an upper bound on $\|PGD_{e*}(x, y; g) - x\|_2$, rather than using $\|PGD_{e*}(x, y; g) - x\|_2$ directly. Also, we upper bound the population expectation of $e*$ (and therefore of $\|PGD_{e*}(x, y; g)-x\|_2$) for each γ with 99% confidence using the empirical expectation on the test set using a Hoeffding bound, using the fact that $0 \leq e* \leq \min(2, 1/\gamma)$.

Attack hyperparameters are taken from Salman et al. [39]: We use 20 attack steps, a step size of $e/10$, and use 128 noise instances when computing gradients. We evaluate using 10% of each dataset.

F Experiment Details for Section 2.6

As mentioned, for the certified models, we use the released pre-trained ResNet110 models from Cohen et al. [36] for CIFAR-10 and train ResNet20 models in a similar manner for SVHN, using the same level of Gaussian Noise for training and testing. For empirical results, we use the implementation of the ℓ_2 Carlini and Wagner [24] attack provided by the IBM ART package [90] with default parameters (except for batch size which we set at 256 to increase processing speed.)

We also tested an alternative attack, which is still strategic but does not require that we measure the Wasserstein distance empirically. In this attack, we define Adv'_γ, that if $\|CW(x, y; g) - x\|_2 \leq \gamma$ always returns $CW(x, y; g)$, and if $\|CW(x, y; g) -$

$x\|_2 > \gamma$, instead returns x with probability $1 - \frac{\gamma}{\|CW(x,y;g)-x\|_2}$. Note that in this case, the perturbation $\|\text{Adv}'_\gamma(x, y; g) - x\|_2$ is guaranteed to be less than or equal to γ in expectation for all x, so γ can be used as an upper bound on the Wasserstein distance. Results are shown in Figure 2.18.

G Function ψ for Different Distributions

For an isometric Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$,

$$\mathsf{TV}(\mathcal{N}(0, \sigma^2 I), \mathcal{N}(\theta, \sigma^2 I)) = \text{erf}(\|\theta\|_2 / 2\sqrt{2}\sigma).$$

Proof. Due to the isometric symmetry of the Gaussian distribution and the ℓ_2-norm, we may assume, without loss of generality, that $\mathcal{N}(\theta, \sigma^2 I)$ is obtained by shifting $\mathcal{N}(0, \sigma^2 I)$ only along the first dimension. Therefore, the total variation of the two distributions is equal to the difference in the probability of a normal random variable with variance σ^2 being less than $\|\theta\|_2/2$ and $-\|\theta\|_2/2$, i.e., $\Phi(\|\theta\|_2/2\sigma) - \Phi(-\|\theta\|_2/2\sigma)$ where Φ is the standard normal CDF.

$$\begin{aligned}
\mathsf{TV}(\mathcal{N}(0, \sigma^2 I), \mathcal{N}(\theta, \sigma^2 I)) &= \Phi(\|\theta\|_2/2\sigma) - \Phi(-\|\theta\|_2/2\sigma) \\
&= 2\Phi(\|\theta\|_2/2\sigma) - 1 \\
&= 2\left(\frac{1 + \text{erf}(\|\theta\|_2/2\sqrt{2}\sigma)}{2}\right) - 1 \\
&= \text{erf}(\|\theta\|_2/2\sqrt{2}\sigma).
\end{aligned}$$

□

For a uniform distribution $\mathcal{U}(\theta, b)$ between θ_i and $\theta_i + b$ in each dimension for $b \geq 0$ (as used for the SV shift transformations), $\mathsf{TV}(\mathcal{U}(0, b), \mathcal{U}(\theta, b)) \leq \|\theta\|_1 / b$. When $\|\theta\|_1$ is constrained, the volume of the overlap between $\mathcal{U}(0, b)$ and $\mathcal{U}(\theta, b)$ is minimized when the shift is only along one dimension.

H Additive Composability of Natural Transformations

In this section, we prove that the natural transformation CS, HS and SV defined in the paper all satisfy the additive composability property in condition (2.5).

Lemma 3. *The transformation* CS *satisfies the additive composability property, i.e.,* $\forall x \in M, \theta_1, \theta_2 \in \mathbb{R}^3$*,*

$$\mathsf{CS}(\mathsf{CS}(x, \theta_1), \theta_2) = \mathsf{CS}(x, \theta_1 + \theta_2).$$

Proof. Let $x = \{(r, g, b)_{ij}\}^{H \times W}$, $x' = \{(r', g', b')_{ij}\}^{H \times W} = \mathsf{CS}(x, \theta_1)$ and $x'' = \{(r'', g'', b'')_{ij}\}^{H \times W} = \mathsf{CS}(x', \theta_2)$. We need to show that $x'' = \mathsf{CS}(x, \theta_1 + \theta_2)$. Let $r_{\max}, g_{\max}$ and $b_{\max}$ be the maximum values of the red, green and blue channels respectively of x and $r'_{\max}, g'_{\max}$ and $b'_{\max}$ be the same for x'. From the definition of CS in Section 2.5.1, we have:

$$\begin{aligned} & r'_{ij} = \frac{2^{\theta_1^R} r_{ij}}{\mathsf{MAX}}, \quad g'_{ij} = \frac{2^{\theta_1^G} g_{ij}}{\mathsf{MAX}}, \quad b'_{ij} = \frac{2^{\theta_1^B} b_{ij}}{\mathsf{MAX}} \\ \text{and} \quad & r''_{ij} = \frac{2^{\theta_2^R} r'_{ij}}{\mathsf{MAX}'}, \quad g''_{ij} = \frac{2^{\theta_2^G} g'_{ij}}{\mathsf{MAX}'}, \quad b''_{ij} = \frac{2^{\theta_2^B} b'_{ij}}{\mathsf{MAX}'} \end{aligned}$$

where $\mathsf{MAX} = \max(2^{\theta_1^R} r_{\max}, 2^{\theta_1^G} g_{\max}, 2^{\theta_1^B} b_{\max})$ and $\mathsf{MAX}' = \max(2^{\theta_2^R} r'_{\max}, 2^{\theta_2^G} g'_{\max}, 2^{\theta_2^B} b'_{\max})$.

From the definition of $r'_{\max}$, we have:

$$r'_{\max} = \max r'_{ij} = \max \frac{2^{\theta_1^R} r_{ij}}{\mathsf{MAX}} = \frac{2^{\theta_1^R} \max r_{ij}}{\mathsf{MAX}} = \frac{2^{\theta_1^R} r_{\max}}{\mathsf{MAX}}.$$

Similarly,

$$g'_{\max} = \frac{2^{\theta_1^G} g_{\max}}{\mathsf{MAX}} \quad \text{and} \quad b'_{\max} = \frac{2^{\theta_1^B} b_{\max}}{\mathsf{MAX}}.$$

Therefore,

$$\mathsf{MAX}' = \frac{\max(2^{\theta_1^R+\theta_2^R} r_{\max}, 2^{\theta_1^G+\theta_2^G} g_{\max}, 2^{\theta_1^B+\theta_2^B} b_{\max})}{\mathsf{MAX}}.$$

Substituting r'_{ij} and MAX' in the expression for r''_{ij}, we get:

$$r''_{ij} = \frac{2^{\theta_2^R} 2^{\theta_1^R} r_{ij}}{\mathsf{MAX}'\mathsf{MAX}} = \frac{2^{\theta_1^R+\theta_2^R} r_{ij}}{\max(2^{\theta_1^R+\theta_2^R} r_{\max}, 2^{\theta_1^G+\theta_2^G} g_{\max}, 2^{\theta_1^B+\theta_2^B} b_{\max})}.$$

Similarly,

$$g''_{ij} = \frac{2^{\theta_1^G+\theta_2^G} g_{ij}}{\max(2^{\theta_1^R+\theta_2^R} r_{\max}, 2^{\theta_1^G+\theta_2^G} g_{\max}, 2^{\theta_1^B+\theta_2^B} b_{\max})}$$

and

$$b''_{ij} = \frac{2^{\theta_1^B+\theta_2^B} b_{ij}}{\max(2^{\theta_1^R+\theta_2^R} r_{\max}, 2^{\theta_1^G+\theta_2^G} g_{\max}, 2^{\theta_1^B+\theta_2^B} b_{\max})}.$$

Hence, $x'' = \mathsf{CS}(x, \theta_1 + \theta_2)$. □

Lemma 4. *The transformation* SV *satisfies the additive composability property, i.e.,* $\forall x \in M, \theta_1, \theta_2 \in \mathbb{R}^2_{\geq 0}$,

$$\mathsf{SV}(\mathsf{SV}(x, \theta_1), \theta_2) = \mathsf{SV}(x, \theta_1 + \theta_2).$$

Proof. Let $x = \{(h, s, v)_{ij}\}^{H \times W}$, $x' = \{(h, s', v')_{ij}\}^{H \times W} = \mathsf{SV}(x, \theta_1)$ and $x'' = \{(h, s'', v'')_{ij}\}^{H \times W} =$

$\mathsf{SV}(x', \theta_2)$ in HSV format. We need to show that $x'' = \mathsf{SV}(x, \theta_1+\theta_2)$. Let $s_{\text{mean}}, s_{\max}, v_{\text{mean}}$ and $v_{\max}$ be the means and maximums of the saturation and brightness values of x and $s'_{\text{mean}}, s'_{\max}, v'_{\text{mean}}$ and $v'_{\max}$ be the same for x'. From the definition of SV in Section 2.5.2, we have:

$$s'_{ij} = \frac{s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}}, \quad v'_{ij} = \frac{v_{ij} + (2^{\theta_1^V} - 1)v_{\text{mean}}}{\mathsf{MAX}}$$
$$\text{and} \quad s''_{ij} = \frac{s'_{ij} + (2^{\theta_2^S} - 1)s'_{\text{mean}}}{\mathsf{MAX}'}, \quad v''_{ij} = \frac{v'_{ij} + (2^{\theta_2^V} - 1)v'_{\text{mean}}}{\mathsf{MAX}'}$$

where $\mathsf{MAX} = \max(s_{\max}+(2^{\theta_1^S}-1)s_{\text{mean}}, v_{\max}+(2^{\theta_1^V}-1)v_{\text{mean}})$ and $\mathsf{MAX}' = \max(s'_{\max}+(2^{\theta_2^S} - 1)s'_{\text{mean}}, v'_{\max} + (2^{\theta_2^V} - 1)v'_{\text{mean}})$. From the definitions of s'_{mean} and $s'_{\max}$, we have:

$$s'_{\text{mean}} = \text{mean}\, s'_{ij} = \text{mean}\, \frac{s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}} = \frac{\text{mean}\, s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}} = \frac{2^{\theta_1^S} s_{\text{mean}}}{\mathsf{MAX}}$$
$$s'_{\max} = \max s'_{ij} = \max \frac{s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}} = \frac{\max s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}} = \frac{s_{\max} + (2^{\theta_1^S} - 1)s_{\text{mean}}}{\mathsf{MAX}}.$$

Similarly,

$$v'_{\text{mean}} = \frac{2^{\theta_1^V} v_{\text{mean}}}{\mathsf{MAX}} \quad \text{and} \quad v'_{\max} = \frac{v_{\max} + (2^{\theta_1^V} - 1)v_{\text{mean}}}{\mathsf{MAX}}.$$

Therefore,

$$\begin{aligned}
\mathsf{MAX}' &= \max(s'_{\max} + (2^{\theta_2^S} - 1)s'_{\text{mean}}, v'_{\max} + (2^{\theta_2^V} - 1)v'_{\text{mean}}) \\
&= \max(\frac{s_{\max} + (2^{\theta_1^S} - 1)s_{\text{mean}} + (2^{\theta_2^S} - 1)2^{\theta_1^S} s_{\text{mean}}}{\mathsf{MAX}}, v'_{\max} + (2^{\theta_2^V} - 1)v'_{\text{mean}}) \\
&= \max(\frac{s_{\max} + (2^{\theta_1^S+\theta_2^S} - 1)s_{\text{mean}}}{\mathsf{MAX}}, v'_{\max} + (2^{\theta_2^V} - 1)v'_{\text{mean}}) \\
&= \max(s_{\max} + (2^{\theta_1^S+\theta_2^S} - 1)s_{\text{mean}}, v_{\max} + (2^{\theta_1^V} - 1)v_{\text{mean}} + (2^{\theta_2^V} - 1)2^{\theta_1^V} v_{\text{mean}})/\mathsf{MAX}
\end{aligned}$$

$$= \max(s_{\max} + (2^{\theta_1^S + \theta_2^S} - 1)s_{\text{mean}}, v_{\max} + (2^{\theta_1^V + \theta_2^V} - 1)v_{\text{mean}})/\mathsf{MAX}.$$

Substituting $s'_{ij}, s'_{\text{mean}}$ and MAX' in the expression for s''_{ij}, we get:

$$\begin{aligned} s''_{ij} &= \frac{s_{ij} + (2^{\theta_1^S} - 1)s_{\text{mean}} + (2^{\theta_2^S} - 1)2^{\theta_1^S} s_{\text{mean}}}{\mathsf{MAX}'\mathsf{MAX}} \\ &= \frac{s_{ij} + (2^{\theta_1^S + \theta_2^S} - 1)s_{\text{mean}}}{\max(s_{\max} + (2^{\theta_1^S + \theta_2^S} - 1)s_{\text{mean}}, v_{\max} + (2^{\theta_1^V + \theta_2^V} - 1)v_{\text{mean}})}. \end{aligned}$$

Similarly,

$$v''_{ij} = \frac{v_{ij} + (2^{\theta_1^V + \theta_2^V} - 1)v_{\text{mean}}}{\max(s_{\max} + (2^{\theta_1^S + \theta_2^S} - 1)s_{\text{mean}}, v_{\max} + (2^{\theta_1^V + \theta_2^V} - 1)v_{\text{mean}})}.$$

Hence, $x'' = \mathsf{SV}(x, \theta_1 + \theta_2)$. □

I Details for Plots in Figure 2.2

The distribution shifts used to evaluate the empirical performance of the base models in Figure 2.2 have been generated by first sampling an image x from the original distribution $\mathcal{D}$ and then randomly transforming it images from the original distribution by adding a noise in the corresponding transformation space. The Wasserstein bound of these shifts can be calculated by computing the expected perturbation size of the smoothing distribution. For example, the expected ℓ_2-norm of a 3-dimensional Gaussian vector is given by $2\sqrt{2}\sigma/\sqrt{\pi}$ and expected ℓ_1-norm a 2-dimensional vector sampled uniformly from $[0, b]^2$ is b.

The training and smoothing noise levels used for color shift, hue shift and SV shift

are (0.8, 10.0), ($180°$, $180°$) and (8.0, 12.0) respectively.

J Hue Shift

Any RGB image can be alternatively represented in the HSV image format by mapping the (r, g, b) tuple for each pixel to a point (h, s, v) in a cylindrical coordinate system where the values h, s and v represent the hue, saturation and brightness (value) of the pixel. The mapping from the RGB coordinate to the HSV coordinate takes the $[0, 1]^3$ color cube and transforms it into a cylinder of unit radius and height. The hue values are represented as angles in $[0, 2\pi)$ and the saturation and brightness values are in $[0, 1]$. Define a hue shift of an $H \times W$ sized image x by an angle $\theta \in [-\pi, \pi]$ in the HSV space that rotates each hue value by an angle θ and wraps it around to the $[0, 2\pi)$ range. In appendix J, we show that the certified accuracy under hue shifts does not depend on the Wasserstein distance of the shifted distribution and report the certified accuracies obtained by various base models trained under different noise levels.

Define a hue shift of an $H \times W$ sized image x by an angle $\theta \in [-\pi, \pi]$ in the HSV space as:

$$\mathsf{HS}(x, \theta) = \left\{ (w(h + \theta), s, v)_{ij} \right\}^{H \times W}$$
$$\text{where} \quad w(x) = x - 2\pi \left\lfloor \frac{x}{2\pi} \right\rfloor$$

which rotates each hue value by an angle θ and wraps it around to the $[0, 2\pi)$ range. It is easy to show that this transformation satisfies additive composability in condition (2.5). The Wasserstein distance is defined using the corresponding distance function d_{HS} by

taking the absolute value of the hue shift $|\theta|$.

Lemma 5. *The transformation* HS *satisfies the additive composability property, i.e.,* $\forall x \in M, \theta_1, \theta_2 \in [-\pi, \pi]$,

$$\mathsf{HS}(\mathsf{HS}(x, \theta_1), \theta_2) = \mathsf{HS}(x, \theta_1 + \theta_2).$$

Proof. Let h be the hue value of the (i, j)th pixel of the image x. Since the transformation only affects the hue values, we ignore the other coordinates. The hue value after the transformation $\mathsf{HS}(\mathsf{HS}(x, \theta_1), \theta_2)$ is given by

$$w(w(h + \theta_1) + \theta_2) = w\left(h + \theta_1 - 2\pi \left\lfloor \frac{h + \theta_1}{2\pi} \right\rfloor + \theta_2\right)$$

□

Define a smoothing distribution that applies a random hue rotation δ sampled uniformly from the range $[-\pi, \pi]$. Since HS wraps the hue values around in the interval, the distributions of $h + \delta$ and $(h + \theta) + \delta$ for two hue values shifted by an angle θ are both uniform in $[0, 2\pi]$. Thus, the smoothing distribution for two hue shifted images is the same which implies that $\psi(d(x_1, x_2)) = 0$ whenever $d(x_1, x_2)$ is finite. Hence, from Theorem 1, we have $\mathbb{E}_{(x_2,y_2)\sim\tilde{\mathcal{D}}}[\bar{h}(x_2, y_2)] \geq \mathbb{E}_{(x_1,y_1)\sim\mathcal{D}}[\bar{h}(x_1, y_1)]$ for hue shifts. Since, the certified accuracy remains constant with respect to the Wasserstein distance of the shift, we just plot the certified accuracies obtained by various base models trained under different noise levels in Figure 2.19. We plot the certified accuracies obtained by various models trained using random hue rotations picked uniformly from the range $[-\beta, \beta]$ for different values of the maximum angle β in the range. The certified accuracy roughly increases with the

training noise achieving a maximum of 87.9% for a max angle $\beta = 180^\circ$ for the training noise level.

K Random Channel Selection

Consider a smoothing distribution that randomly picks one of the RGB channels with equal probability, scales it so that the maximum pixel value in that channel is one and sets all the other channels to zero. This smoothing distribution is invariant to the color shift transformation CS and thus, satisfies $\psi(d_{\mathcal{T}}(x_1, x_2)) = 0$ whenever $d_{\mathcal{T}}(x_1, x_2)$ is finite. Therefore, from Theorem 1, we have $\mathbb{E}_{z\sim\tilde{\mathcal{D}}}[\bar{h}(z)] \geq \mathbb{E}_{x\sim\mathcal{D}}[\bar{h}(x)]$ under this smoothing distribution for all Wasserstein bounds ϵ with respect to d_{CS}. Figure 2.20 plots the certified accuracies, using random channel selection for smoothing, achieved by models trained using Gaussian distributions of varying noise levels in the transformation space. The certified accuracy roughly increases with the training noise achieving a maximum of 87.1% for a training noise of 0.8.

L Experimental details for Section 2.7

Our experimental setting is adapted from the "sample-wise perturbation" CIFAR-10 experiments in Huang et al. [78]: hyperparameters are the same as in that work unless otherwise stated. For background, Huang et al. [78] creates an unlearnable dataset by performing the following "bi-level" minimization, to simultaneously train a proxy classifier model and create unlearnable examples:

$$\min_{\theta} \min_{(\epsilon_1,\ldots,\epsilon_n)} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i + \epsilon_i), y_i) \tag{2.13}$$

In other words, in contrast with standard training, both the samples and the proxy classifier are optimized to decrease the loss. New classifiers trained on the resulting samples fail to generalize to unperturbed samples. In the experiments, as in Huang et al. [78], the inner minimization over perturbations is performed for 20 steps over the entire dataset, for every one batch update step of the outer minimization. Training stops when training accuracy reaches a threshold value of 99%.

We now detail differences in experimental setup from Huang et al. [78]:

M Adaptation to ℓ_2 attack setting

After each optimization step, we project ϵ_i's into an ℓ_2 ball (of radius given by the Wasserstein bound ϵ) rather than an ℓ_∞ ball. We also use an ℓ_2 PGD step:

$$\epsilon_i' = \epsilon_i + \tau \frac{\nabla_{\epsilon_i} \mathcal{L}(\cdot)}{\|\nabla_{\epsilon_i} \mathcal{L}(\cdot)\|_2} \tag{2.14}$$

Step size τ was set as 0.1 times the total ℓ_2 ϵ bound.

N Adaptation to offline setting

As discussed in the test, we modify the algorithm such that the simultaneous training of the proxy model and generation of perturbations does not introduce statistical dependencies between perturbed training samples. This is especially important because, if the victim later makes a train-validation split, this would introduce statistical dependencies between training and validation samples, making it hard to generalize certificates to a test set.

To avoid this, we construct four data splits:

- Test set (10000 samples): The original CIFAR-10 test set. Never perturbed, only used in final model evaluation.

- Proxy training set (20000 samples): Used for the optimization of the proxy classifier model parameters θ in Equation 2.13 and discarded afterward.

- Training set (20000 samples): Perturbed using one round of the the standard 20 steps of the inner optimization of Equation 2.13, while keeping θ fixed.

- Validation set (10000 samples): Perturbed using the same method as the "Training set."

The victim model is trained on the "Training Set" and evaluated on the "Validation set" and "Test set". We also tested on the clean (unperturbed) version of the validation set.

O Adaptive attack setting

When testing our smoothing algorithm, we tested two types of attacks:

- Non-adaptive attack: the proxy model is trained and perturbations are generated using undefended models without smoothing: only the victim policy applies smoothing noise during training and evaluation.

- Adaptive attack: In the minimization of Equation 2.13, the loss term $\mathcal{L}(f_\theta(x_i + \epsilon_i), y_i)$ is replaced by the expectation:

$$\mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} \mathcal{L}(f_\theta(x_i + \epsilon_i + \delta), y_i) \tag{2.15}$$

In other words, this models the expectation of a *smoothed* model, like the victim classifier. This smoothed optimization is used in both the proxy model training, as well as the generation of the training and validation sets. Following Salman et al. [39], which proposed a similar adaptive attack for sample-wise smoothed classifiers we approximate the expectation using a small number of random perturbations, which are held fixed for the 20 steps of the inner optimization. In our experiments, we use 8 samples for approximation. Because, at large smoothing noises, this makes the attack much less effective, we cut off training after 20 steps of the outer maximization, rather than relying on the accuracy to reach 99%. (the maximum number of steps required to converge we observed for the non-adaptive attack was 15).

P Results

Complete experimental results are presented in Figure 2.21. All results are means of 5 independent runs, and error bars represent standard errors of the means.

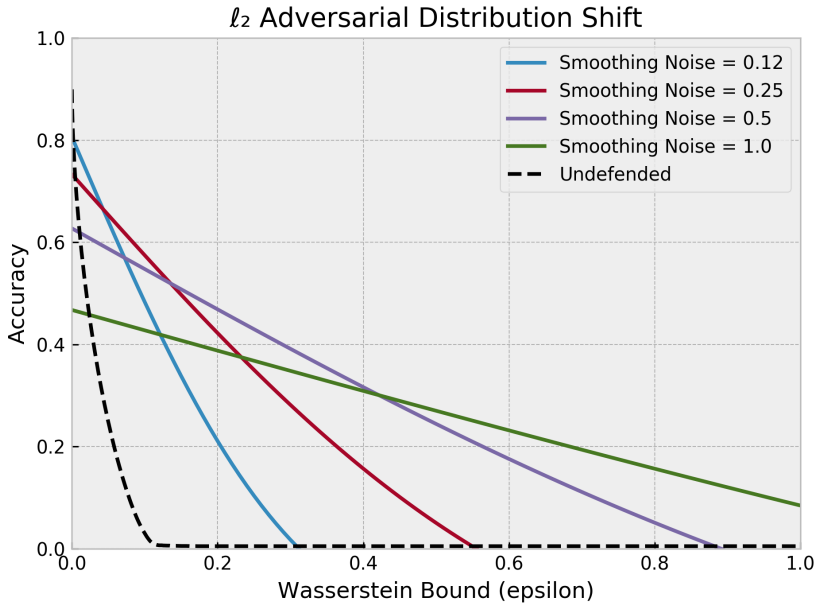


Figure 2.11: CIFAR-10.

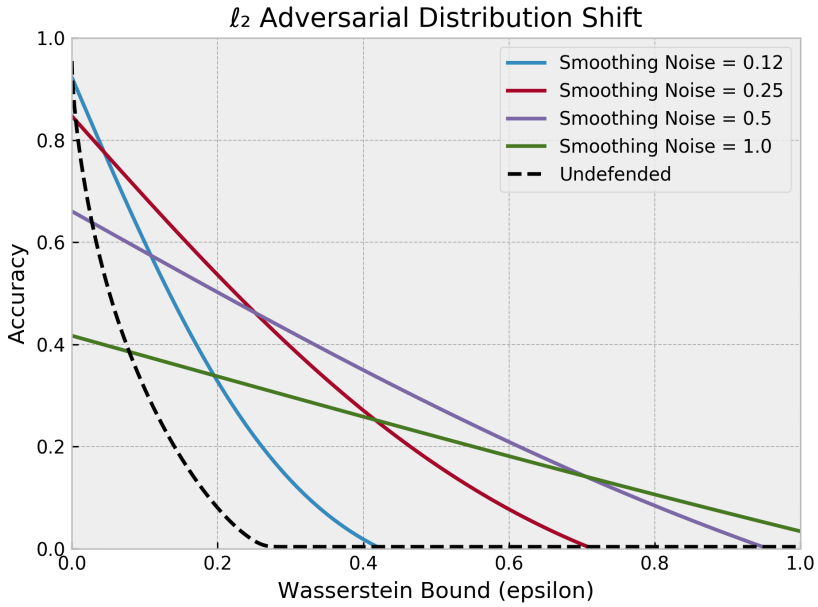


Figure 2.12: SVHN

Figure 2.13: Distributional certificates against adversarial attacks on (a) CIFAR-10 and (b) SVHN. The solid lines represent certified accuracy of the robust models and the dashed lines represent the adversarial accuracy of undefended models.

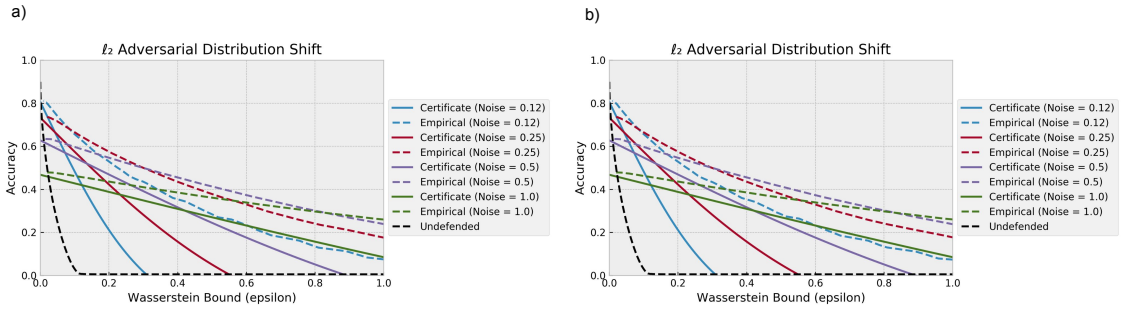


Figure 2.14: Adversarial attack on distributionally-smoothed classifiers, for CIFAR-10. For smoothed classifiers, we use the PGD attack described in is section; see Section 2.6 for details on the baseline. The dashed lines represent the empirical performance of the smoothed model for different noise levels. In plot (a), we use the loss function in Equation 2.9, while in (b) we use Equation 2.10.

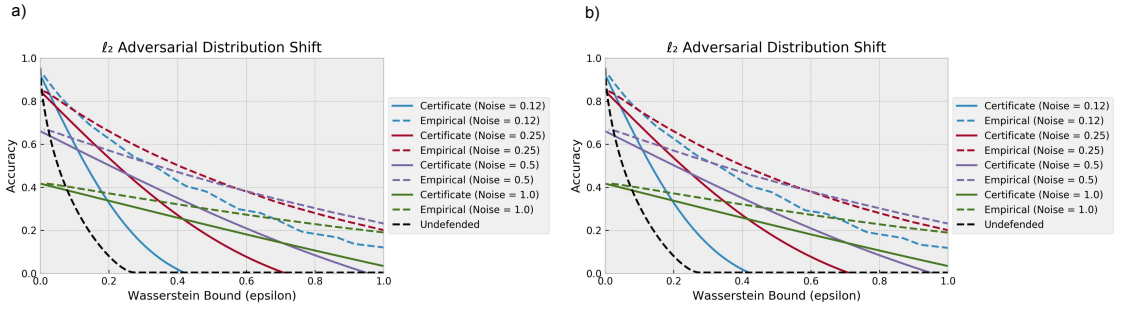


Figure 2.15: Adversarial attack on distributionally-smoothed classifiers, for SVHN. For smoothed classifiers, we use the PGD attack described in is section; see Section 2.6 for details on the baseline. The dashed lines represent the empirical performance of the smoothed model for different noise levels. In plot (a), we use the loss function in Equation 2.9, while in (b) we use Equation 2.10.

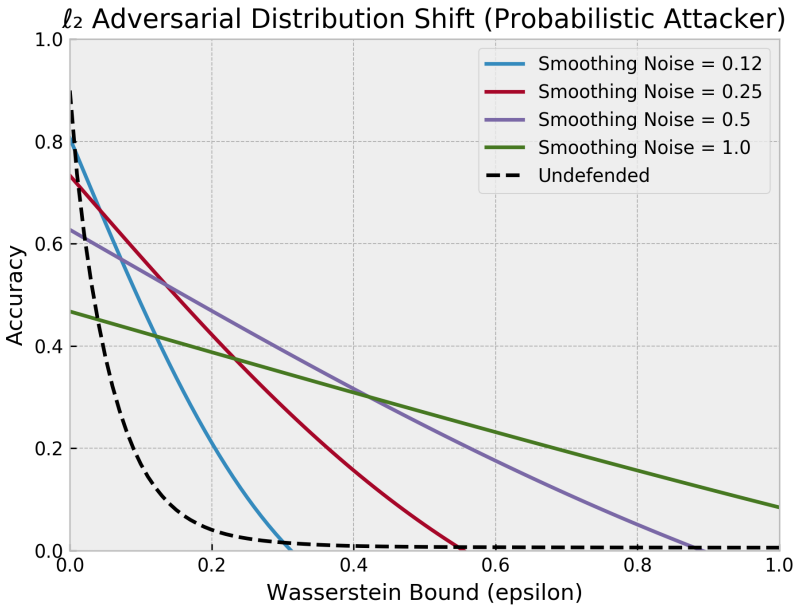


Figure 2.16: CIFAR-10

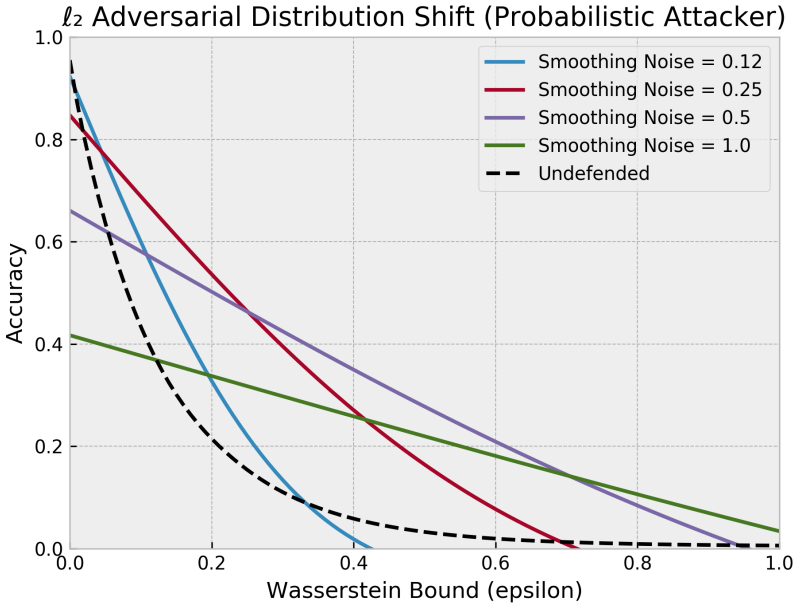


Figure 2.17: SVHN

Figure 2.18: Certified robustness to ℓ_2 Wasserstein distributional attacks. The undefended baseline baseline is here attacked using the alternative attack formulation Adv' described in Section F.

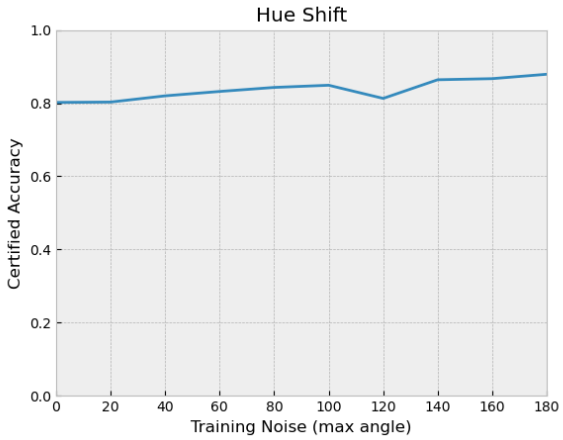


Figure 2.19: Certified accuracy under hue shift for different levels of training noise. Since, the certified accuracy remains constant with respect to the Wasserstein distance (ϵ) of the shifted distribution, we plot the certified accuracy of models trained with different noise levels β.

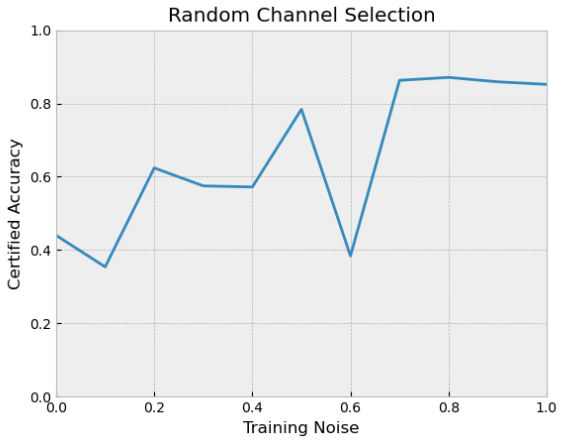


Figure 2.20: Certified robustness against color shift using random channel selection as the smoothing distribution. Since, the certified accuracy remains constant with respect to the Wasserstein distance (ϵ) of the shifted distribution, we plot the certified accuracy of models trained with various levels of Gaussian noise in the transformation space.

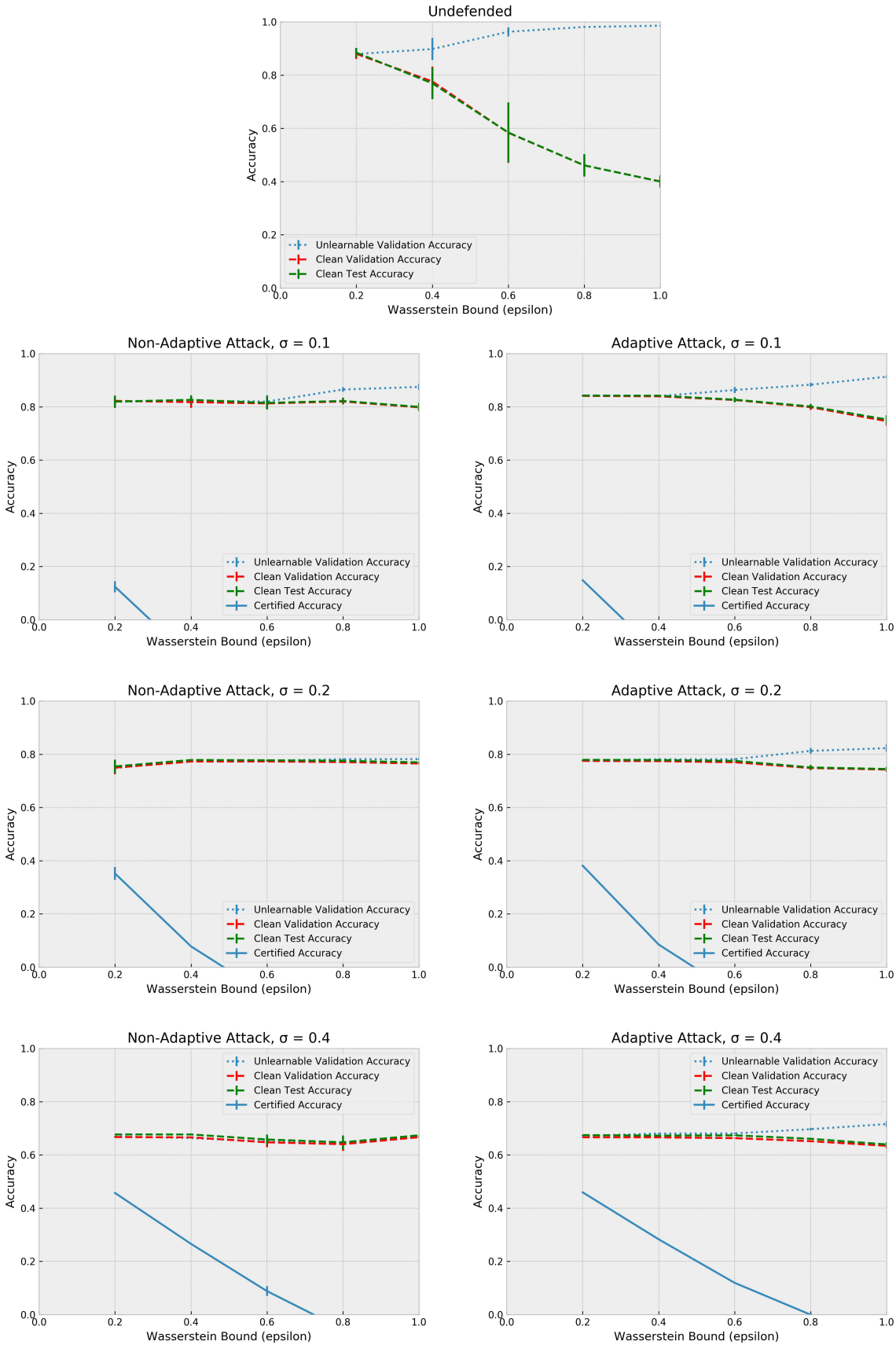


Figure 2.21: Complete Experimental results for unlearnability experiments.

Chapter 3: Policy Smoothing

3.1 Introduction

Deep neural networks (DNNs) have been widely employed for reinforcement learning (RL) problems as they enable the learning of policies directly from raw sensory inputs, like images, with minimal intervention from humans. From achieving super-human level performance in video-games [91, 92, 93], Chess [94] and Go [95] to carrying out complex real-world tasks, such as controlling a robot [96] and driving a vehicle [97], deep-learning based algorithms have not only established the state of the art, but also become more effortless to train. However, DNNs have been shown to be susceptible to tiny malicious perturbations of the input designed to completely alter their predictions [1, 2, 3]. In the RL setting, an attacker may either directly corrupt the observations of an RL agent [6, 7, 8] or act adversarially in the environment [5] to significantly degrade the performance of the victim agent. Most of the adversarial defense literature has focused mainly on classification tasks [18, 19, 20, 21, 22, 23, 98]. In this paper, we study a defense procedure for RL problems that is provably robust against norm-bounded adversarial perturbations of the observations of the victim agent.

Problem setup. A reinforcement learning task is commonly described as a game between an agent and an environment characterized by the Markov Decision Process

(MDP) $M = (S, A, T, R, \gamma)$, where S is a set of states, A is a set of actions, T is the transition probability function, R is the one-step reward function and $\gamma \in [0, 1]$ is the discount factor. However, as described in Section 3.3, our analysis applies to an even more general setting than MDPs. At each time-step t, the agent makes an observation $o_t = o(s_t) \in \mathbb{R}^d$ which is a probabilistic function of the current state of the environment, picks an action $a_t \in A$ and receives an immediate reward $R_t = R(s_t, a_t)$. We define an adversary as an entity that can corrupt the agent's observations of the environment by augmenting them with a perturbation ϵ_t at each time-step t which can depend on the states, actions, observations, etc., generated so far. We use $\epsilon = (\epsilon_1, \epsilon_2, \ldots)$ to denote the entire sequence of adversarial perturbations. The goal of the adversary is to minimize the total reward obtained by the agent policy π while keeping the overall ℓ_2-norm of the perturbation within a budget B. Formally, the adversary seeks to optimize the following objective:

$$\min_{\epsilon} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right], \text{ where } R_t = R(s_t, a_t), a_t \sim \pi(\cdot | o(s_t) + \epsilon_t)$$
$$\text{s.t. } \|(\epsilon_1, \epsilon_2, \ldots)\|_2 = \sqrt{\sum_{t=0}^{\infty} \|\epsilon_t\|_2^2} \leq B.$$

Note that the size of the perturbation ϵ_t in each time-step t need not be the same and the adversary may choose to distribute the budget B over different time-steps in a way that allows it to produce a stronger attack. Also, our formulation accounts for cases when the agent may only partially observe the state of the environment, making M a Partially Observable Markov Decision Process (POMDP).

Objective. Our goal in provably robust RL is to design a policy π such that the total reward in the presence of a norm-bounded adversary is guaranteed to remain above a certain threshold, i.e.,

$$\min_{\epsilon} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right] \geq \underline{R}, \text{ s.t. } \|\epsilon\|_2 \leq B. \tag{3.1}$$

In other words, no norm-bounded adversary can lower the expected total reward of the policy π below a certain threshold. In our discussion, we restrict out focus to finite-step games that end after t time-steps. This is a reasonable approximation for infinite games with $\gamma < 1$, as for a sufficiently large t, γ^t becomes negligibly small. For games where $\gamma = 1$, R_t must become sufficiently small after a finite number of steps to keep the total reward finite.

Step-wise vs. episodic certificates. Previous works on robust RL have sought to certify the behaviour of the policy function at *each* time-step of an episode, e.g., the output of a Deep Q-Network [99] and the action taken for a given state [100]. Ensuring that the behaviour of the policy remains unchanged in each step can also certify that the final total reward remains the same under attack. However, if the per-step guarantee fails at even one of the intermediate steps, the certificate on the total reward becomes vacuous or impractical to compute (as noted in Appendix E of [100]). Our approach gets around this issue by directly certifying the final total reward for the entire episode *without* requiring the policy to be provably robust at each intermediate step. Also, the threat-model we consider is more general as we allow the adversary to choose the size of the perturbation

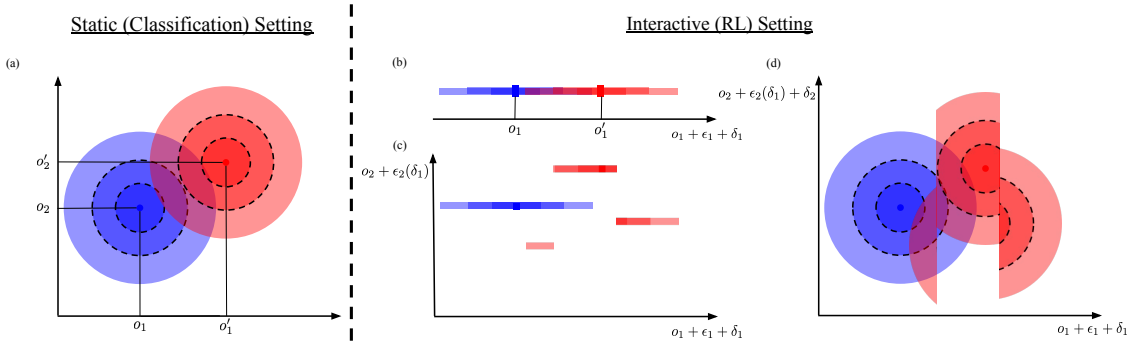


Figure 3.1: The standard [36] smoothing-based robustness certificate relies on the clean and the adversarial distributions being isometric Gaussians (panel a). However, adding noise to sequential observations in an RL setting (panels b-d) *does not* result in an isometric Gaussian distribution over the space of observations. In all figures, the distributions associated with clean and adversarially-perturbed values are shown in blue and red, respectively.

for each time-step. Thus, our method can defend against more sophisticated attacks that focus more on states that are crucial for the victim agent's performance.

Technical contributions. In this paper, we study a defense procedure based on "randomized smoothing" [36, 37, 38, 39] since at least in "static" settings, its robustness guarantee scales up to high-dimensional problems and does not need to make stringent assumptions about the model. We ask: *can we utilize the benefits of randomized smoothing to make a general high-dimensional RL policy provably robust against adversarial attacks?* The answer to this question turns out to be non-trivial as the adaptive nature of the adversary in the RL setting makes it difficult to apply certificates from the static setting. For example, the ℓ_2-certificate by [36] critically relies on the clean and adversarial distributions being isometric Gaussians (Figure 3.1-a). However, in the RL setting, the adversarial perturbation in one step might depend on states, actions, observations, etc., of the previous steps, which could in turn depend on the random Gaussian noise samples added to the observations in these steps. Thus, the resulting adversarial distribution need not be isometric as in the static setting (Figure 3.1-(b-d)). For more details on this example, see Appendix B.

Our main theoretical contribution is to prove an *adaptive version* of the Neyman-Pearson lemma [101] to produce robustness guarantees for RL. We emphasize that this is *not* a straightforward extension (refer to Appendix D, E and F for the entire proof). To prove this fundamental result, we first eliminate the effect of randomization in the adversary (Lemma 6) by converting a general adversary to one where the perturbation at each time-step is a deterministic function of the previous states, actions, observations, etc., and showing that the modified adversary is as strong as the general one. Then, we prove the adaptive Neyman-Pearson lemma where we show that, in the worst-case, the deterministic adversary can be converted to one that uses up the entire budget B in the first coordinate of the perturbation in the first time-step (Lemma 8). Finally, we derive the robustness guarantee under an isometric Gaussian smoothing distribution (Theorem 2). In section A, we establish the *tightness* of our certificates by constructing the worst-case environment-policy pair which attains our derived bounds. More formally, out of all the environment-policy pairs that achieve a certain total reward with probability p, we show a worst-case environment-policy pair and a corresponding adversary such that the probability of achieving the same reward under the presence of the adversary is minimum. A discussion on the Neyman-Pearson lemma in the context of randomized smoothing is available in Appendix C.

Building on these theoretical results, we propose **Policy Smoothing**, a simple model-agnostic randomized-smoothing based technique that can provide certified robustness without increasing the computational complexity of the agent's policy. Our main contribution is to show that by augmenting the policy's input by a random smoothing noise, we can achieve provable robustness guarantees on the total reward under a norm-bounded

adversarial attack (Section 3.4.2). Policy Smoothing does not need to make assumptions about the agent's policy function and is also oblivious to the workings of RL environment. Thus, this method can be applied to any RL setting without having to make restrictive assumptions on the environment or the agent. In section 3.3, we model the entire adversarial RL process under Policy Smoothing as a sequence of interactions between a system **A**, which encapsulates the RL environment and the agent, and a system **B**, which captures the addition of the adversarial perturbation and the smoothing noise to the observations. Our theoretical results do not require these systems to be Markovian and can thus have potential applications in real-time decision-making processes that do not necessarily satisfy the Markov property.

Empirical Results. We use four standard Reinforcement Learning benchmark tasks to evaluate the effectiveness of our defense and the significance of our theoretical results: the Atari games 'Pong' and 'Freeway' [91] and the classical 'Cartpole' and 'Mountain Car' control environments [102, 103] – see Figure 3.3. We find that our method provides highly nontrivial certificates. In particular, on at least two of the tasks, 'Pong' and 'cartpole', the *provable lower bounds* on the average performances of the defended agents, against any adversary, exceed the observed average performances of undefended agents under a practical attack.

3.2 Prior Work

Adversarial RL. Adversarial attacks on RL systems have been extensively studied in recent years. DNN-based policies have been attacked by either directly corrupting

their inputs [6, 7, 8] or by making adversarial changes in the environment [5]. Empirical defenses based on adversarial training, whereby the dynamics of the RL system is augmented with adversarial noise, have produced good results in practice [104, 105]. [106] propose training policies together with a learned adversary in an online alternating fashion to achieve robustness to perturbations of the agent's observations.

Robust RL. Prior work by [99] has proposed a 'certified' defense against adversarial attacks to observations in deep reinforcement learning, particularly for Deep Q-Network agents. However, that work essentially only guarantees the stability of the *network approximated Q-value* at each time-step of an episode. By contrast, our method provides a bound on the expected *true reward* of the agent under any norm-bounded adversarial attack.

Zhang et al. [100] certify that the action in each time-step remains unchanged under an adversarial perturbation of fixed budget for every time-step. This can guarantee that the final total reward obtained by the robust policy remains the same under attack. However, this approach would not be able to yield any robustness certificate if even one of the intermediate actions changed under attack. Our approach gets around this difficulty by directly certifying the total reward, letting some of the intermediate actions of the robust policy to potentially change under attack. For instance, consider an RL agent playing Atari Pong. The actions taken by the agent when the ball is close to and approaching the paddle are significantly more important than the ones when the ball is far away or retreating from the paddle. By allowing some of the intermediate actions to potentially change, our approach can certify for larger adversarial budgets and provide a more fine-grained control over the desired total-reward threshold. Moreover, we study a more general threat model where the adversary may allocate different attack budgets for each

time-step focusing more on the steps that are crucial for the agent's performance, e.g., attacking a Pong agent when the ball is close to the paddle.

Provable Robustness in Static Settings: Notable provable robustness methods in static settings are based on interval-bound propagation [32, 33, 34, 35], curvature bounds [27, 28, 29, 30, 31, 107], randomized smoothing [36, 37, 38, 39, 67], etc. Certified robustness has also been extended to problems with structured outputs such as images and sets [108]. Focusing on Gaussian smoothing, Cohen et al. [36] showed that if a classifier outputs a class with some probability under an isometric Gaussian noise around an input point, then it will output that class with high probability at any perturbation of the input within a particular ℓ_2 distance. Kumar et al. [51] showed how to certify the expectation of softmax scores of a neural network under Gaussian smoothing by using distributional information about the scores.

3.3 Preliminaries and Notations

We model the finite-step adversarial RL framework as a t-round communication between two systems $\mathbf{A}$ and $\mathbf{B}$ (Figure 3.2). System $\mathbf{A}$ represents the RL game. It contains the environment M and the agent, and when run independently, simulates the interactions between the two for some given policy π. At each time-step i, it generates a *token* τ_i from some set $\mathcal{T}$, which is a tuple of the current state s_i and its observation o_i, the action a_{i-1} in the previous step (and potentially some other objects that we ignore in this discussion), i.e., $\tau_i = (s_i, a_{i-1}, o_i, \ldots) \in S \times A \times \mathbb{R}^d \times \ldots = \mathcal{T}$. For the first step, replace the action in τ_1 with some dummy element $*$ from the action space A. System $\mathbf{B}$

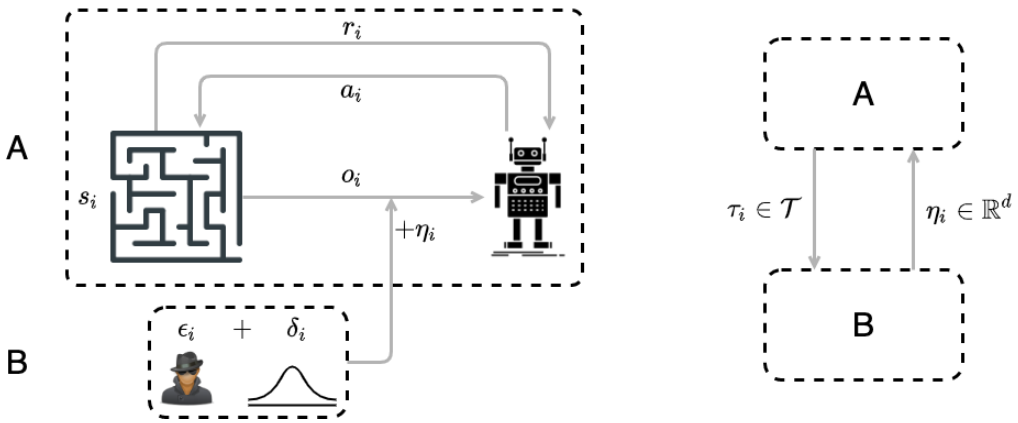


Figure 3.2: Adversarial robustness framework.

comprises of the adversary and the smoothing distribution which generate an adversarial perturbation ϵ_i and a smoothing noise vector δ_i, respectively, at each time-step i, the sum of which is denoted by an *offset* $\eta_i = \epsilon_i + \delta_i \in \mathbb{R}^d$.

When both systems are run together in an interactive fashion, in each round i, system $\mathbf{A}$ generates τ_i as a probabilistic function of $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_{i-1}, \eta_{i-1}$, i.e., $\tau_i : (\mathcal{T} \times \mathbb{R}^d)^{i-1} \to \Delta(\mathcal{T})$. τ_1 is sampled from a fixed distribution. It passes τ_i to $\mathbf{B}$, which generates ϵ_i as a probabilistic function of $\{\tau_j, \eta_j\}_{j=1}^{i-1}$ and τ_i, i.e., $\epsilon_i : (\mathcal{T} \times \mathbb{R}^d)^{i-1} \times \mathcal{T} \to \Delta(\mathbb{R}^d)$ and adds a noise vector δ_i sampled independently from the smoothing distribution to obtain η_i. It then passes η_i to $\mathbf{A}$ for the next round. After running for t steps, a deterministic or random 0/1-function h is computed over all the tokens and offsets generated. We are interested in bounding the probability with which h outputs 1 as a function of the adversarial budget B. In the RL setting, h could be a function indicating whether the total reward is above a certain threshold or not.

3.4 Provably Robust RL

3.4.1 Adaptive Neyman-Pearson Lemma

Let X be the random variable representing the tuple $z = (\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_t, \eta_t) \in (\mathcal{T} \times \mathbb{R}^d)^t$ when there is no adversary, i.e., $\epsilon_i = 0$ and $\eta_i = \delta_i$ is sampled directly form the smoothing distribution $\mathcal{P}$. Let Y be the random variable representing the same tuple in the presence of a general adversary ϵ satisfying $\|\epsilon\|_2 \leq B$. Thus, if $h(X) = 1$ with some probability p, we are interested in deriving a lower-bound on the probability of $h(Y) = 1$ as a function of p and B. Let us now define a deterministic adversary ϵ^{dt} for which the adversarial perturbation at each step is a deterministic function of the tokens and offsets of the previous steps and the token generated in the current step. i.e., $\epsilon_i^{dt} : (\mathcal{T} \times \mathbb{R}^d)^{i-1} \times \mathcal{T} \to \mathbb{R}^d$. Let Y^{dt} be its corresponding random variable. Then, we have the following lemma that converts a probabilistic adversary into a deterministic one.

Lemma 6 (Reduction to Deterministic Adversaries). *For any general adversary ϵ and an $\Gamma \subseteq (\mathcal{T} \times \mathbb{R}^d)^t$, there exists a deterministic adversary ϵ^{dt} such that,*

$$\mathbb{P}[Y^{dt} \in \Gamma] \leq \mathbb{P}[Y \in \Gamma],$$

where Y^{dt} is the random variable for the distribution defined by the adversary ϵ^{dt}.

This lemma says that for any adversary (deterministic or random) and a subset Γ of the space of z, there exists a deterministic adversary which assigns a lower probability to Γ than the general adversary. In the RL setting, this means that the probability with which

a smoothed policy achieves a certain reward value under a general adversary is lower-bounded by the probability of the same under a deterministic adversary. The intuition behind this lemma is that out of all the possible values that the internal randomness of the adversary may assume, there exists a sequence of values that assigns the minimum probability to Γ (over the randomness of the environment, policy, smoothing noise, etc.). We defer the proof to the appendix.

Next, we formulate an adaptive version of the Neyman-Pearson lemma for the case when the smoothing distribution $\mathcal{P}$ is an isometric Gaussian $\mathcal{N}(0, \sigma^2 I)$. If we applied the classical Neyman-Pearson lemma on the distributions of X and Y^{dt}, it will give us a characterization of the worst-case 0/1 function among the class of functions that achieve a certain probability p of being 1 under the distribution of X that has the minimum probability of being 1 under Y^{dt}. Let μ_X and $\mu_{Y^{dt}}$ be the probability density function of X and Y^{dt}, respectively.

Lemma 7 (Neyman-Pearson Lemma, 1933). *If* $\Gamma_{Y^{dt}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{dt}}(z) \leq q\mu_X(z)\}$ *for some* $q \geq 0$ *and* $\mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in \Gamma_{Y^{dt}}]$, *then* $\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}]$.

For an arbitrary element h in the class of functions $H_p = \{h \mid \mathbb{P}[h(X) = 1] \geq p\}$, construct the set $\Gamma_{Y^{dt}}$ for an appropriate value of q for which $\mathbb{P}[X \in \Gamma_{Y^{dt}}] = p$. Now, consider a function h' which is 1 if its input comes from $\Gamma_{Y^{dt}}$ and 0 otherwise. Then, the above lemma says that the function h' has the minimum probability of being 1 under Y^{dt}, i.e.,

$$h' = \operatorname*{argmin}_{h \in H_p} \mathbb{P}[h(Y^{dt}) = 1].$$

This gives us the worst-case function that achieves the minimum probability under an adversarial distribution. However, in the adaptive setting, $\Gamma_{Y^{dt}}$ could be a very complicated set and obtaining an expression for $\mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}]$ might be difficult. To simplify our analysis, we construct a *structured* deterministic adversary ϵ^{st} which exhausts its entire budget in the first coordinate of the first perturbation vector, i.e., $\epsilon_1^{st} = (B, 0, \ldots, 0)$ and $\epsilon_i^{st} = (0, 0, \ldots, 0)$ for $i > 1$. Let Y^{st} be the corresponding random variable and $\mu_{Y^{st}}$ its density function. We formulate the following adaptive version of the Neyman-Pearson lemma:

Lemma 8 (Adaptive Neyman-Pearson Lemma). *If* $\Gamma_{Y^{st}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{st}}(z) \leq q\mu_X(z)\}$ *for some* $q \geq 0$ *and* $\mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in \Gamma_{Y^{st}}]$*, then* $\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}]$.

The key difference from the classical version is that the worst-case set we construct in this lemma is for the structured adversary and the final inequality relates the probability of h outputting 1 under the adaptive adversary to the probability that the structured adversary assigns to the worst-case set. It says that for the appropriate value of q for which $\mathbb{P}[X \in \Gamma_{Y^{st}}] = p$, any function $h \in H_p$ outputs 1 with at least the probability that Y^{st} assigns to $\Gamma_{Y^{st}}$. It shows that over all possible functions in H_p and over all possible adversaries ϵ, the indicator function $\mathbf{1}_{z \in \Gamma_{Y^{st}}}$ and the structured adversary capture the worst-case scenario where probability of h being 1 under the adversarial distribution is the minimum. Since both Y^{st} and X are just isometric Gaussian distribution with the same variance σ^2 centered at different points on the first coordinate of η_1, the set $\Gamma_{Y^{st}}$ is the set of all tuples z for which $\{\eta_1\}_1$ is below a certain threshold.[1] We use lemmas 6

[1] We use $\{\eta_i\}_j$ to denote the jth coordinate of the vector η_i.

and 8 to derive the final bound on the probability of $h(Y) = 1$ in the following theorem, the proof of which is deferred to the appendix.

Theorem 2 (**Robustness Guarantee).** *For an isometric Gaussian smoothing noise with variance σ^2, if $\mathbb{P}[h(X) = 1] \geq p$, then:*

$$\mathbb{P}[h(Y) = 1] \geq \Phi(\Phi^{-1}(p) - B/\sigma),$$

where Φ is the standard normal CDF.

The above analysis can be adapted to obtain an upper-bound on $\mathbb{P}[h(Y) = 1]$ of $\Phi(\Phi^{-1}(p) + B/\sigma)$.

3.4.2 Policy Smoothing

Building on these results, we develop *policy smoothing*, a simple model-agnostic randomized-smoothing based technique that can provide certified robustness without increasing the computational complexity of the agent's policy. Given a policy π, we define a smoothed policy $\bar{\pi}$ as:

$$\bar{\pi}\left(\cdot \mid o(s_t)\right) = \pi\left(\cdot \mid o(s_t) + \delta_t\right), \text{ where } \delta_t \sim \mathcal{N}(0, \sigma^2 I).$$

Our goal is to certify the expected sum of the rewards collected over multiple time-steps under policy $\bar{\pi}$. We modify the technique developed by Kumar et al. [51] to certify the expected class scores of a neural network by using the empirical cumulative distribution function (CDF) of the scores under the smoothing distribution to work for the RL setting.

This approach utilizes the fact that the expected value of a random variable $\mathcal{X}$ representing a class score under a Gaussian $\mathcal{N}(0, \sigma^2 I)$ smoothing noise can be expressed using its CDF $F(.)$ as below:

$$\mathbb{E}[\mathcal{X}] = \int_0^\infty (1 - F(x))dx - \int_{-\infty}^0 F(x)dx. \tag{3.2}$$

Given m samples $\{x_i\}_{i=1}^m$ of the random variable $\mathcal{X}$, let us define its empirical CDF at a point x, $F_m(x) = |\{x_i \mid x_i \leq x\}|/m$, as the fraction of samples that are less than or equal to x. Using $F_m(x)$, the Dvoretzky–Kiefer–Wolfowitz inequality can produce high-confidence bounds on the true CDF of $\mathcal{X}$. It says that with probability $1-\alpha$, for $\alpha \in (0, 1]$, the true CDF $F(x)$ is in the range $[\underline{F}(x), \overline{F}(x)]$, where $\underline{F}(x) = F_m(x) - \sqrt{\ln(2/\alpha)/2m}$ and $\overline{F}(x) = F_m(x) + \sqrt{\ln(2/\alpha)/2m}$. For an adversarial perturbation of ℓ_2-size B, the result of [36] bounds the CDF within $[\Phi(\Phi^{-1}(\underline{F}(x)) - B/\sigma), \Phi(\Phi^{-1}(\overline{F}(x)) + B/\sigma)]$, which in turn bounds $E[\mathcal{X}]$ using equation (3.2).

In the RL setting, we can model the total reward as a random variable and obtain its empirical CDF by playing the game using policy $\bar{\pi}$. As above, we can bound the CDF $F(x)$ of the total reward in a range $[\underline{F}(x), \overline{F}(x)]$ using the empirical CDF. Applying Theorem 2, we can bound the CDF within $[\Phi(\Phi^{-1}(\underline{F}(x)) - B/\sigma), \Phi(\Phi^{-1}(\overline{F}(x)) + B/\sigma)]$ for an ℓ_2 adversary of size B. The function h in Theorem 2 could represent the CDF $F(x)$ by indicating whether the total reward computed for an input $z \in (\mathcal{T} \times \mathbb{R}^d)^t$ is below a value x. Finally, equation (3.2) puts bounds on the expected total reward under an adversarial attack.

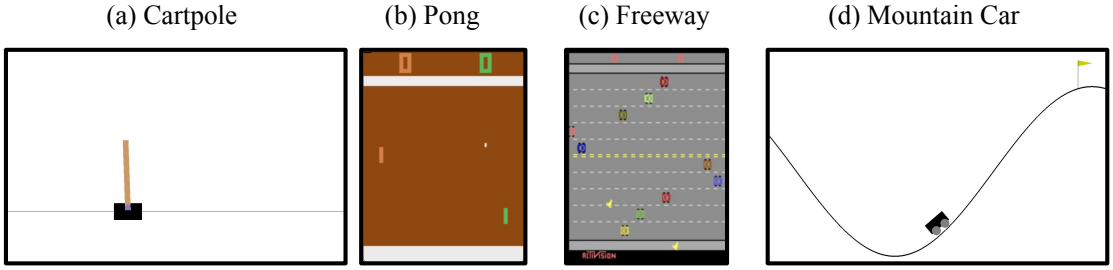


Figure 3.3: Environments used in evaluations rendered by OpenAI Gym [109].

3.5 Experiments

3.5.1 Environments and Setup

We tested on four standard environments: the classical cortrol problems 'Cartpole' and 'Mountain Car' and the Atari games 'Pong' and 'Freeway.' We consider three tasks which use a discrete action space ('Cartpole' and the two Atari games) as well as one task that uses a continuous action space ('Mountain Car'). For the discrete action space tasks, we use a standard Deep Q-Network (DQN) [91] model, while for 'Mountain Car', we use Deep Deterministic Policy Gradient (DDPG) [110].

As is common in DQN and DDPG, our agents choose actions based on multiple frames of observations. In order to apply a realistic threat model, we assume that the adversary acts on each frame *only once* when it is first observed. The adversarial distortion is then maintained when the same frame is used in future time-steps. In other words, we consider the observation at time step o_t (discussed in Section 3.3) to be only the *new* observation at time t: this means that the adversarial/noise perturbation η_t, as a *fixed* vector, continues to be used to select the next action for several subsequent time-steps. This is a realistic model because we are assuming that the adversary can affect the agent's observation of states, not necessarily the agent's *memory* of previous observations. As

in other works on smoothing-based defenses (e.g., [36]), we add noise during training as well as at test time. We use DQN and DDPG implementations from the popular stable-baselines3 package [111]: hyperparameters are provided in the appendix. In experiments, we report and certify for the total *non-discounted* ($\gamma = 1$) reward.

In 'Cartpole', the observation vector consists of four kinematic features. We use a simple MLP model for the Q-network, and tested two variations: one in which the agent uses five frames of observation, and one in which the agent uses only a single frame (shown in the appendix).

In order to show the effectiveness of our technique on tasks involving high-dimensional state observations, we chose two tasks ('Pong' and 'Freeway') from the Atari environment, where state observations are image frames, observed as 84×84 pixel greyscale images. For 'Pong', we test on a "one-round" variant of the original environment. In our variant, the game ends after one player, either the agent or the opponent, scores a goal: the reward is then either zero or one. Note that this is *not* a one-timestep episode: it takes typically on the order of 100 timesteps for this to occur. Results for a full Pong game are presented in the appendix: as explained there, we find that the certificates unfortunately do not scale with the length of the game. For the 'Freeway' game, we play on 'Hard' mode and end the game after 250 timesteps.

In order to test on an environment with a *continuous* action space, we chose the 'Mountain Car' environment. Note that previous certification results for reinforcement learning, which certify actions at individual states rather than certifying the overall reward [100] *cannot* be applied to continuous action state problems. In this environment, the observation vector consists of two kinematic features (position and velocity), and the

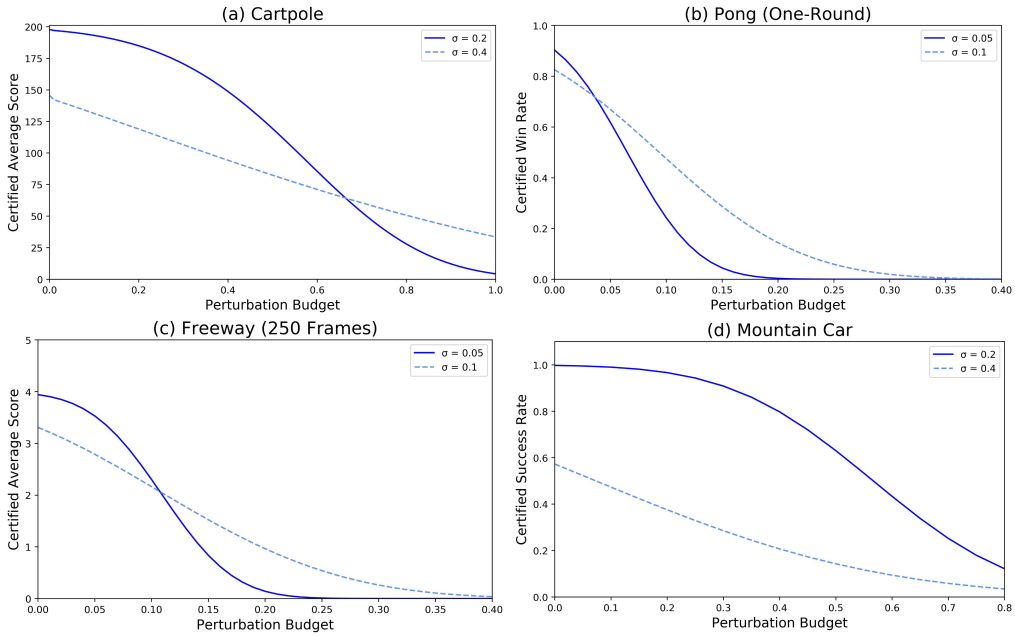


Figure 3.4: Certified performance for various environments. The certified lower-bound on the mean reward is based on a 95% lower confidence interval estimate of the mean reward of the smoothed model, using 10,000 episodes.

action is one continuous scalar (acceleration). As in 'Cartpole', we use five observation frames and a simple MLP policy. We use a slight variant of the original environment: we do not penalize for fuel cost so the reward is a boolean representing whether or not the car reaches the destination in the time allotted (999 steps).

3.5.2 Results

Certified lower bounds on the expected total reward, as a function of the total perturbation budget, are presented in Figure 3.4. For tasks with zero-one total reward ('Pong' and 'Mountain Car'), the function to be smoothed represents the total reward: $h(\cdot) = R$ where R is equal to 1 if the agent wins the round, and 0 otherwise. To compute certificates on games with continuous scores ('Cartpole' and 'Freeway'), we use CDF smoothing [51]: see appendix for technical details.

In order to evaluate the robustness of both undefended and policy-smoothed agents,

we developed an attack tailored to the threat model defined in 3.1, where the adversary makes a perturbation to state observations which is *bounded over the entire episode*. For DQN agents, as in [99], we perturb the observation o such that the perturbation-induced action $a' := \arg\max_a Q(o + \epsilon_t, a)$ minimizes the (network-approximated) Q-value of the true observation $Q(o, a')$. However, in order to conserve adversarial budget, we *only* attack if the gap between attacked q-value $Q(o, a')$ and the clean q-value $\max_a Q(o, a)$ is sufficiently large, exceeding a preset threshold λ_Q. In practice, this allows the attacker to concentrate the attack budget only on the time-steps which are critical to the agent's performance. When attacking DDPG, where both a Q-value network and a policy network π are trained and the action is taken according to π, we instead minimize $Q(o, \pi(o + \epsilon_t)) + \lambda\|\epsilon_t\|^2$ where the hyperparameter λ plays an analogous role in focusing perturbation budget on "important" steps, as judged by the effect on the approximated Q-value. Empirical results are presented in Figure 3.5. We see that the attacks are effective on the undefended agents (red, dashed lines). In fact, from comparing Figures 3.4 and 3.5, we see that, for the Pong and Cartpole environments, the undefended performance under attack is worse than the *certified lower bound* on the performance of the policy-smoothed agents under *any possible* attack: our certificates are the clearly non-vacuous for these environments. Further details on the attack optimizations are provided in the appendix.

We also present an attempted empirical attack on the smoothed agent, adapting techniques for attacking smoothed classifiers from [39] (solid blue lines). We observed that our model was highly robust to this attack – significantly more robust than guaranteed by our certificate. However, it is not clear whether this is due to looseness in the certificate or to weakness of the attack: the significant practical challenges to attacking smoothed

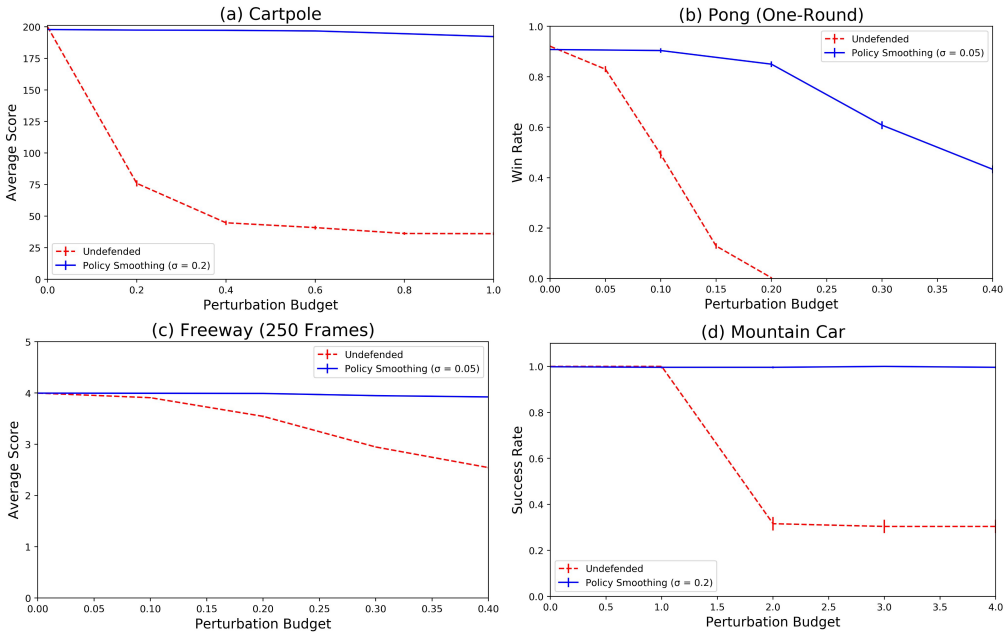


Figure 3.5: Empirical robustness of defended and undefended agents. Full details of attacks are presented in appendix.

agents are also discussed in the appendix.

3.6 Conclusion

In this work, we extend randomized smoothing to design a procedure that can make any reinforcement learning agent provably robust against adversarial attacks without significantly increasing the complexity of the agent's policy. We show how to adapt existing theory on randomized smoothing from static tasks such as classification, to the dynamic setting of RL. By proving an adaptive version of the celebrated Neyman-Pearson Lemma, we show that by adding Gaussian smoothing noise to the input of the policy, one can certifiably defend it against norm-bounded adversarial perturbations of its input. The policy smoothing technique and its theory covers a wide range of adversaries, policies and environments. Our analysis is tight, meaning that the certificates we achieve are best possible unless restrictive assumptions about the RL game are made. In our experiments, we show that our method provides meaningful guarantees on the robustness of the defended policies

and the total reward they achieve even in the worst case is higher than an undefended policy. In the future, the introduction of randomized smoothing to RL could inspire the design of provable robustness techniques for control problems in dynamic real-world environments and multi-agent RL settings.

3.7 Appendices

A Tightness of the Certificate

Here, we present a worst-case environment-policy pair that achieves the bound in Theorem 2, showing that our robustness certificate is in fact tight. For a given environment $M = (S, A, T, R, \gamma)$ and a policy π, let p be a lower-bound on the probability that the total reward obtained by policy π under Gaussian smoothing (no adversary) with variance σ^2 is above a certain threshold ν, i.e.,

$$\mathbb{P}\left[\sum_{i=1}^{t} \gamma^{i-1} R_i \geq \nu\right] \geq p.$$

Let H_p be the class of all such environment-policy pairs that cross this reward threshold with probability at least p. We construct an environment-policy pair (M', π') that achieves the reward threshold ν with probability $\Phi(\Phi^{-1}(p) - B/\sigma)$ under the structured adversary ϵ^{st}. Note that, this does not mean that ϵ^{st} is the strongest possible adversary for a general environment-policy pair. It only shows that the performance of policy π' in environment M' under the adversary ϵ^{st} is a lower-bound on the performance of a general environment-policy pair under a general adversary. Consider a one-step game with environment $M' = (S, A, T', R', \gamma)$ with a deterministic observation function o of the state-space and a policy π' such that π' returns an action $a_1 \in A$ if the first coordinate of $o(s_1) + \eta_1$ is at most $\omega = \{o(s_1)\}_1 + \sigma\Phi^{-1}(p)$ and another action $a_2 \in A$ otherwise. Here $\{o(s_1)\}_1$ represents the first coordinate of $o(s_1)$. The environment offers a reward ν if the action in the first step is

a_1 and 0 when it is a_2. The game terminates immediately. The probability of the reward being above ν is equal to the probability of the action being a_1. When η_1 is sampled from the Gaussian distribution, this probability is equal to $\Phi((\omega - \{o(s_1)\}_1)/\sigma) = p$. Therefore, $(M', \pi') \in H_p$. Under the presence of the structured adversary ϵ^{st} defined in Section 3.4.1, this probability after smoothing becomes $\Phi((\omega - \{o(s_1)\}_1 - B)/\sigma) = \Phi(\Phi^{-1}(p) - B/\sigma)$, which is same as the bound in Theorem 2.

B Static Vs. Adaptive Setting

In this section, we illustrate the difference between the adversarial distributions in the static setting and the adaptive setting. Naively, one might assume that smoothing-based robustness guarantees can be applied directly to reinforcement learning, by adding noise to observations. For example, it seems plausible to use Cohen et al.'s ℓ_2 certificate [36], which relies on the overlap in the distributions of isometric Gaussians with different means, by simply adding Gaussian noise to each observation (Figure 3.1-a). However, as we demonstrate with a toy example in Figure 3.1-(b-d), the Cohen et al. certificate *cannot* be applied directly to the RL setting, because adding noise to sequential observations *does not* result in an isometric Gaussian distribution over the space of observations. This is because the adversarial offset to later observations may be conditioned on the noise added to previous observations. In 3.1-(b-d), we consider a two-step episode, and for simplicity, we consider a case where the ground-truth observations at each step are fixed. At step 1, the noised distributions of the clean observation o_1 and the adversarially-perturbed observation o_1' are both Gaussians and overlap substantially, similar to in the

standard classification setting (panel b). However, we see in panel (c) that the adversarial perturbation ϵ_2 added to o_2 *can depend on* the smoothed value of o_1'. This is because the agent may leak information about the observation that it receives after smoothing $(o_1 + \eta_1)$ to the adversary, for example through its choice of actions. After smoothing is performed on o_2, the adaptive nature of the adversary causes the distribution of smoothed observations to no longer be an isometric Gaussian in the adversarial case (panel d). The standard certification results therefore cannot be applied.

C Neyman–Pearson lemma [1993] in Smoothing

In the context of randomized smoothing, the Neyman–Pearson lemma produces the worst-case decision boundary of a classifier based on the estimated probability of the top class under the smoothing distribution. It says that this boundary is a region where the ratio of the probability density functions of the smoothing distributions at the clean input and the perturbed input is a constant. When the two distributions are isometric Gaussians, as is the case in static settings like image classification, this boundary takes the form of a hyper-plane (see Appendix A of [36]). However, in the dynamic setting of RL, the smoothing distribution after adding the adversarial perturbation may not be isometric even if the smoothing noise at each time-step was sampled from an isometric Gaussian distribution (see figure 1, section 'Technical contributions' and Appendix A). So, we formulate and prove an adaptive version of the Neyman-Pearson lemma to obtain provable robustness in RL through randomized smoothing.

D Proof of Lemma 6

Statement: *For any general adversary ϵ and an $\Gamma \subseteq (\mathcal{T} \times \mathbb{R}^d)^t$, there exists a deterministic adversary ϵ^{dt} such that,*

$$\mathbb{P}[Y^{dt} \in \Gamma] \leq \mathbb{P}[Y \in \Gamma],$$

where Y^{dt} is the random variable for the distribution defined by the adversary ϵ^{dt}.

Proof. Consider a time-step j such that $\forall i < j, \epsilon_i$ is a deterministic function of $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_{i-1}, \eta_{i-1}, \tau_i$. Let $\mathcal{H} = \{z \mid z_1 = \tau_1, z_2 = \eta_1, z_3 = \tau_2, z_4 = \eta_2, \ldots, z_{2j-1} = a_j\}$ be the set of points whose first $2j-1$ coordinates are fixed to an arbitrary set of values $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_j$. In the space defined by $\mathcal{H}$, $\epsilon_1, \ldots, \epsilon_{j-1}$ are fixed vectors in $\mathbb{R}^d$ and ϵ_j is sampled from a fixed distribution over the vectors with ℓ_2-norm at most B_r^j. Let $Y_{\mathcal{H}}^{\gamma}$ be the random variable representing the distribution over points in $\mathcal{H}$ defined by the adversary for which $\epsilon_j = \gamma$, such that $\|\gamma\|_2 \leq B_r^j$. Define an adversary ϵ', such that, $\epsilon'_i = \epsilon_i, \forall i \neq j$. Set ϵ'_j to the vector γ that minimizes the probability that $Y_{\mathcal{H}}^{\gamma}$ assigns to $\Gamma \cap \mathcal{H}$, i.e.,

$$\epsilon'_j = \underset{\|\gamma\|_2 \leq B_r^j}{\arg\min}\, \mathbb{P}[Y_{\mathcal{H}}^{\gamma} \in \Gamma \cap \mathcal{H}]$$

The adversary ϵ' behaves as ϵ up to step $j-1$. At step j, it sets ϵ'_j to the γ that minimizes the probability it assigns to $\Gamma \cap \mathcal{H}$, based on the values $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_j$. After that, it mimics ϵ till the last time-step t. Therefore, for a given tuple $(z_1, z_2, \ldots, z_{2j-1}) = (\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_j)$,

$$\mathbb{P}[Y_{\mathcal{H}}^{\epsilon'_j} \in \Gamma \cap \mathcal{H}] \leq \mathbb{P}[Y \in \Gamma \cap \mathcal{H}]$$

Since both adversaries are same up to step $j-1$, their respective distributions over $z_1, z_2, \ldots, z_{2j-1}$ remains same as well. Therefore, integrating both sides of the above inequality over the space of all tuples $(z_1, z_2, \ldots, z_{2j-1})$, we have:

$$\begin{aligned}
&\int \mathbb{P}[Y_{\mathcal{H}}^{\epsilon'_j} \in \Gamma \cap \mathcal{H}] p_Y(z_1, z_2, \ldots, z_{2j-1}) dz_1 dz_2 \ldots dz_{2j-1} \\
&\qquad \leq \int \mathbb{P}[Y \in \Gamma \cap \mathcal{H}] p_Y(z_1, z_2, \ldots, z_{2j-1}) dz_1 dz_2 \ldots dz_{2j-1} \\
&\qquad \implies \mathbb{P}[Y' \in \Gamma] \leq \mathbb{P}[Y \in \Gamma],
\end{aligned}$$

where Y' is the random variable corresponding to ϵ'. Thus, we have constructed an adversary where the first j adversarial perturbations are a deterministic function of the τ_is and η_is of the previous rounds. Applying the above step sufficiently many times we can construct a deterministic adversary ϵ^{dt} represented by the random variable Y^{dt} such that

$$\mathbb{P}[Y^{dt} \in \Gamma] \leq \mathbb{P}[Y \in \Gamma].$$

□

E Proof of Lemma 8

Lemma 8 states that the structured adversary characterises the worst-case scenario. Before proving this lemma, let us first show that any deterministic adversary can be converted to one that uses up the entire budget of B without increasing the probability it assigns to h being one in the worst-case. For each step i, let us define a *used budget* $B_u^i = \|(\epsilon_1, \epsilon_2, \ldots, \epsilon_{i-1})\|_2$ as the norm of the perturbations of the previous steps and a *remaining*

budget $B_r^i = \sqrt{B^2 - (B_u^i)^2}$ as an upper-bound on the norm of the perturbations of the remaining steps. Note that, $B_u^1 = 0$ and $B_r^1 = B$.

Consider a version $\tilde{\epsilon}^{dt}$ of the deterministic adversary that uses up the entire available budget B by scaling up ϵ_t^{dt} such that its norm is equal to B_r^t, i.e., setting it to $\epsilon_t^{dt} B_r^t / \|\epsilon_t^{dt}\|_2$. Let $\tilde{Y}^{dt}$ be the random variable representing $\tilde{\epsilon}^{dt}$.

Lemma. *If* $\Gamma_{\tilde{Y}^{dt}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{\tilde{Y}^{dt}}(z) \leq q\mu_X(z)\}$ *for some* $q \geq 0$ *and* $\mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}]$, *then* $\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}]$.

Proof. Consider $\Gamma_{Y^{dt}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{dt}}(z) \leq q'\mu_X(z)\}$ for some $q' \geq 0$, such that, $\mathbb{P}[X \in \Gamma_{Y^{dt}}] = p$ for some lower-bound p on $\mathbb{P}[h(X) = 1]$. Then, by the Neyman-Pearson Lemma we have that,

$$\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}].$$

Now consider a space $\mathcal{H}$ in $(\mathcal{T} \times \mathbb{R}^d)^t$ where all but the last element of the tuple z are fixed, i.e., $\mathcal{H} = \{z \mid z_1 = \tau_1, z_2 = \eta_1, z_3 = \tau_2, z_4 = \eta_2, \ldots, z_{2t-1} = \tau_t\}$ Since, ϵ^{dt} is a deterministic adversary where each ϵ_i^{dt} is a deterministic function of the previous τ_is and η_is, each ϵ_i^{dt} is also fixed in $\mathcal{H}$. Therefore, in $\mathcal{H}$, both μ_X and $\mu_{Y^{dt}}$ are two isometric Gaussians in the space of the η_is and the set $\mathcal{H} \cap \Gamma_{Y^{dt}}$ is a hyperplane. The probability assigned by Y^{dt} to $\mathcal{H} \cap \Gamma_{Y^{dt}}$ is proportional to the distance of the center of the corresponding Gaussian. In the construction of $\tilde{\epsilon}^{dt}$, this distance can only increase, therefore,

$$\mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{Y^{dt}}]$$

Now, consider a function $h_{\Gamma_{Y^{dt}}}(z)$ which outputs one if $z \in \Gamma_{Y^{dt}}$ and zero otherwise. Construct the set $\Gamma_{\tilde{Y}^{dt}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{\tilde{Y}^{dt}}(z) \leq q\mu_X(z)\}$ for some $q \geq 0$ such that,

$$\mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}] = p = \mathbb{P}[h_{\Gamma_{Y^{dt}}}(X) = 1].$$

Then, by the Neyman-Pearson Lemma, we have,

$$\begin{aligned}
&\mathbb{P}[h_{\Gamma_{Y^{dt}}}(\tilde{Y}^{dt}) = 1] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}] \\
\text{or,}\quad &\mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{Y^{dt}}] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}] && \text{(from definition of } h_{\Gamma_{Y^{dt}}}) \\
\text{or,}\quad &\mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}] \\
\text{or,}\quad &\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}], && \text{(from the above two inequalities)}
\end{aligned}$$

proving the statement of the lemma. □

Now, we prove lemma 8 below:

Statement: *If* $\Gamma_{Y^{st}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{st}}(z) \leq q\mu_X(z)\}$ *for some* $q \geq 0$ *and* $\mathbb{P}[h(X) = 1] \geq \mathbb{P}[X \in \Gamma_{Y^{st}}]$, *then* $\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}]$.

Proof. Construct the set $\Gamma_{\tilde{Y}^{dt}}$ as defined in the above lemma for a $q \geq 0$ such that $\mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}] = p$, for some lower-bound p on $\mathbb{P}[h(X) = 1]$. Then,

$$\mathbb{P}[h(Y^{dt}) = 1] \geq \mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}]$$

Now consider the structured adversary ϵ^{st} in which $\epsilon^{st}_1 = (B, 0, \ldots, 0)$ and $\epsilon^{st}_i = (0, 0, \ldots, 0)$ for $i > 1$. Define the set $\Gamma_{Y^{st}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{st}}(z) \leq q\mu_X(z)\}$ for the same q as

above. Then, we can show that:

1. $\mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}]$, and

2. $\mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[X \in \Gamma_{Y^{st}}]$

which, in turn, prove the statement of the lemma.

Let $\mathcal{N}$ and $\mathcal{N}_{\epsilon_i}$ represent Gaussian distributions centered at origin and ϵ_i respectively. Then, we can write μ_X and μ_Y as below:

$$\mu_X(z) = \prod_{i=1}^{t} \mu_{T_i}(\tau_i \mid \tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_{i-1}, \eta_{i-1}) \mu_{\mathcal{N}}(\eta_i)$$
$$\mu_{\tilde{Y}^{dt}}(z) = \prod_{i=1}^{t} \mu_{T_i}(\tau_i \mid \tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_{i-1}, \eta_{i-1}) \mu_{\mathcal{N}_{\tilde{\epsilon}_i^{dt}}}(\eta_i)$$

where μ_{T_i} is the conditional probability distribution of token τ_i given the previous tokens and offsets. Therefore,

$$\frac{\mu_{\tilde{Y}^{dt}}(z)}{\mu_X(z)} = \prod_{i=1}^{t} \frac{\mu_{\mathcal{N}_{\tilde{\epsilon}_i^{dt}}}(\eta_i)}{\mu_{\mathcal{N}}(\eta_i)} = \prod_{i=1}^{t} e^{\frac{\eta_i^T \eta_i - (\eta_i - \tilde{\epsilon}_i^{dt})^T (\eta_i - \tilde{\epsilon}_i^{dt})}{2\sigma^2}}$$
$$\frac{\mu_{\tilde{Y}^{dt}}(z)}{\mu_X(z)} \leq q \iff \sum_{i=1}^{t} 2\eta_i^T \tilde{\epsilon}_i^{dt} - (\tilde{\epsilon}_i^{dt})^T \tilde{\epsilon}_i^{dt} \leq 2\sigma^2 \ln q$$

Consider a round $j \leq t$ such that $\tilde{\epsilon}_i^{dt} = 0, \forall i > j+1$ and $\tilde{\epsilon}_{j+1}^{dt} = (B_r^{j+1}, 0, \ldots, 0)$. We can always find such a j as we always have $\tilde{\epsilon}_{t+1}^{dt} = (B_r^{t+1}, 0, \ldots, 0)$, since $B_r^{t+1} = 0$. Note that, $B_r^{j+1} = \sqrt{B^2 - \left(B_u^{j+1}\right)^2}$ and in turn $\tilde{\epsilon}_{j+1}^{dt}$ are functions of $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_j$ and not τ_{j+1}. Let $\mathcal{H} = \{z \mid z_1 = \tau_1, z_2 = \eta_1, z_3 = \tau_2, z_4 = \eta_2, \ldots, z_{2j-1} = \tau_j\}$ be the set of points whose first $2j-1$ coordinates are fixed to an arbitrary set of values $\tau_1, \eta_1, \tau_2, \eta_2, \ldots, \tau_j$. For points in $\mathcal{H}$, all $\tilde{\epsilon}_i^{dt}$ for $i \leq j+1$ are fixed and for $i > j+1$

are set to zero. Let $\tilde{Y}_{\mathcal{H}}^{dt}$ denote the random variable representing the distribution of points in $\mathcal{H}$ defined by the adversary $\tilde{\epsilon}^{dt}$ (corresponding random variable $\tilde{Y}^{dt}$). In the space of $\eta_j, \eta_{j+1}, \ldots, \eta_t$, this is an isometric Gaussian centered at $(\tilde{\epsilon}_j^{dt}, \tilde{\epsilon}_{j+1}^{dt}, 0, \ldots, 0)$. Therefore, $\Gamma \cap \mathcal{H}$ is given by

$$\sum_{i=1}^{j+1} 2\eta_i^T \tilde{\epsilon}_i^{dt} - (\tilde{\epsilon}_i^{dt})^T \tilde{\epsilon}_i^{dt} \leq 2\sigma^2 \ln t$$

$$\text{or,} \quad \eta_j^T \tilde{\epsilon}_j^{dt} + \eta_{j+1}^T \tilde{\epsilon}_{j+1}^{dt} \leq \beta, \tag{3.3}$$

for some constant β dependent on $\eta_1, \tilde{\epsilon}_1^{dt}, \ldots, \eta_{j-1}, \tilde{\epsilon}_{j-1}^{dt}, \sigma$ and t. The probability assigned by the Gaussian random variable $Y_{\mathcal{H}}$ to the half-space defined by (3.3) is proportional to the distance of the center of the Gaussian from the hyper-plane in (3.3), which is equal to:

$$\frac{\|\tilde{\epsilon}_j^{dt}\|^2 + \|\tilde{\epsilon}_{j+1}^{dt}\|^2 - \beta}{\sqrt{\|\tilde{\epsilon}_j^{dt}\|^2 + \|\tilde{\epsilon}_{j+1}^{dt}\|^2}} = \frac{(B_r^j)^2 - \beta}{B_r^j},$$

where the equality follows from:

$$\begin{aligned}
\|\tilde{\epsilon}_j^{dt}\|^2 + \|\tilde{\epsilon}_{j+1}^{dt}\|^2 &= \|\tilde{\epsilon}_j^{dt}\|^2 + (B_r^{j+1})^2 \\
&= \|\tilde{\epsilon}_j^{dt}\|^2 + B^2 - (B_u^{j+1})^2 && \text{(from definition of } B_r^i\text{)} \\
&= \|\tilde{\epsilon}_j^{dt}\|^2 + B^2 - (\|\tilde{\epsilon}_1^{dt}\|^2 + \|\tilde{\epsilon}_2^{dt}\|^2 + \ldots + \|\tilde{\epsilon}_j^{dt}\|^2) \\
&= B^2 - (\|\tilde{\epsilon}_1^{dt}\|^2 + \|\tilde{\epsilon}_2^{dt}\|^2 + \ldots + \|\tilde{\epsilon}_{j-1}^{dt}\|^2) \\
&= B^2 - (B_u^j)^2 = (B_r^j)^2.
\end{aligned}$$

Now, consider an adversary $\tilde{\epsilon}^{dt'}$ such that $\tilde{\epsilon}_i^{dt'} = \tilde{\epsilon}_i^{dt}, \forall i \leq j-1$, $\tilde{\epsilon}_j^{dt'} = (B_r^j, 0, \ldots, 0)$, and

$\tilde{\epsilon}_i^{dt} = 0, \forall i > j$. Let $\tilde{Y}^{dt'}$ be the corresponding random variable. Define $\Gamma_{\tilde{Y}^{dt'}}$ similar to $\Gamma_{\tilde{Y}^{dt}}$. Then, $\Gamma_{\tilde{Y}^{dt'}} \cap \mathcal{H}$ is given by

$$\eta_j^T(B_r^j, 0, \ldots, 0) \leq \beta, \tag{3.4}$$

which is obtained by replacing $\tilde{\epsilon}_j^{dt}$ with $(B_r^j, 0, \ldots, 0)$ and $\tilde{\epsilon}_{j+1}^{dt}$ with $(0, 0, \ldots, 0)$ in inequality (3.3) about the origin. Define $\tilde{Y}_{\mathcal{H}}^{dt'}$ similar to $\tilde{Y}_{\mathcal{H}}^{dt}$, and just like $\tilde{Y}_{\mathcal{H}}^{dt}$, the distribution of $\tilde{Y}_{\mathcal{H}}^{dt'}$ is also an isometric Gaussian, but is centered at $((B_r^j, 0, \ldots, 0), (0, 0, \ldots, 0))$. The probability assigned by this Gaussian distribution to $\Gamma_{\tilde{Y}^{dt'}} \cap \mathcal{H}$ is proportional to the distance of its center to the hyper-plane defining the region in (3.4), which is equal to $((B_r^j)^2 - \beta)/B_r^j$. Therefore,

$$\mathbb{P}[\tilde{Y}_{\mathcal{H}}^{dt} \in \Gamma_{\tilde{Y}^{dt}} \cap \mathcal{H}] = \mathbb{P}[\tilde{Y}_{\mathcal{H}}^{dt'} \in \Gamma_{\tilde{Y}^{dt'}} \cap \mathcal{H}].$$

The key intuition behind this step is that, for isometric Gaussian smoothing distribution, the worst-case probability assigned by the adversarial distribution only depends on the magnitude of the perturbation and not its direction. Figure 3.6 illustrates this property for a two-dimensional input space.

Since both adversaries are same up to step $j-1$, their respective distributions over $z_1, z_2, \ldots, z_{2j-1}$ remains same as well, i.e., $p_{\tilde{Y}^{dt}}(z_1, z_2, \ldots, z_{2j-1}) = p_{\tilde{Y}^{dt'}}(z_1, z_2, \ldots, z_{2j-1})$. Integrating over the space of all tuples $(z_1, z_2, \ldots, z_{2j-1})$, we have:

$$\begin{aligned}
&\int \mathbb{P}[\tilde{Y}_{\mathcal{H}}^{dt} \in \Gamma_{\tilde{Y}^{dt}} \cap \mathcal{H}] p_{\tilde{Y}^{dt}}(z_1, z_2, \ldots, z_{2j-1}) dz_1 dz_2 \ldots dz_{2j-1} \\
&\qquad = \int \mathbb{P}[\tilde{Y}_{\mathcal{H}}^{dt'} \in \Gamma_{\tilde{Y}^{dt'}} \cap \mathcal{H}] p_{\tilde{Y}^{dt'}}(z_1, z_2, \ldots, z_{2j-1}) dz_1 dz_2 \ldots dz_{2j-1}
\end{aligned}$$

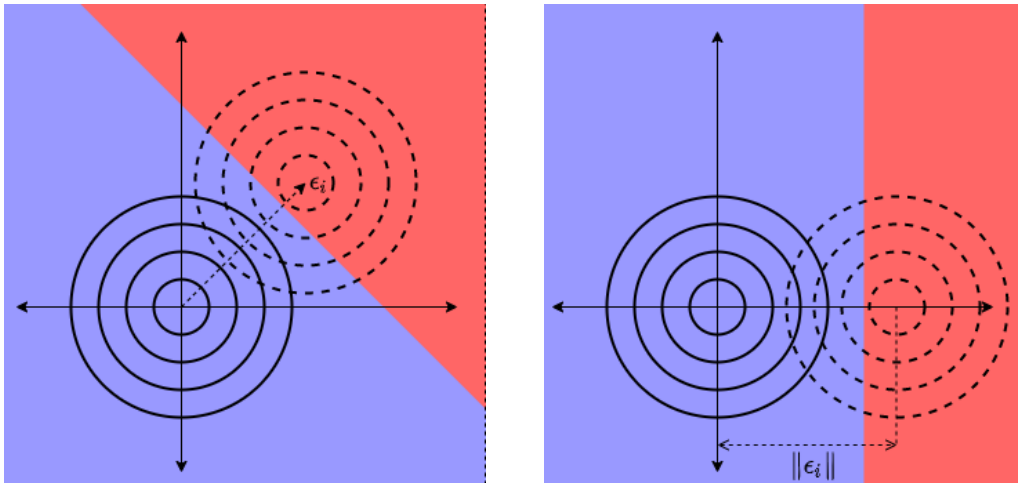


Figure 3.6: General adversarial perturbation vs. perturbation aligned along the first dimension. Blue and red regions denote where the worst-case function is one and zero respectively.

$$\implies \mathbb{P}[\tilde{Y}^{dt}_{\mathcal{H}} \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[\tilde{Y}^{dt'}_{\mathcal{H}} \in \Gamma_{\tilde{Y}^{dt'}}],$$

Since the distribution defined by X (with no adversary) over the space of η_is is a Gaussian centered at origin whose distance to both $\Gamma_{\tilde{Y}^{dt}} \cap \mathcal{H}$ and $\Gamma_{\tilde{Y}^{dt'}} \cap \mathcal{H}$ is the same (equal to $-\beta/B^j_r$), it assigns the same probability to both (3.3) and (3.4). Therefore,

$$\mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt'}}].$$

Thus, we have constructed an adversary with one less non-zero ϵ_i. Applying, this step sufficiently many times we can obtain the adversary ϵ^{st} such that,

$$\mathbb{P}[\tilde{Y}^{dt} \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}] \quad \text{and} \quad \mathbb{P}[X \in \Gamma_{\tilde{Y}^{dt}}] = \mathbb{P}[X \in \Gamma_{Y^{st}}]$$

which completes the proof. □

F Proof of Theorem 2

Statement: *For an isometric Gaussian smoothing noise with variance σ^2, if $\mathbb{P}[h(X) = 1] \geq p$, then:*

$$\mathbb{P}[h(Y) = 1] \geq \Phi(\Phi^{-1}(p) - B/\sigma).$$

Proof. Define $\Gamma_Y = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_Y(z) \leq q\mu_X(z)\}$ for an appropriate q such that $\mathbb{P}[X \in \Gamma_Y] = p$. Then, by the Neyman-Pearson lemma, we have $\mathbb{P}[h(Y) = 1] \geq \mathbb{P}[Y \in \Gamma_Y]$. Applying lemma 1, we know that there exists a deterministic adversary ϵ^{dt} represented by random variable Y^{dt}, such that,

$$\mathbb{P}[h(Y) = 1] \geq \mathbb{P}[Y \in \Gamma_Y] \geq \mathbb{P}[Y^{dt} \in \Gamma_Y]. \tag{3.5}$$

Now define a function $h_{\Gamma_Y}(z) = \mathbf{1}_{\{z \in \Gamma_Y\}}$ and a set $\Gamma_{Y^{dt}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{dt}}(z) \leq q'\mu_X(z)\}$ for an appropriate $q' > 0$, such that, $\mathbb{P}[X \in \Gamma_{Y^{dt}}] = \mathbb{P}[h_{\Gamma_Y}(X) = 1] = p$. Applying the Neyman-Pearson lemma again, we have:

$$\begin{aligned}
&\mathbb{P}[h_{\Gamma_Y}(Y^{dt}) = 1] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] \\
\text{or,}\quad &\mathbb{P}[Y^{dt} \in \Gamma_Y] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] && \text{(from definition of } h_{\Gamma_Y}\text{)} \\
\text{or,}\quad &\mathbb{P}[h(Y) = 1] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] && \text{(from inequality (3.5))}
\end{aligned}$$

Define $h_{\Gamma_{Y^{dt}}}(z) = \mathbf{1}_{\{z \in \Gamma_{Y^{dt}}\}}$. For the structured adversary ϵ^{st} represented by Y^{st}, define $\Gamma_{Y^{st}} = \{z \in (\mathcal{T} \times \mathbb{R}^d)^t \mid \mu_{Y^{st}}(z) \leq q''\mu_X(z)\}$ for an appropriate $q'' > 0$, such that,

$\mathbb{P}[X \in \Gamma_{Y^{st}}] = \mathbb{P}[h_{\Gamma_{Y^{dt}}}(X) = 1] = p$. Applying lemma 3, we have:

$$\begin{aligned}
\mathbb{P}[h_{\Gamma_{Y^{dt}}}(Y^{dt}) = 1] &\geq \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}] \\
\mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}] &\geq \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}] && \text{(from definition of } h_{\Gamma_{Y^{dt}}}) \\
\mathbb{P}[h(Y) = 1] &\geq \mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}] && \text{(since } \mathbb{P}[h(Y) = 1] \geq \mathbb{P}[Y^{dt} \in \Gamma_{Y^{dt}}])
\end{aligned}$$

$\Gamma_{Y^{st}}$ is defined as the set of points z which satisfy:

$$\begin{aligned}
\frac{\mu_{Y^{st}}(z)}{\mu_X(z)} \leq q'' \quad \text{or,} \quad \frac{\mu_{\mathcal{N}_{\tilde{\epsilon}_1^{st}}}(\eta_1)}{\mu_{\mathcal{N}}(\eta_1)} \leq q'' \\
\eta_1^T(B, 0, \ldots, 0) \leq \beta \quad \text{or,} \quad \{\eta_1\}_1 \leq \beta/B
\end{aligned}$$

for some constant β. This is the set of all tuples z where the first coordinate of η_1 is below a certain threshold γ. Since $\mathbb{P}[X \in \Gamma_{Y^{st}}] = p$,

$$\Phi(\gamma/\sigma) = p \implies \gamma = \sigma\Phi^{-1}(p).$$

Therefore,

$$\mathbb{P}[Y^{st} \in \Gamma_{Y^{st}}] = \Phi\left(\frac{\gamma - B}{\sigma}\right) = \Phi(\Phi^{-1}(p) - B/\sigma).$$

□

G Additional Cartpole Results

We performed two additional experiment on Cartpole: we tested at larger noise levels, ($\sigma = 0.6$ and 0.8) and we tested a variant of the agent architecture. Specifically, in

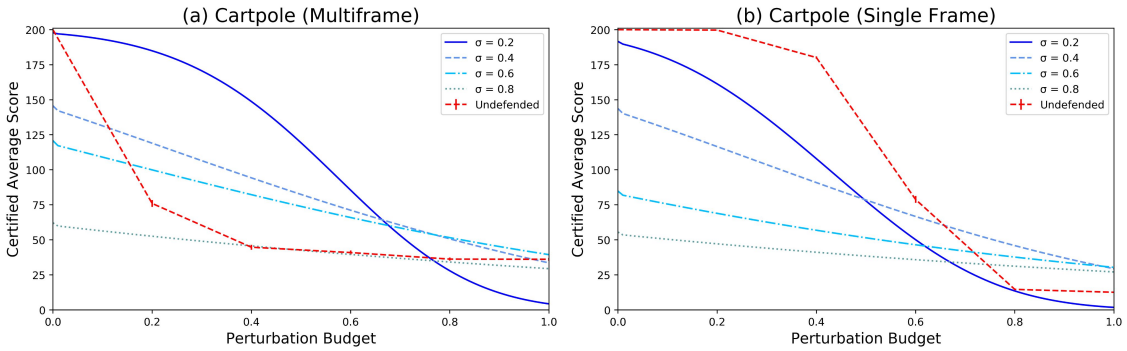


Figure 3.7: Additional Cartpole results. Attacks on smoothed agents at all σ for the multiframe agents are presented in Appendix J

addition to the agent shown in the main text, which uses five frames of observation, we also tested an agent which uses only a single frame. Unlike the Atari environment, the task is in fact solvable (in the non-adversarial case) using only one frame: the observation vector represents the complete system state. We computed certificates for the policy-smoothed version of this model, and tested attacks on the undefended version. (We did not test attacks on the smoothed single-frame variant). As we see in Figure 3.7, we achieve non-vacuous certificates in both settings (i.e, at large perturbation sizes, the smoothed agent is guaranteed to be more robust than the empirical robustness of a non-smoothed agent). However, observe that the undefended agent in the multi-frame setting is much more vulnerable to adversarial attack. This is likely because the increased number of total features (20 vs. four) introduces more complexity of the Q-network, making it more vulnerable against adversarial attack.

H Full Pong Game

In Figure 3.8, we explore a failure case of our technique: we fail to produce non-vacuous certificates for a full Pong game, where the game ends after either player scores

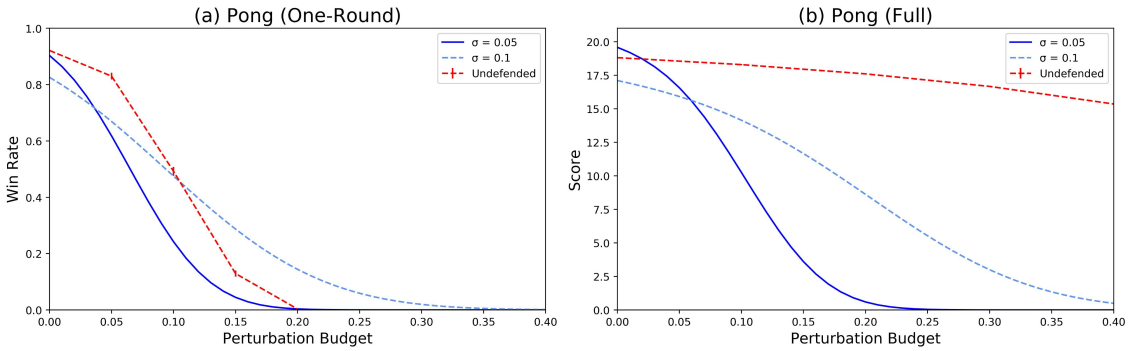


Figure 3.8: Results for the Full Pong game, compared to the single-round game.

21 goals. In particular, while, for the one-round Pong game, the smoothed agent is provably more robust than the empirical performance of the undefended agent, this is clearly not the case for the full game. To understand why our certificate is vacuous here, note that in the the "worst-case" environment that our certificate assumes, any perturbation will (maximally) affect all future rewards. However, in the multi-round Pong game, each round of the game is only loosely coupled to the previous rounds (the ball momentum – but not position – as well as the paddle positions are retained). Therefore, any perturbation can only have a very limited effect on the total reward. Another way to think about this is to recall that in smoothing-based certificates, the noise added to *each* feature is proportional to the *total* perturbation budget of the adversary. In this sort of serial game, the perturbation budget required to attack the average reward scales with the (square root of the) number of rounds, but the noise tolerance of the agent does not similarly scale.

I Training and Clean Test Results

In Figure 3.9, we present the clean (non-attacked) test performance for the experiments presented in the main text, as a function of the smoothing noise σ.

In Figure 3.10, we present the clean training (i.e., validation round) performance as a function of the training time step and the smoothing noise σ. Note that early stopping was applied: the model from the best validation round was kept, and only replaced if a strictly better validation performance was recorded later.

- For Cartpole: logs were not kept after the first time an evaluation round had a perfect average score of 200 (this is because the "best model" was saved for this evaluation, and it would be impossible to beat this score, so training was not continued). However, for other tasks (i.e. mountain car) logs continued after a perfect evaluation round.

- For Freeway: as mentioned in Appendix Section L, we trained 5 times at each noise level, and kept the best of all 5 models. All 5 training curves are shown here for each noise level.

J Complete Attack Results

In Figures 3.11 and 3.12, we report the empirical robustness under attack for all tested values of λ_Q: in the main text, we show only the result for the λ_Q that represents the strongest attack. Figure 3.12 also shows the attacks on smoothed agents for all smoothing noises. All attack results are means over 1000 episodes (except for Mountain Car results,

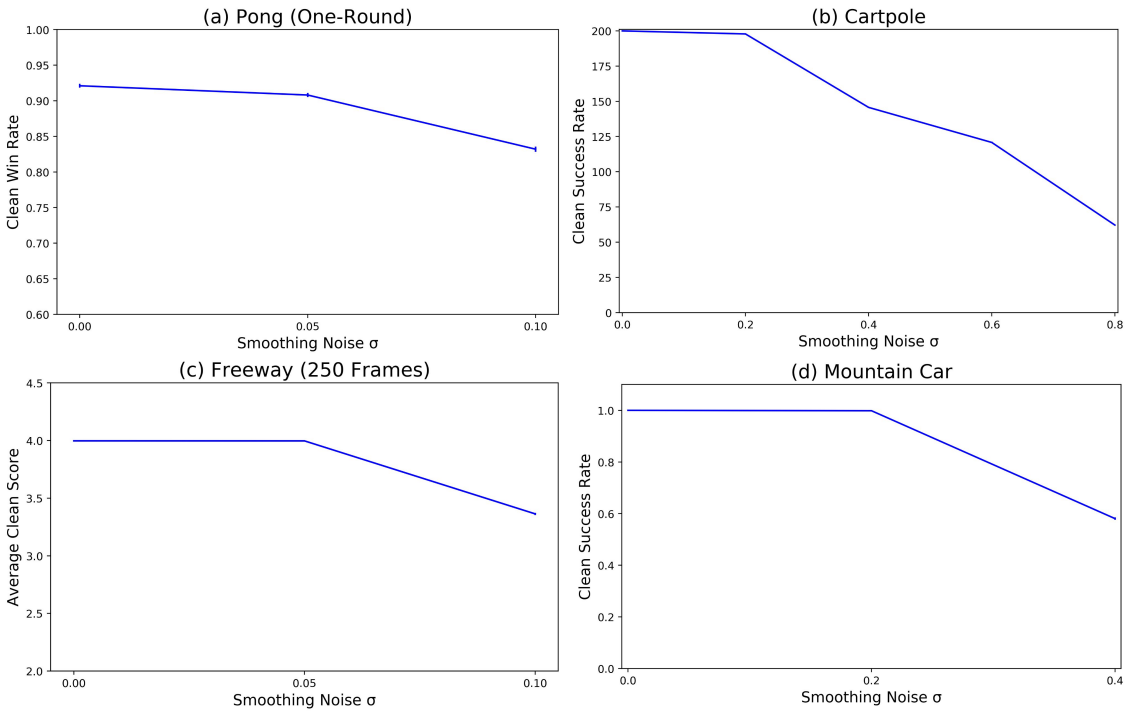


Figure 3.9: Clean test performance as a function of smoothing noise σ.

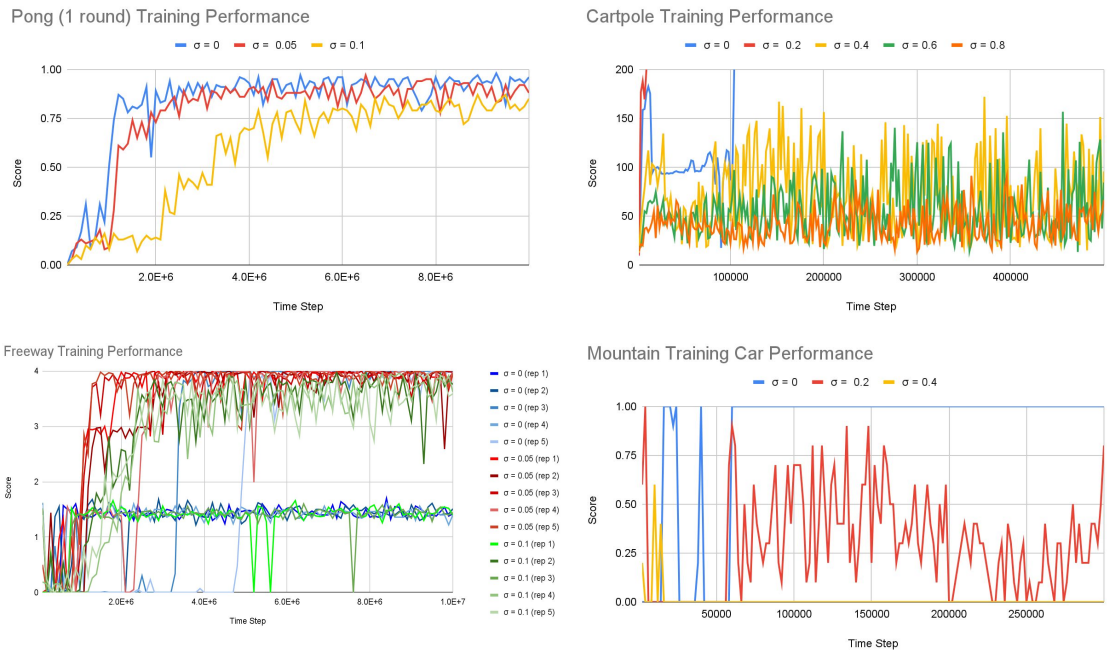


Figure 3.10: Clean training performance as a function of smoothing noise σ and training step.

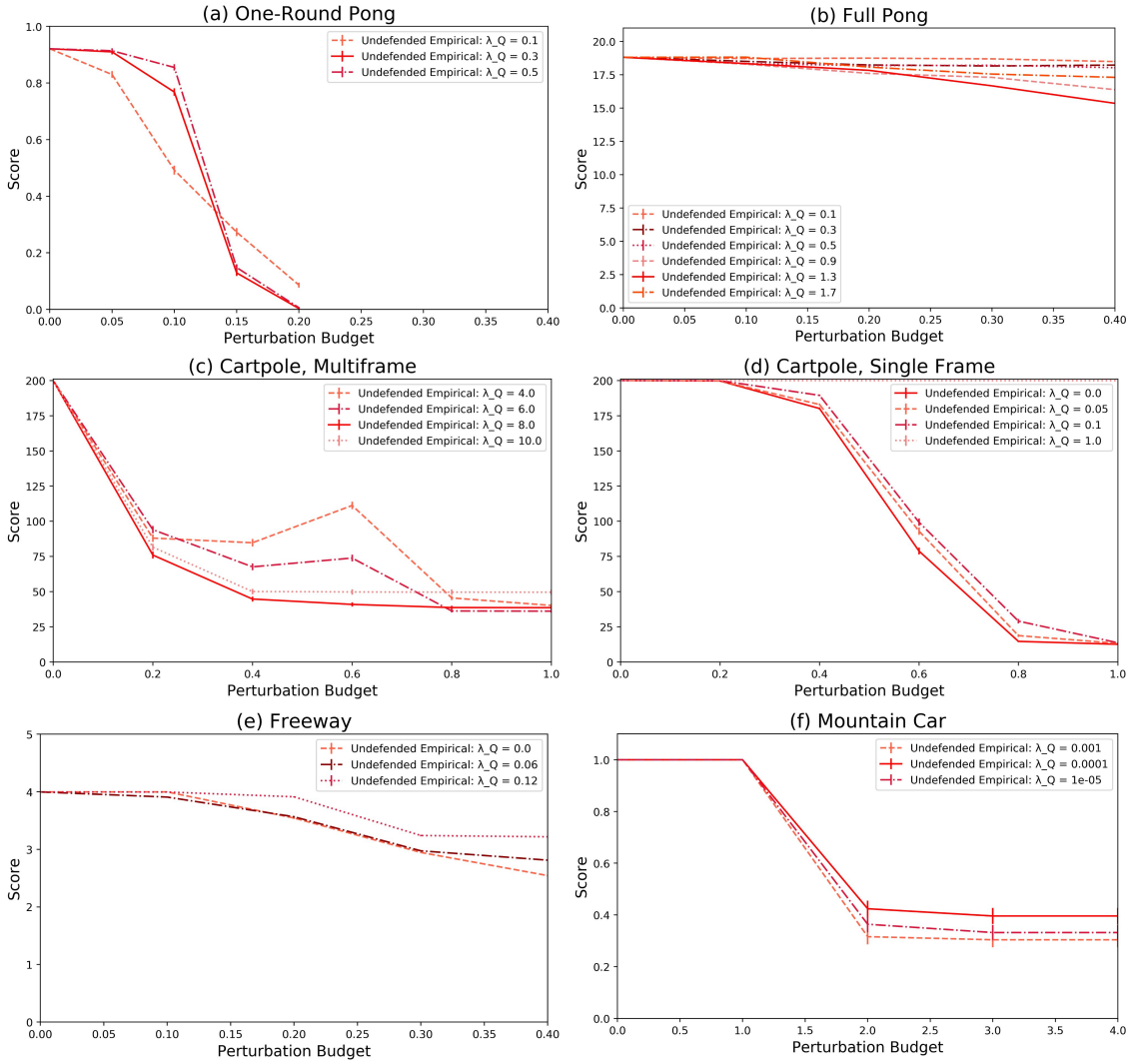


Figure 3.11: Empirical robustness of undefended agents on for all tested values of λ_Q (or λ). The results in the main text are the pointwise minima over λ of these curves.

where 250 episodes were used) and error bars represent the standard error of the mean.

K Empirical Attack Details

Our empirical attack on (undefended) RL observations for DQN is described in Algorithm 3. To summarize, the core of the attack is a standard targeted L_2 PGD attack on the Q-value function. However, because we wish to "save" our total perturbation

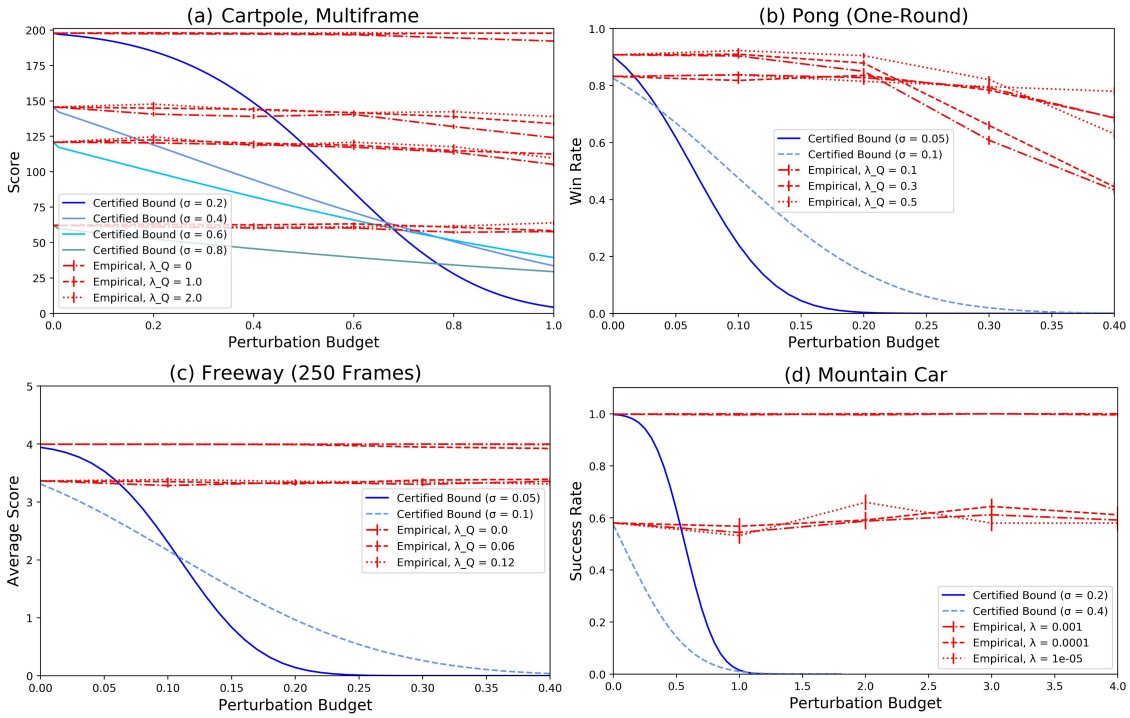


Figure 3.12: Empirical robustness of smoothed agents on for all tested values of σ and λ_Q (or λ). We also plot the associated certificate curves.

budget B for use in later steps, some modifications are made. First, we only target actions a for which the *clean-observation* Q-value is sufficiently below (by a gap given by the parameter λ_Q) the Q-value of the 'best' action, which would be taken in the absence of adversarial attack. Among these possible Targets, we ultimately choose whichever action will maximally decrease the Q-value, and which the agent can be successfully be induced to choose within the adversarial budget B. If no such action exists, then the original observation will be returned, and the entire budget will be saved. In order to preserve budget, the PGD optimization is stopped as soon as the "decision boundary" is crossed.

We use a constant step size η. In order to deal with the variable budget B, we optimize of a number of iterations which is a constant multiple ν of $\frac{B}{\eta}$.

For most environments, there is some context used by the Q-value function (i.e,

the previous frames) which is carried over from previous steps, but is not directly being attacked in this round. We need both the clean version of the context, C, in order to evaluate the "ground-truth" values of the Q-value function under various actions; as well as the "dirty" version of the context, C', based on the adversarial observations which have already been fed to the agent, in order to run the attack optimization.

Our attack for DDPG is described in Algorithm 4. Here, we use the policy π to determine what action a the agent will take when it observes a corrupted observation o' (with corrupted context C'), and use the Q-value function supplied by the DDPG algorithm to determine the "value" of that action on the ground-truth observation o. Because our goal is to minimize this value, this amounts to minimizing $Q(C; o, \pi(C'; o'))$. In order to ensure that a large amount of L_2 "budget" is only used when the Q value can be substantially minimized, we include a regularization term $\lambda \|o - o'\|_2^2$.

Attacks on smoothed agents are described in Appendix M.

Note that on image data (i.e., Pong), we do not consider integrality constraints on the observations; however, we do incorporate box constraints on the pixel values. We also incorporate box constraints on the kinematic quantities when attacking Mountain Car, but not when attacking Cartpole: the distinction is that the constraints in Mountain Car represent artificial constraints on the kinematics [i.e., the velocity of the car is arbitrarily clipped], while the constraints in Cartpole arise naturally from the problem setup.

L Environment details and Hyperparameters

For Atari games, we use the "NoFrameskip-v0" variations of these environments with the standard "AtariPreprocessing" wrapper from the OpenAI Gym [109] package: this provides This environment also injects non-determinism into the originally-deterministic Atari games, by adding randomized "stickiness" to the agent's choice of actions – without this, the state-observation robustness problem could be trivially solved by memorizing a winning sequence of actions, and ignoring all observations at test-time.

Due to instability in training, for the freeway environment, we trained each model five times, and selected the base model based on the performance of validation runs. See training hyperparameters, Tables 3.1 and 3.2. For attack hyperparameters, see Table 3.3 and 3.4.

M Attacks on Smoothed Agents

In order to attack smoothed agents, we adapted Algorithms 3 and 4 using techniques suggested by [39] for attacking smoothed classifiers. In particular, whenever the Q-value function is evaluated or differentiated, we instead evaluate/differentiate the mean output under $m = 128$ smoothing perturbations. Following [39], we use the same noise perturbation vectors at each step during the attack. In the multi-frame case, for the "dirty" context C', we include the actually-realized smoothing perturbations used by the agents for previous steps. However, when determining the "clean" Q-values $Q(C; o, a)$, for the "clean" context C, we use the unperturbed previous state observations: we then take the average over m smoothing perturbations of both C and o to determine the clean Q-values.

	1-Round Pong	Full Pong	Multiframe Cartpole	Single-frame Cartpole	Freeway
Training discount factor γ	0.99	0.99	0.99	0.99	0.99
Total timesteps	10000000	10000000	500000	500000	10000000
Validation interval (steps)	100000	100000	2000	2000	100000
Validation episodes	100	10	10	10	100
Learning Rate	0.0001	0.0001	0.0001	0.00005	0.0001
DQN Buffer Size	10000	10000	100000	100000	10000
DQN steps collected before learning	100000	100000	1000	1000	100000
Fraction of steps for exploration (linearly decreasing exp. rate)	0.1	0.1	0.16	0.16	0.1
Initial exploration rate	1	1	1	1	1
Final exploration rate	0.01	0.01	0	0	0.01
DQN target update interval (steps)	1000	1000	10	10	1000
Batch size	32	32	1024	1024	32
Training interval (steps)	4	4	256	256	4
Gradient descent steps	1	1	128	128	1
Frames Used	4	4	5	1	4
Training Repeats	1	1	1	1	5
Architecture	CNN*	CNN*	MLP $20\times$ $256\times$ $256\times$ 2	MLP $4\times$ $256\times$ $256\times$ 2	CNN*

Table 3.1: Training Hyperparameters for DQN models. *CNN refers to the 3-layer convolutional network defined by the CNNPolicy class in stable-baselines3 [111], based on the CNN architecture used for Atari games by [112]. Note that hyperparameters for Atari games are based on hyperparameters from the stable-baselines3 Zoo package [113], for a slightly different (more deterministic) variant of the Pong environment.

	Mountain Car
Training discount factor γ	0.99
Total timesteps	300000
Validation interval (steps)	2000
Validation episodes	10
Learning Rate	0.0001
DDPG Buffer Size	1000000
DDPG steps collected before learning	100
Batch size	100
Update coefficient τ	0.005
Train frequency	1 per episode
Gradient steps	= episode length
Training action noise	Ornstein Uhlenbeck ($\sigma = 0.5$)
Architecture	MLP $2 \times 400 \times 300 \times 1$

Table 3.2: Training Hyperparameters for DDPG models. Hyperparameters are based on hyperparameters from the stable-baselines3 Zoo package [113], for the unmodified Mountain Car environment.

	1-Round Pong	Full Pong	Multiframe Cartpole	Single-frame Cartpole	Freeway
Attack step size η	0.01	0.01	0.01	0.01	0.01
Attack step multiplier ν	2	2	2	2	2
Q-value thresholds λ_Q searched	.1, .3, .5	.1, .3, .5, .9, 1.3, 1.7	4,6,8,10	0, .05, .1, 1	0, .06, .12

Table 3.3: Attack Hyperparameters for DQN models.

	Mountain Car
Attack step size η	0.01
Attack steps τ	100
Regularization values λ searched	.001, .0001, .00001

Table 3.4: Attack Hyperparameters for DDPG models.

Algorithm 3: Empirical Attack on DQN Agents

Input: Q-value function Q, clean prior observation context C, adversarial prior observation context C', observation o, budget B, Q-value threshold λ_Q, step size η, step multiplier ν

Output: Attacked observation o_{worst}, remaining budget B'.

$Q_{\text{clean}} := \max_{a \in A} Q(C; o, a)$

Targets $:= \{a \in A | Q(C; o, a) \leq Q_{\text{clean}} - \lambda_Q\}$

$Q_{\text{worst}} := Q_{\text{clean}}$

$o_{\text{worst}} := o$

for $a \in$ *Targets* **do**

  $o' := o$

  inner:

  for i *in 1, ...,* $\lfloor \frac{\nu B}{\eta} \rfloor$ **do**

    if $\arg\max_{a'} Q(C'; o', a') = a$ **then**

      if $Q(C; o, a) < Q_{worst}$ **then**

        $o_{\text{worst}} := o'$

        $Q_{\text{worst}} := Q(C; o, a)$

      end

      break inner

    end

    $D := \nabla_{o'} \log([\text{SoftMax}(Q(C'; o', \cdot)]_a)$

    $o' := o' + \frac{\eta D}{\|D\|_2}$

    if $\|o' - o\|_2 > B$ **then**

      $o' := o + \frac{B}{\|o' - o\|_2}(o' - o)$

    end

  end

end

return o_{worst}, $\sqrt{B^2 - \|o_{\text{worst}} - o\|_2^2}$

Algorithm 4: Empirical Attack on DDPG Agents

Input: Q-value function Q, policy π, clean prior observation context C, adversarial prior observation context C', observation o, budget B, weight parameter λ, step size η, step count τ

Output: Attacked observation o_{worst}, remaining budget B'.

$o' := o$

for *i in 1, ..., τ* **do**

  $D := \nabla_{o'}[Q(C; o, \pi(C'; o')) + \lambda \|o' - o\|_2^2]$

  if $\frac{\|D\|_2}{\|o'\|_2} \leq 0.001$ **then**

    break

  end

  $o' := o' + \frac{\eta D}{\|D\|_2}$

  if $\|o' - o\|_2 > B$ **then**

    $o' := o + \frac{B}{\|o' - o\|_2}(o' - o)$

  end

end

return o', $\sqrt{B^2 - \|o' - o\|_2^2}$

This gives an unbiased estimate for the Q-values of an undisturbed smoothed agent in this state.

When attacking DDPG, in evaluating $Q(C; o, \pi(C'; o')$, we average over smoothing perturbations for both o and o', in addition to C: this is because both π and Q are trained on noisy samples. Note that we use independently-sampled noise perturbations on o' and o.

Our attack does not appear to be successful, compared with the lower bound given by our certificate (Figures 3.12). One contributing factor may be that attacking a smoothed *agent* is more difficult that attacking a smoothed *classifier*, for the following reason: a smoothed classifier evaluates the expected output at test time, while a smoothed agent does not. Thus, while the *average* Q-value for the targeted action might be greater than the *average* Q-value for the clean action, the actual realization will depend on the specific realization of the random smoothing vector that the agent actually uses.

N Runtimes and Computational Environment

Each experiment is run on an NVIDIA 2080 Ti GPU. Typical training times are shown in Table 3.5. Typical clean evaluation times are shown in Table 3.6. Typical attack times are shown in Table 3.7.

Experiment	Time (hours)
Pong (1-round)	11.1
Pong (Full)	12.0
Cartpole (Multi-frame)	0.27
Cartpole (Single-frame)	0.32
Freeway	14.2
Mountain Car	0.63

Table 3.5: Training times

Experiment Experiment	Time (seconds): smallest noise σ	Time (seconds): largest noise σ
Pong (1-round)	0.46	0.38
Pong (Full)	3.82	4.65
Cartpole (Multi-frame)	0.20	0.13
Cartpole (Single-frame)	0.18	0.12
Freeway	1.36	1.35
Mountain Car	0.67	0.91

Table 3.6: Evaluation times. Note that the times reported here are *per episode*: in order to statistically bound the mean rewards, we performed 10,000 such episode evaluations for each environment.

Experiment Experiment	Time (seconds): smallest budget B	Time (seconds): largest budget B
Pong (1-round)	1.01	0.68
Pong (Full)	8.84	10.2
Cartpole (Multi-frame)	0.35	0.32
Cartpole (Single-frame)	0.79	0.56
Freeway	2.67	2.80
Mountain Car	44.0	19.6

Table 3.7: Attack times. Note that the times reported here are *per episode*: in the paper, we report the mean of 1000 such episodes.

O CDF Smoothing Details

Due to the very general form of our certification result ($h(\cdot)$, as a 0/1 function, can represent any outcome, and we can bound the lower-bound the probability of this outcome), there are a variety of ways we can use the basic result to compute a certificate for an entire episode entire game. In the main text, we introduce CDF smoothing [51] as one such option. In CDF smoothing for any threshold value x, we can define $h_x(\cdot)$ as an indicator function for the event that the total episode reward is greater than x. Then, by the definition of the CDF function, the expectation of $h_x(\cdot)$ is equal to $1 - F(x)$, where $F(\cdot)$ is the CDF function of the reward. Then our lower-bound on the expectation of $h_x(\cdot)$ under adversarial attack is in fact an upper-bound on $F(x)$: combining this with Equation 2 in the main text,

$$E[\mathcal{X}] = \int_0^\infty (1 - F(x))dx - \int_{-\infty}^0 F(x)dx,$$

provides a lower bound on the total expectation of the reward under adversarial perturbation.

However, in order to perform this integral from empirical samples, we must bound $F(x)$ at all points: this requires first upper-bounding the *non-adversarial* CDF function at all x, before applying our certificate result. Following [51], we accomplish this using the Dvoretzky–Kiefer–Wolfowitz inequality (for the Full Pong environment.)

In the case of the Cartpole environment, we explore a different strategy: note that the reward at each timestep is itself a 0/1 function, so we can define $h_t(\cdot)$ as simply the reward at timestep t. We can then apply our certificate result at each timestep independently,

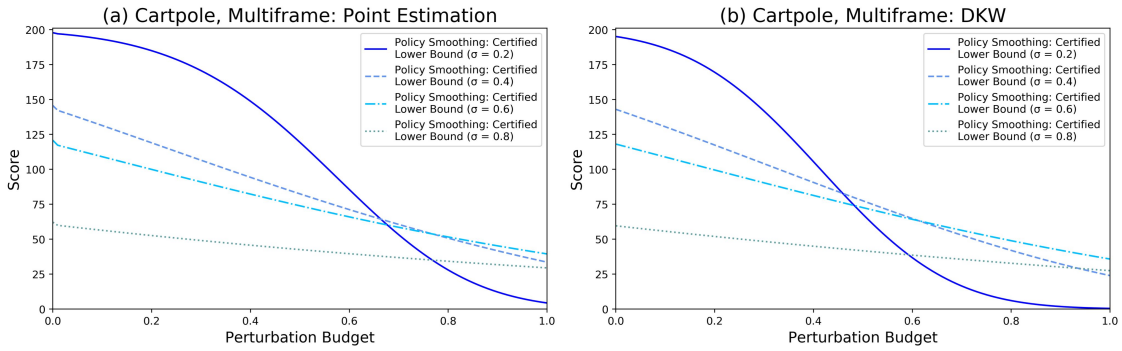


Figure 3.13: Comparison of certified bounds on the total reward in Cartpole, using (a) point estimation, and (b) the DKW inequality to generate empirical bounds.

and take a sum. Note that this requires estimating the average reward at each step independently: we use the Clopper-Pearson method (following [36]), and in order to certify in total to the desired 95% confidence bound, we certify each estimate to (100 - 5/T)% confidence, where T is the total number of timesteps per episode (= 200).

However, note that, in the particular case of the cartpole environment, $h_t(\cdot) = 1$ if and only if we have "survived" to time-step t: in other words, $h_t(\cdot)$ is simply an indicator function for the total reward being $\geq t$. Therefore in this case, this independent estimation method is equivalent to CDF smoothing, just using Clopper-Pearson point-estimates of the CDF function rather than the Dvoretzky–Kiefer–Wolfowitz inequality. In practice, we find that this produced slightly better certificates for this task. (Figure 3.13)

P Environment Licenses

OpenAI Gym [109] is Copyright 2016 by OpenAI and provided under the MIT License. The stable-baselines3 package[111] is Copyright 2019 by Antonin Raffin and also provided under the MIT License.

Chapter 4: Center Smoothing

4.1 Introduction

The study of adversarial robustness in machine learning (ML) has gained a lot of attention ever since deep neural networks (DNNs) have been demonstrated to be vulnerable to adversarial attacks. These attacks are generated by making tiny perturbations of the input that can completely alter a model's predictions [1, 2, 3, 4]. They can significantly degrade the performance of a model, like an image classifier, and make it output almost any class of the attacker's choice. However, these attacks are not limited just to classification problems. They have also been shown to exist for DNNs with structured outputs like text, images, probability distributions, sets, etc. For instance, automatic speech recognition systems can be attacked with 100% success rate to output any phrase of the attackers choice [10]. Similar attacks can cause neural image captioning systems to produce specific target captions with high success-rate [9]. Quality of image segmentation models have been shown to degrade severely under adversarial attacks [114, 115, 116]. Facial recognition systems can be deceived to evade detection, impersonate authorized individuals and even render them completely ineffective [117, 118, 119]. Image reconstruction models have been targeted to introduce unwanted artefacts or miss important details, such as tumors in MRI scans, through adversarial inputs [11, 12, 13, 14]. Super-resolution systems can

be made to generate distorted images that can in turn deteriorate the performance of subsequent tasks that rely on the high-resolution outputs [16, 17]. Deep neural network based policies in reinforcement learning problems also have been shown to succumb to imperceptible perturbations in the state observations [5, 6, 7, 8]. Such widespread presence of adversarial attacks is concerning as it threatens the use of deep neural networks in critical systems, such as facial recognition, self-driving vehicles, medical diagnosis, etc., where safety, security and reliability are of utmost importance.

Adversarial defenses have mostly focused on classification tasks [18, 19, 20, 21, 22, 23, 98]. Certified defenses based on convex-relaxation [27, 28, 29, 30, 31], interval-bound propagation [32, 33, 34, 35] and randomized smoothing [36, 37, 38, 39] that guarantee that the predicted class will remain the same in a certified region around the input point have also been studied. Compared to empirical robustness methods that are often shown to be broken by stronger attacks [24, 25, 26], procedures with provable robustness guarantees are of special importance to the study of robustness in ML as their guarantees hold regardless of improvements in attack strategies. Among these approaches, certified defenses based on randomized smoothing have been show to scale up to high-dimensional inputs, such as images, and does not need to make assumptions about the underlying model. The robustness certificates produced by these defenses are probabilistic, meaning that they hold with high probability and not absolute certainty.

Unlike classification problems, where certificates guarantee that the predicted class remains unchanged under bounded-size perturbations, it is not immediately obvious what the goal of robustness should be for problems with structured outputs like images, text, sets, etc. While accuracy is the standard quality measure for classification, more complex

tasks may require other quality metrics like total variation for images, intersection over union for object localization, earth-mover distance for distributions, etc. In general, neural networks can be cast as functions of the type $f : \mathbb{R}^k \to (M, d)$ which map a k dimensional real-valued space into a metric space M with distance function $d : M \times M \to \mathbb{R}_{\geq 0}$. In this work, we design a randomized smoothing based technique to obtain provable robustness for functions of this type with minimal assumptions on the distance metric d. We generate a robust version $\bar{f}$ such that the change in its output, as measured by d, is small for a small change in its input. More formally, given an input x and an ℓ_2-perturbation size ϵ_1, we produce a value ϵ_2 with the guarantee that, with high probability,

$$\forall x' \text{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ d(\bar{f}(x), \bar{f}(x')) \leq \epsilon_2.$$

Our contributions: We develop *center smoothing*, a procedure to make functions like f provably robust against adversarial attacks. For a given input x, center smoothing samples a collection of points in the neighborhood of x using a Gaussian smoothing distribution, computes the function f on each of these points and returns the center of the smallest ball enclosing at least half the points in the output space (see figure 4.1). Computing the minimum enclosing ball in the output space is equivalent to solving the 1-center problem with outliers (hence the name of our

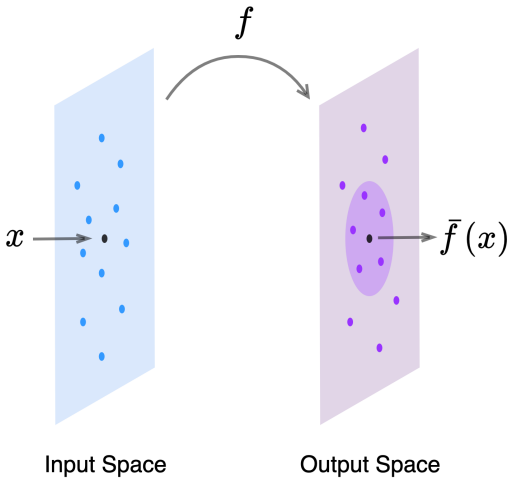


Figure 4.1: Center smoothing.

procedure), which is an NP-complete problem for a general metric [120]. We approximate it by computing the point that has the smallest median distance to all the other points in the sample. We show that the output of the smoothed function is robust to input perturbations of bounded ℓ_2-size. We restrict the input perturbations to be inside an ℓ_2-ball as the main focus of this work is on the output space of f. However, our method does not critically rely on the ℓ_2 threat model or Gaussian smoothing noise, and can be adapted to other perturbations types and smoothing distributions. Although we define the output space as a metric, our proofs only require the symmetry property and triangle inequality to hold. Thus, center smoothing can also be applied to pseudometric distances that need not satisfy the identity of indiscernibles. Many distances defined for images, such as total variation, cosine distance, perceptual distances, etc., fall under this category. Center smoothing steps outside the world of ℓ_p metrics, and certifies robustness in metrics like IoU/Jaccard distance for object localization, and total-variation, which is a good measure of perceptual similarity for images. In our experiments, we show that this method can produce meaningful certificates for a wide variety of output metrics without significantly compromising the quality of the base model.

Related Work: Randomized smoothing has been extensively used for provable adversarial robustness in the classification setting to defend against different ℓ_p [36, 37, 39, 69, 121, 122, 123, 124] and non-ℓ_p [68, 82] threat models. Beyond classification, it has also been used for certifying the median output of regression models [125] and the expected softmax scores of neural networks [51]. Smoothing a bounded vector-valued function by taking the mean of the output vectors has been shown to have a bounded Lipschitz constant when both input and output spaces are ℓ_2-metrics [126].

Center smoothing does not require the base function to be bounded because the minimum enclosing ball is resistant to outliers. Moving an outlier point away from this ball does not affect the output of the smoothed function. On the other hand, smoothing techniques that compute the mean of the output samples are more susceptible to outliers as changing any of the samples can alter the mean. Recently, a provable defense for segmentation tasks was developed by certifying each individual pixel of the output using randomized smoothing [127]. Due to the accumulating uncertainty over individual certifications, it is difficult to produce guarantees for large images, often leading to certified outputs with ambiguous pixels. Center smoothing bypasses this challenge by directly certifying the similarity between a clean segmentation output and an adversarial one under a metric such as intersection over union.

4.2 Preliminaries and Notations

Given a function $f : \mathbb{R}^k \to (M, d)$ and a distribution $\mathcal{D}$ over the input space $\mathbb{R}^k$, let $f(\mathcal{D})$ denote the probability distribution of the output of f in M when the input is drawn from $\mathcal{D}$. For a point $x \in \mathbb{R}^k$, let $x + \mathcal{P}$ denote the probability distribution of the points $x + \delta$ where δ is a smoothing noise drawn from a distribution $\mathcal{P}$ over $\mathbb{R}^k$ and let X be the random variable for $x + \mathcal{P}$. For elements in M, define $\mathcal{B}(z, r) = \{z' \mid d(z, z') \leq r\}$ as a ball of radius r centered at z. Define a smoothed version of f under $\mathcal{P}$ as the center of the ball with the smallest radius in M that encloses at least half of the probability mass of $f(x + \mathcal{P})$, i.e.,

$$\bar{f}_{\mathcal{P}}(x) = \underset{z}{\operatorname{argmin}}\ r \text{ s.t. } \mathbb{P}[f(X) \in \mathcal{B}(z, r)] \geq \frac{1}{2}.$$

If there are multiple balls with the smallest radius satisfying the above condition, return one of the centers arbitrarily. Let $r^*_{\mathcal{P}}(x)$ be the value of the minimum radius. Hereafter, we ignore the subscripts and superscripts in the above definitions whenever they are obvious from context. In this work, we sample the noise vector δ from an i.i.d Gaussian distribution of variance σ^2 in each dimension, i.e., $\delta \sim \mathcal{N}(0, \sigma^2 I)$.

4.2.1 Gaussian Smoothing

Cohen et al. in 2019 showed that a classifier $h : \mathbb{R}^k \to \mathcal{Y}$ smoothed with a Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ as,

$$\bar{h}(x) = \underset{c \in \mathcal{Y}}{\text{argmax}}\ \mathbb{P}\left[h(x + \delta) = c\right],$$

where $\mathcal{Y}$ is a set of classes, is certifiably robust to small perturbations in the input. Their certificate relied on the fact that, if the probability of sampling from the top class at x under the smoothing distribution is p, then for an ℓ_2 perturbation of size at most ϵ, the probability of the top class is guaranteed to be at least

$$p_\epsilon = \Phi(\Phi^{-1}(p) - \epsilon/\sigma), \tag{4.1}$$

where Φ is the CDF of the standard normal distribution $\mathcal{N}(0, 1)$. This bound applies to any $\{0, 1\}$-function over the input space $\mathbb{R}^k$, i.e., if $\mathbb{P}[h(x) = 1] = p$, then for any ϵ-size perturbation $x', \mathbb{P}[h(x') = 1] \geq p_\epsilon$.

We use this bound to generate robustness certificates for center smoothing. We identify a ball $\mathcal{B}(\bar{f}(x), R)$ of radius R enclosing a very high probability mass of the

output distribution. One can define a function that outputs one if f maps a point to inside $\mathcal{B}(\bar{f}(x), R)$ and zero otherwise. The bound in (4.1) gives us a region in the input space such that for any point inside it, at least half of the mass of the output distribution is enclosed in $\mathcal{B}(\bar{f}(x), R)$. We show in section 4.3 that the output of the smoothed function for a perturbed input is guaranteed to be within a constant factor of R from the output of the original input.

4.3 Center Smoothing

As defined in section 4.2, the output of $\bar{f}$ is the center of the smallest ball in the output space that encloses at least half the probability mass of the $f(x + \mathcal{P})$. Thus, in order to significantly change the output, an adversary has to find a perturbation such that a majority of the neighboring points map far away from $\bar{f}(x)$. However, for a function that is roughly accurate on most points around x, a small perturbation in the input cannot change the output of the smoothed function by much, thereby making it robust.

For an ℓ_2 perturbation size of ϵ_1 of an input point x, let R be the radius of a ball around $\bar{f}(x)$ that encloses more than half the probability mass of $f(x' + \mathcal{P})$ for all x' satisfying $\|x - x'\|_2 \leq \epsilon_1$, i.e.,

$$\forall x' \text{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ \mathbb{P}[f(X') \in \mathcal{B}(\bar{f}(x), R)] > \frac{1}{2}, \tag{4.2}$$

where $X' \sim x' + \mathcal{P}$. Basically, R is the radius of a ball around $\bar{f}(x)$ that contains at least half the probability mass of $f(x' + \mathcal{P})$ for any ϵ_1-size perturbation x' of x. Then, we have the following robustness guarantee on $\bar{f}$:

Theorem 3. *For all x' such that $\|x - x'\|_2 \leq \epsilon_1$,*

$$d(\bar{f}(x), \bar{f}(x')) \leq 2R.$$

Proof. Consider the balls $\mathcal{B}(\bar{f}(x'), r^*(x'))$ and $\mathcal{B}(\bar{f}(x), R)$ (see figure 4.2). From the definition of $r^*(x')$ and R, we know that the sum of the probability masses of $f(x' + \mathcal{P})$ enclosed by the two balls must be strictly greater than one. Thus, they must have an element y in common. Since d satisfies the triangle inequality, we have:

$$\begin{aligned} d(\bar{f}(x), \bar{f}(x')) &\leq d(\bar{f}(x), y) + d(y, \bar{f}(x')) \\ &\leq R + r^*(x'). \end{aligned}$$

Since, the ball $\mathcal{B}(\bar{f}(x), R)$ encloses more than half of the probability mass of $f(x + \mathcal{P})$, the minimum ball with at least half the probability mass cannot have a radius greater than R, i.e., $r^*(x') \leq R$. Therefore, $d(\bar{f}(x), \bar{f}(x')) \leq 2R$. □

The above result, in theory, gives us a smoothed version of f with a provable guarantee of robustness. However, in practice, it may not be feasible to obtain $\bar{f}$ just from samples of $f(x + \mathcal{P})$. Instead, we will use some procedure that approximates the smoothed output with high probability. For some $\Delta \in [0, 1/2]$, let $\hat{r}(x, \Delta)$ be the radius of

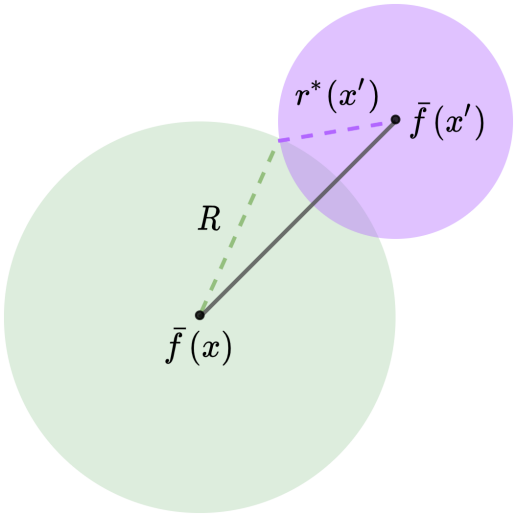


Figure 4.2: Robustness guarantee.

the smallest ball that encloses at least $1/2 + \Delta$

probability mass of $f(x + \mathcal{P})$, i.e.,

$$\hat{r}(x, \Delta) = \min_{z'} r \text{ s.t. } \mathbb{P}[f(X) \in \mathcal{B}(z', r)] \geq \frac{1}{2} + \Delta.$$

Now define a probabilistic approximation $\hat{f}(x)$ of the smoothed function $\bar{f}$ to be a point $z \in M$, which with probability at least $1 - \alpha_1$ (for $\alpha_1 \in [0, 1]$), encloses at least $1/2 - \Delta$ probability mass of $f(x + \mathcal{P})$ within a ball of radius $\hat{r}(x, \Delta)$. Formally, $\hat{f}(x)$ is a point $z \in M$, such that, with at least $1 - \alpha_1$ probability,

$$\mathbb{P}\left[f(X) \in \mathcal{B}(z, \hat{r}(x, \Delta))\right] \geq \frac{1}{2} - \Delta.$$

Defining $\hat{R}$ to be the radius of a ball centered at $\hat{f}(x)$ that satisfies:

$$\forall x' \text{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ \mathbb{P}[f(X') \in \mathcal{B}(\hat{f}(x), \hat{R})] > \frac{1}{2} + \Delta, \tag{4.3}$$

we can write a probabilistic version of theorem 3,

Theorem 4. *With probability at least* $1 - \alpha_1$*,*

$$\forall x' \textit{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ d(\hat{f}(x), \hat{f}(x')) \leq 2\hat{R},$$

The proof of this theorem is in the appendix, and logically parallels the proof of theorem 3.

4.3.1 Computing $\hat{f}$

For an input x and a given value of Δ, sample n points independently from a Gaussian distribution $x + \mathcal{N}(0, \sigma^2 I)$ around the point x and compute the function f on each of these points. Let $Z = \{z_1, z_2, \ldots, z_n\}$ be the set of n samples of $f(x+\mathcal{N}(0, \sigma^2 I))$ produced in the output space. Compute the minimum enclosing ball $\mathcal{B}(z, r)$ that contains at least half of the points in Z. The following lemma bounds the radius r of this ball by the radius of the smallest ball enclosing at least $1/2 + \Delta_1$ probability mass of the output distribution (proof in appendix).

Lemma 9. *With probability at least* $1 - e^{-2n\Delta_1^2}$,

$$r \leq \hat{r}(x, \Delta_1).$$

Now, sample a fresh batch of n random points. Let $p_{\Delta_1} = \rho - \Delta_1$, where ρ is the fraction of points that fall inside $\mathcal{B}(z, r)$. Then, by Hoeffding's inequality, with probability at least $1 - e^{-2n\Delta_1^2}$,

$$\mathbb{P}\left[f(X) \in \mathcal{B}(z, r)\right] \geq p_{\Delta_1}.$$

Let $\Delta_2 = 1/2 - p_{\Delta_1}$. If $\max(\Delta_1, \Delta_2) \leq \Delta$, the point z satisfies the conditions in the definition of $\hat{f}$, with at least $1 - 2e^{-2n\Delta_1^2}$ probability. If $\max(\Delta_1, \Delta_2) > \Delta$, discard the computed center z and abstain. In our experiments, we select Δ_1, n and α_1 appropriately so that the above process succeeds easily.

Computing the minimum enclosing ball $\mathcal{B}(z, r)$ exactly can be computationally challenging, as for certain metrics, it is known to be NP-complete [120]. Instead, we

Algorithm 5: Smooth

Input: $x \in \mathbb{R}^k, \sigma, \Delta, \alpha_1$.
Output: $z \in M$.
Set
$Z = \{z_i\}_{i=1}^n$ s.t. $z_i \sim f(x + \mathcal{N}(0, \sigma^2 I))$.

Set $\Delta_1 = \sqrt{\ln(2/\alpha_1)/2n}$.
Compute $z = \beta$-MEB$(Z, 1/2)$.
Re-sample Z.
Compute p_{Δ_1}.
Set $\Delta_2 = 1/2 - p_{\Delta_1}$.
If $\Delta < \max(\Delta_1, \Delta_2)$, discard z and abstain.

Algorithm 6: Certify

Input: $x \in \mathbb{R}^k, \epsilon_1, \sigma, \Delta, \alpha_1, \alpha_2$.
Output: $\epsilon_2 \in \mathbb{R}$.
Compute $\hat{f}(x)$ using algorithm 5.
Set
$Z = \{z_i\}_{i=1}^m$ s.t. $z_i \sim f(x + \mathcal{N}(0, \sigma^2 I))$.

Compute
$\tilde{\mathcal{R}} = \{d(\hat{f}(x), f(z_i)) \mid z_i \in Z\}$.
Set $p = \Phi(\Phi^{-1}(1/2 + \Delta) + \epsilon_1/\sigma)$.
Set $q = p + \sqrt{\ln(1/\alpha_2)/2m}$.
Set $\hat{R} = q$th-quantile of $\tilde{\mathcal{R}}$.
Set $\epsilon_2 = (1 + \beta)\hat{R}$.

approximate it by computing a ball β-MEB$(Z, 1/2)$ that contains at least half the points in Z, but has a radius that is within a β factor of the optimal radius r. We modify theorem 3 to account for this approximation (see appendix for proof).

Theorem 5. *With probability at least* $1 - \alpha_1$*,*

$$\forall x' \textit{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ d(\hat{f}(x), \hat{f}(x')) \leq (1 + \beta)\hat{R}$$

where $\alpha_1 = 2e^{-2n\Delta_1^2}$.

We use a simple approximation that works for all metrics and achieves an approximation factor of two, producing a certified radius of $3\hat{R}$. It computes a point from the set Z, instead of a general point in M, that has the minimum median distance from all the points in the set (including itself). This can be achieved using $O(n^2)$ pair-wise distance computations. To see how the factor 2-approximation is achieved, consider the optimal ball with radius r. By triangle inequality of d, each pair of points is at most $2r$ distance from each other. Thus, a ball with radius $2r$, centered at any one of these points will

cover every other point in the optimal ball. Better approximations can be obtained for specific norms, e.g., there exists a $(1+\epsilon)$-approximation algorithm for the ℓ_2 norm [128]. For graph distances or when the support of the output distribution is a small discrete set of points, the optimal radius can be computed exactly using the above algorithm. The smoothing procedure is outlined in algorithm 5.

4.3.2 Certifying $\hat{f}$

Given an input x, compute $\hat{f}(x)$ as described above. Now, we need to compute a radius $\hat{R}$ that satisfies condition 4.3. As per bound 4.1, in order to maintain a probability mass of at least $1/2+\Delta$ for any ϵ_1-size perturbation of x, the ball $\mathcal{B}(\hat{f}(x), \hat{R})$ must enclose at least

$$p = \Phi\left(\Phi^{-1}\left(\frac{1}{2} + \Delta\right) + \frac{\epsilon_1}{\sigma}\right) \tag{4.4}$$

probability mass of $f(x+\mathcal{P})$. Again, just as in the case of estimating $\bar{f}$, we may only compute $\hat{R}$ from a finite number of samples m of the distribution $f(x+\mathcal{P})$. For each sample $z_i \sim x+\mathcal{P}$, we compute the distance $d(\hat{f}(x), f(z_i))$ and set $\hat{R}$ to be the qth-quantile $\tilde{R}_q$ of these distances for a q that is slightly greater than p (see equation 4.5 below). The qth-quantile $\tilde{R}_q$ is a value larger than at least q fraction of the samples. We set q as,

$$q = p + \sqrt{\frac{\ln(1/\alpha_2)}{2m}}, \tag{4.5}$$

for some small $\alpha_2 \in [0, 1]$. This guarantees that, with high probability, the ball $\mathcal{B}(\hat{f}(x), \tilde{R}_q)$ encloses at least p fraction of the probability mass of $f(x + \mathcal{P})$. We prove the following lemma by bounding the cumulative distribution function of the distances of $f(z_i)$s from $\hat{f}(x)$ using the Dvoretzky–Kiefer–Wolfowitz inequality.

Lemma 10. *With probability* $1 - \alpha_2$,

$$\mathbb{P}\left[f(X) \in \mathcal{B}(\hat{f}(x), \tilde{R}_q)\right] > p.$$

Combining with theorem 5, we have the final certificate:

$$\forall x' \text{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ d(\hat{f}(x), \hat{f}(x')) \leq (1 + \beta)\hat{R},$$

with probability at least $1 - \alpha$, for $\alpha = \alpha_1 + \alpha_2$. In our experiments, we set $\alpha_1 = \alpha_2 = 0.005$ to achieve an overall success probability of $1 - \alpha = 0.99$, and calculate the required Δ_1, Δ_2 and q values accordingly. We set Δ to be as small as possible without violating $\max(\Delta_1, \Delta_2) \leq \Delta$ too often. We use a $\beta = 2$-approximation for computing the minimum enclosing ball in the smoothing step. Algorithm 6 provides the pseudocode for the certification procedure.

4.4 Relaxing Metric Requirements

Although we defined our procedure for metric outputs, our analysis does not critically use all the properties of a metric. For instance, we do not require $d(z_1, z_2)$ to be strictly greater than zero for $z_1 \neq z_2$. An example of such a distance measure is the total

variation distance that returns zero for two vectors that differ by a constant amount on each coordinate. Our proofs do implicitly use the symmetry property, but asymmetric distances can be converted to symmetric ones by taking the sum or the max of the distances in either directions. Perhaps the most important property of metrics that we use is the triangle inequality as it is critical for the robustness guarantee of the smoothed function. However, even this constraint may be partially relaxed. It is sufficient for the distance function d to satisfy the triangle inequality approximately, i.e., $d(a, c) \leq \gamma(d(a, b) + d(b, c))$, for some constant γ. The theorems and lemmas can be adjusted to account for this approximation, e.g., the bound in theorem 3 will become $2\gamma R$. A commonly used distance measure for comparing images and documents is the cosine distance defined as the inner-product of two vectors after normalization. This distance can be show to be proportional to the squared Euclidean distance between the normalized vectors which satisfies the relaxed version of triangle inequality for $\gamma = 2$.

These relaxations extend the scope of center smoothing to many commonly used distance measures that need not necessarily satisfy all the metric properties. For instance, perceptual distance metrics measure the distance between two images in some feature space rather than image space. Such distances align well with human judgements when the features are extracted from a deep neural network [129] and are considered more natural measures for image similarity. For two images I_1 and I_2, let $\phi(I_1)$ and $\phi(I_2)$ be their feature representations. Then, for a distance function d in the feature space that satisfies the relaxed triangle inequality, we can define a distance function $d_\phi(I_1, I_2) = d(\phi(I_1), \phi(I_2))$ in the image space, which also satisfies the relaxed triangle inequality.

For any image I_3,

$$\begin{aligned} d_\phi(I_1, I_2) &= d(\phi(I_1), \phi(I_2)) \\ &\leq \gamma \left(d(\phi(I_1), \phi(I_3)) + d(\phi(I_3), \phi(I_2)) \right) \\ &= \gamma \left(d_\phi(I_1, I_3) + d_\phi(I_3, I_2) \right). \end{aligned}$$

4.5 Experiments

We apply center smoothing to certify a wide range of output metrics: Jaccard distance based on intersection over union (IoU) of sets, total variation distances for images, and perceptual distance. We certify the bounding box generated by a face detector – a key component of most facial recognition systems – by guaranteeing the minimum overlap (measured using IoU) it must have with the output under an adversarial perturbation of the input. For instance, if $\epsilon_1 = 0.2$, the Jaccard distance (1-IoU) is guaranteed to be bounded by 0.2, which implies that the bounding box of a perturbed image must have at least 80% overlap with that of the clean image. We use a pre-trained face detection model for this experiment. We certify the perceptual distance of the output of a generative model (trained on ImageNet) that produces 128×128 RGB images using a high-dimensional version of the smoothing procedure Smooth-HD described in the appendix. For total variation distance, we use simple, easy-to-train convolutional neural network based dimensionality reduction (autoencoder) and image reconstruction models. Our goal is to demonstrate the effectiveness of our method for a wide range of applications and so, we place less emphasis on the performance of the underlying models being smoothed. In each case, we

show that our method is capable of generating certified guarantees without significantly degrading the performance of the underlying model. We provide additional experiments for other metrics and parameter settings in the appendix.

As is common in the randomized smoothing literature, we train our base models (except for the pre-trained ones) on noisy data with different noise levels $\sigma_{train} = 0.1, 0.2, \ldots, 0.5$ to make them more robust to input perturbations. We keep the smoothing noise σ of the robust model same as the training noise σ_{train} of the base model. We use $n = 10^4$ samples to estimate the smoothed function and $m = 10^6$ samples to generate certificates, unless stated otherwise. We set $\Delta = 0.05, \alpha_1 = 0.005$ and $\alpha_2 = 0.005$ as discussed in previous sections. We grow the smoothing noise σ linearly with the input perturbation ϵ_1. Specifically, we maintain $\epsilon_1 = h\sigma$ for different values of $h = 2, 1$ and 1.5 in our experiments. We plot the median certified output radius ϵ_2 and the median smoothing error, defined as the distance between the outputs of the base model and the smoothed model $d(f(x), \hat{f}(x))$, of fifty random test examples for different values of ϵ_1. In all our experiments, we observe that both these quantities increase as the input radius ϵ_1 increases, but the smoothing error remains significantly below the certified output radius. Also, increasing the value of h improves the quality of the certificates (lower ϵ_2). This could be due to the fact that for a higher h, the smoothing noise σ is lower (keeping ϵ_1 constant), which means that the radius of the minimum enclosing ball in the output space is smaller leading to a tighter certificate. However, setting h too high can cause the value of q in equation 4.5 to exceed one (q depends on p, which in turn depends on h in eq. 4.4), leading the certification procedure (algorithm 6) to fail. We ran all our experiments on a single NVIDIA GeForce RTX 2080 Ti GPU in an internal cluster. Each of the fifty

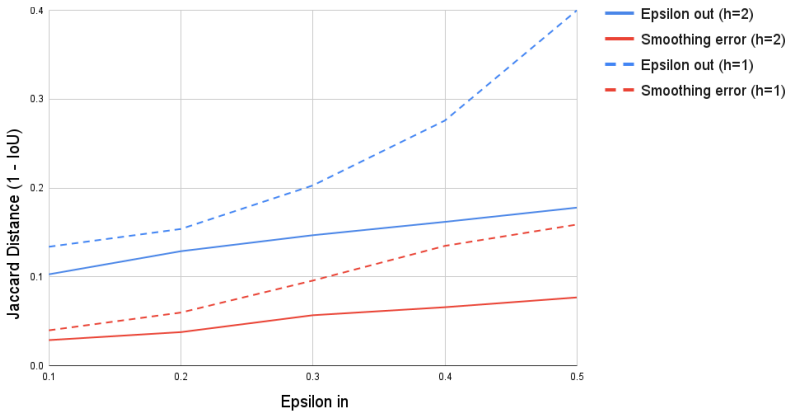


Figure 4.3: Certifying Jaccard Distance (1 - IoU).

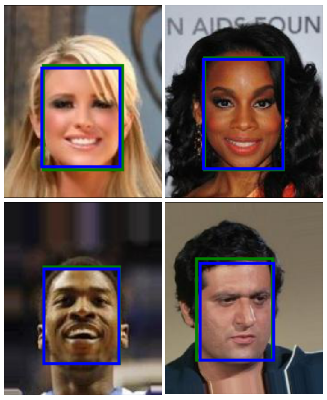


Figure 4.4: Smoothed Output.

Figure 4.5: Face Detection on CelebA using MTCNN detector: Part (a) plots the certified output radius ϵ_2 and the smoothing error for $h = 1$ and 2. Part (b) compares the smoothed output (blue box) to the output of the base model (green box, mostly hidden behind the blue box) showing a significant overlap.

examples we certify took somewhere between 1-3 minutes depending on the underlying model.

4.5.1 Jaccard distance

It is known that facial recognition systems can be deceived to evade detection, impersonate authorized individuals and even render completely ineffective [117, 118, 119]. Most facial recognition systems first detect a region that contains a persons face,

e.g. a bounding box, and then uses facial features to identify the individual in the image. To evade detection, an attacker may seek to degrade the quality of the bounding boxes produced by the detector and can even cause it to detect no box at all. Bounding boxes are often interpreted as sets and the their quality is measured as the amount of overlap with the desired output. When no box is output, we say the overlap is zero. The overlap between two sets is defined as the ratio of the size of the intersection between them to the size of their union (IoU). Thus, to certify the robustness of the output of a face detector, it makes sense to bound the worst-case IoU of the output of an adversarial input to that of a clean input. The corresponding distance function, known as Jaccard distance, is defined as $1 - IoU$ which defines a metric over the universe of sets.

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad d_J(A, B) = 1 - IoU(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}.$$

In this experiment, we certify the output of a pre-trained face detection model MTCNN [130] on the CelebA face dataset [131]. We set $n = 5000$ and $m = 10000$, and use default values for other parameters discussed above. Figure 4.3 plots the certified output radius ϵ_2 and the smoothing error for $h = \epsilon_1/\sigma = 1$ and 2 for $\epsilon_1 = 0.1, 0.2, \ldots, 0.5$. Certifying the Jaccard distance allows us to certify IoU as well, e.g., for $h = 2$, ϵ_2 is consistently below 0.2 which means that even the worst bounding box under adversarial perturbation of the input has an overlap of at least 80% with the box for the clean input. The low smoothing error shows that the performance of the base model does not drop significantly as the actual output of the smoothed model has a large overlap with that of the base model. Figure 4.4 compares the outputs of the smoothed model (blue box) and

the base model (green box). For most of the images, the blue box overlaps with the green one almost perfectly.

4.5.2 Perceptual Distance

Deep generative models like GANs and VAEs have been shown to be vulnerable to adversarial attacks [15]. One attack model is to produce an adversarial example that is close to the original input in the latent space, measured using ℓ_2-norm. The goal is to make the model generate a different looking image using a latent representation that is close to that of the original image. We apply center smoothing to a generative adversarial network BigGAN pre-trained on ImageNet images [132]. We use the version of the GAN that generates 128×128 resolution ImageNet images from a set of 128 latent variables. Since we are interested in producing similar looking images for similar latent representations, a good output metric would be the perceptual distance between two images measured by LPIPS metric [129]. This distance function takes in two images, passes them through a deep neural network, such as VGG, and computes a weighted sum of the square of the differences of the activations (after some normalization) produced by the two images. The process can be thought of as generating two feature vectors ϕ_1 and ϕ_2 for the two input images I_1 and I_2 respectively, then computing a weighted sum of the element-wise square of the differences between the two feature vectors, i.e.,

$$d(I_1, I_2) = \sum_i w_i(\phi_{1i} - \phi_{2i})^2$$

The square of differences metric can be shown to follow the relaxed triangle inequality for $\gamma = 2$. Therefore, the the final bound on the certified output radius will be $\gamma(1 + 2\gamma)\hat{R} = 10\hat{R}$. Figure 4.6 plots the median smoothing error and certified output radius ϵ_2 for fifty randomly picked latent vectors for $\epsilon_1 = 0.01, 0.02, \ldots, 0.05$ and $h = 1, 1.5$. For these experiments, we set $n = 2000, m = 10^4$ and $\Delta = 0.8$. We use the modified smoothing procedure Smooth-HD (see appendix) for high-dimensional outputs with a small batch size of 150 to accommodate the samples in memory. It takes about three minutes to smooth and certify each input on a single NVIDIA GeForce RTX 2080 Ti GPU in an internal cluster. Due to the higher factor of ten in the certified output radius in this case compared to our other experiments where the factor is three, the certified output radius increases faster with the input radius ϵ_1, but the smoothing error remains low showing that, in practice, the method does not significantly degrade the performance of the base model. Figure 4.7 shows that, visually, the smoothed output is not very different from the output of the base model. The input radii we certify for are lower in this case than our other experiments due to the low dimensionality (only 128 dimensions) of the input (latent) space as compared to the input (image) spaces in our other experiments.

4.5.3 Total Variation Distance

The total variation norm of a vector x is defined as the sum of the magnitude of the difference between pairs of coordinates defined by a *neighborhood* set N. For a 1-dimensional array x with k elements, one can define the neighborhood as the set of

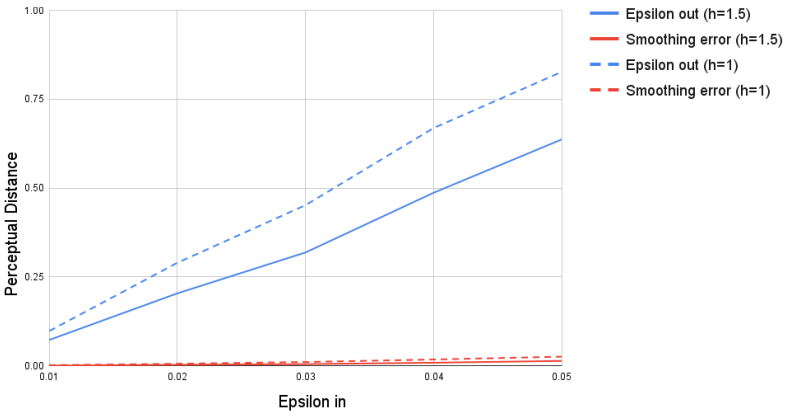


Figure 4.6: Certifying perceptual distance.

Figure 4.7: Model Output vs Smoothed Output.

Figure 4.8: Generative model for ImageNet: Part (a) plots the certified output radius ϵ_2 and the smoothing error for $h = 1$ and 1.5. Part (b) compares the output of the base model to that of the smoothed model.

consecutive elements.

$$TV(x) = \sum_{(i,j)\in N} |x_i - x_j|, \quad TV_{1D}(x) = \sum_{i=1}^{k-1} |x_i - x_{i+1}|.$$

Similarly, for a grayscale image represented by a $h \times w$ 2-dimensional array x, the neighborhood can be defined as the next element (pixel) in the row/column. In case of an RGB image, the difference between the neighboring pixels is a vector, whose magnitude

can be computed using an ℓ_p-norm. For, our experiments we use the ℓ_1-norm.

$$TV_{RGB}(x) = \sum_{i=1}^{h-1}\sum_{j=1}^{w-1} \|x_{i,j} - x_{i+1,j}\|_1 + \|x_{i,j} - x_{i,j+1}\|_1$$

The total variation distance between two images I_1 and I_2 can be defined as the total variation norm of the difference $I_1 - I_2$, i.e., $TVD(I_1, I_2) = TV(I_1 - I_2)$. The above distance defines a pseudometric over the space of images as it satisfies the symmetry property and the triangle inequality, but may violate the identity of indiscernibles as an image obtained by adding the same value to all the pixel intensities has a distance of zero from the original image. However, as noted in section 4.4, our certificates hold even for this setting.

We certify total variation distance for the problems of dimensionality reduction and image reconstruction on MNIST [133] and CIFAR-10 [134]. The base-model for dimensionality reduction is an autoencoder that uses convolutional layers in its encoder module to map an image down to a small number of latent variables. The decoder applies a set of de-convolutional operations to reconstruct the same image. We insert batch-norm layers in between these operations to improve performance. For image reconstruction, the goal is to recover an image from small number of measurements of the original image. We apply a transformation defined by Gaussian matrix A on each image to obtain the measurements. The base model tries to reconstruct the original image from the measurements. The attacker, in this case, is assumed to add a perturbation in the measurement space instead of the image space (as in dimensionality reduction). The model first reverts the measurement vector to a vector in the image space by simply

applying the pseudo-inverse of A and then passes it through a similar autoencoder model as for dimensionality reduction. We present results for $\epsilon_1 = 0.2, 0.4, \ldots, 1.0$ and $h = 2, 1.5$ and use 256 latent dimensions and measurements for these experiments in figure 4.13. To put these plots in perspective, the maximum TVD between two CIFAR-10 images could be $6 \times 31 \times 31 = 5766$ and between MNIST images could be $2 \times 27 \times 27 = 1458$ (pixel values between 0 and 1).

4.6 Conclusion

Provable adversarial robustness can be extended beyond classification tasks to problems with structured outputs. We design a smoothing-based procedure that can make a model of this kind provably robust against norm bounded adversarial perturbations of the input. In our experiments, we demonstrate that this method can generate meaningful certificates under a wide variety of distance metrics in the output space without significantly compromising the quality of the base model. We also note that the metric requirements on the distance measure can be partially relaxed in exchange for weaker certificates.

We focus on ℓ_2-norm bounded adversaries and the Gaussian smoothing distribution. An important direction for future investigation could be whether this method can be generalised beyond ℓ_p-adversaries to more natural threat models, e.g., adversaries bounded by total variation distance, perceptual distance, cosine distance, etc. Center smoothing does not critically rely on the shape of the smoothing distribution or the threat model. Thus, improvements in these directions could potentially be coupled with our method to further broaden the scope of provable robustness in machine learning.

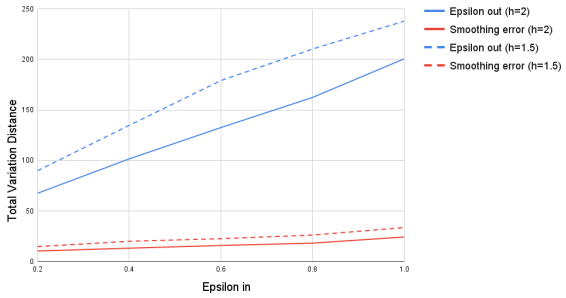


Figure 4.9: Dimensionality Reduction on MNIST

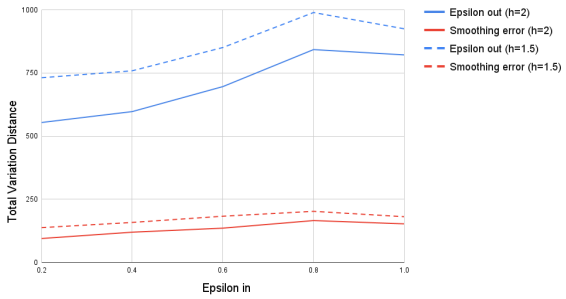


Figure 4.10: Dimensionality Reduction on CIFAR-10

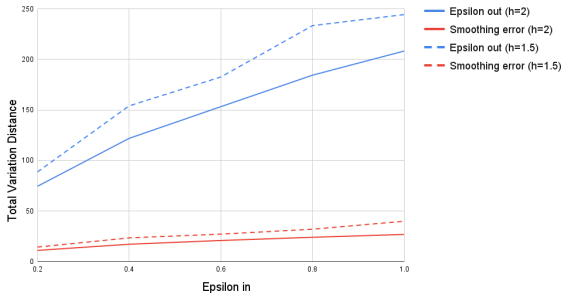


Figure 4.11: Image Reconstruction on MNIST

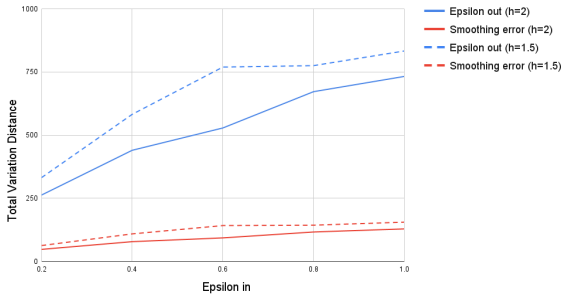


Figure 4.12: Image Reconstruction on CIFAR-10

Figure 4.13: Certifying Total Variation Distance

4.7 Appendices

A Proof of Theorem 4

Let $z' = \hat{f}(x')$. Then, by definition of $\hat{f}$,

$$\mathbb{P}\left[f(X') \in \mathcal{B}(z', \hat{r}(x', \Delta))\right] \geq \frac{1}{2} - \Delta, \tag{4.6}$$

where $X' \sim x' + \mathcal{P}$ and

$$\hat{r}(x', \Delta) = \min_{z''} r \text{ s.t. } \mathbb{P}[f(X') \in \mathcal{B}(z'', r)] \geq \frac{1}{2} + \Delta.$$

And, by definition of $\hat{R}$,

$$\mathbb{P}[f(X') \in \mathcal{B}(\hat{f}(x), \hat{R})] > \frac{1}{2} + \Delta. \tag{4.7}$$

Therefore, from (4.6) and (4.7), $\mathcal{B}(z', \hat{r}(x', \Delta))$ and $\mathcal{B}(\hat{f}(x), \hat{R})$ must have a non-empty intersection. Let, y be a point in that intersection. Then,

$$\begin{aligned} d(\hat{f}(x), \hat{f}(x')) &\leq d(\hat{f}(x), y) + d(y, z') \\ &\leq \hat{r}(x', \Delta) + \hat{R}. \end{aligned}$$

Since, by definition, $\hat{r}(x', \Delta)$ is the radius of the smallest ball with $1/2 + \Delta$ probability mass of $f(x' + \mathcal{P})$ over all possible centers in $\mathbb{R}^k$ and $\hat{R}$ is the radius of the smallest such

ball centered at $\hat{f}(x)$, we must have $\hat{r}(x', \Delta) \leq \hat{R}$. Therefore,

$$d(\hat{f}(x), \hat{f}(x')) \leq 2\hat{R}.$$

B Proof of Lemma 9

Consider the smallest ball $\mathcal{B}(z', \hat{r}(x, \Delta_1))$ that encloses at least $1/2+\Delta_1$ probability mass of $f(x+\mathcal{P})$. By Hoeffding's inequality, with at least $1-e^{-2n\Delta_1^2}$ probability, at least half the points in Z must be in this ball. Since, r is the radius of the minimum enclosing ball that contains at least half of the points in Z, we have $r \leq \hat{r}(x, \Delta_1)$.

C Proof of Theorem 5

β-MEB$(Z, 1/2)$ computes a β-approximation of the minimum enclosing ball that contains at least half of the points of Z. Therefore, by lemma 9, with probability at least $1-e^{-2n\Delta_1^2}$,

$$\beta\text{-MEB}(Z, 1/2) \leq \beta\hat{r}(x, \Delta_1) \leq \beta\hat{r}(x, \Delta),$$

since $\Delta \geq \Delta_1$. Thus, the procedure to compute $\hat{f}$, if succeeds, will output a point $z \in \mathbb{R}^k$ which, with probability at least $1-2e^{-2n\Delta_1^2}$, will satisfy,

$$\mathbb{P}\left[f(X) \in \mathcal{B}(z, \beta\hat{r}(x, \Delta))\right] \geq \frac{1}{2} - \Delta.$$

Now, using the definition of $\hat{R}$ and following the same reasoning as theorem 4, we can say that,

$$\begin{aligned} d(\hat{f}(x), \hat{f}(x')) &\leq \beta \hat{r}(x', \Delta) + \hat{R} \\ &\leq (1+\beta)\hat{R}. \end{aligned}$$

D Proof of Lemma 10

Given $z = \hat{f}(x)$, define a random variable $Q = d(z, f(X))$, where is $X \sim x + \mathcal{P}$. For m i.i.d. samples of X, the values of Q are independently and identically distributed. Let $F(r)$ denote the true cumulative distribution function of Q and define the empirical cdf $F_m(r)$ to be the fraction of the m samples of Q that are less than or equal to r, i.e.,

$$F_m(r) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\{Q_i \leq r\}}$$

Using the Dvoretzky–Kiefer–Wolfowitz inequality, we have,

$$\mathbb{P}\left[\sup_{r \in \mathbb{R}} \left(F_m(r) - F(r)\right) > \epsilon\right] \leq e^{-2m\epsilon^2}$$

for $\epsilon \geq \sqrt{\frac{1}{2m} \ln 2}$. Setting, $e^{-2m\epsilon^2} = \alpha_2$ for some $\alpha_2 \leq 1/2$, we have,

$$\sup_{r \in \mathbb{R}} \left(F_m(r) - F(r)\right) < \sqrt{\frac{\ln\left(1/\alpha_2\right)}{2m}}$$

with probability at least $1 - \alpha_2$. Set $r = \tilde{R}_q$, the qth quantile of of the m samples. Then,

$$F(\tilde{R}_q) > F_m(\tilde{R}_q) - \sqrt{\frac{\ln(1/\alpha_2)}{2m}}$$
$$\text{or,} \quad \mathbb{P}\left[Q \leq \tilde{R}_q\right] > q - \sqrt{\frac{\ln(1/\alpha_2)}{2m}} = p.$$

With probability $1 - \alpha_2$,

$$\mathbb{P}\left[f(X) \in \mathcal{B}(\hat{f}(x), \tilde{R}_q)\right] > p.$$

E High-dimensional Outputs

For functions with high-dimensional outputs, like high-resolution images, it might be difficult to compute the minimum enclosing ball (MEB) for a large number of points. The smoothing procedure needs us to store all the $n \sim 10^3 - 10^4$ sampled points until the MEB computation is complete, requiring $O(nk')$ space, where k' is the dimensionality of the output space. It does not allow us to sample the n points in batches as is possible for the certification step. Also, computing the MEB by considering the pair-wise distances between all the sampled points is time-consuming and requires $O(n^2)$ pair-wise distance computations. To bring down the space and time requirements, we design another version (Smooth-HD, algorithm 7) of the smoothing procedure where we compute the MEB by first sampling a small number $n_0 \sim 30$ of candidate centers and then returning one of these candidate centers that has the smallest median distance to a separate sample of n ($\gg n_0$) points. We sample the n points in batches and compute the distance $d(c_i, z_j)$ for each pair of candidate center c_i and point z_j in a batch. The rest of the procedure remains the same

Algorithm 7: Smooth-HD

Input: $x \in \mathbb{R}^k, \sigma, \Delta, \alpha_1$.
Output: $z \in M$.
Set $C = \{c_i\}_{i=1}^{n_0}$ s.t. $c_i \sim f(x + \mathcal{N}(0, \sigma^2 I))$.
Set $\Delta_1 = \sqrt{\ln(2/\alpha_1)/2n}$.
Sample $Z = \{z_j\}_{j=1}^{n}$ s.t. $z_j \sim f(x + \mathcal{N}(0, \sigma^2 I))$ in batches.
For each batch, compute pair-wise distances $d(c_i, z_j)$ for $c_i \in C$ and z_j in the batch.
Compute the center $c \in C$ with the minimum median distance to the points in Z.
Re-sample Z in batches.
Compute p_{Δ_1}.
Set $\Delta_2 = 1/2 - p_{\Delta_1}$.
If $\Delta < \max(\Delta_1, \Delta_2)$, discard c and abstain.

as algorithm 5. It only requires us to store batch-size number of output points and the n_0 candidate centers at any given time, significantly reducing the space complexity. Also, this procedure only requires $O(n_0 n)$ pair-wise distance computations. The key idea here is that, with very high probability ($> 1 - 10^{-9}$), at least one of the n_0 candidate centers will lie in the smallest ball that encloses at least $1/2 + \Delta_1$ probability mass of $f(x + \mathcal{P})$. Also, with high probability, at least half of the n samples will lie in this ball too. Thus, the median distance of this candidate center to the n samples is at most $2\gamma\hat{r}(x, \Delta_1)$, after accounting for the factor of γ in the relaxed version of the triangle inequality as discussed in section 4.4. Ignoring the probability that none of the n_0 points lie inside the ball, we can derive the following version of theorem 5:

Theorem 6. *With probability at least* $1 - \alpha_1$*,*

$$\forall x' \textit{ s.t. } \|x - x'\|_2 \leq \epsilon_1,\ d(\hat{f}(x), \hat{f}(x')) \leq \gamma(1 + 2\gamma)\hat{R}$$

where $\alpha_1 = 2e^{-2n\Delta_1^2}$.

F Baseline for ℓ_2-Metric

In this section, we compare the certificates from center smoothing against a bound derived in [126] for functions like f smoothed by taking the expectation of f under a Gaussian noise. This bound only applies when the output metric is ℓ_2. For a vector-valued function f, the change in the function defined as $\mathbb{E}_\delta[f(x+\delta)]$ where $\delta \sim \mathcal{N}(0, \sigma^2 I)$, under an ℓ_2-perturbation of the input of size ϵ_1, can be bounded by $(\max \|f\|_2 + \min \|f\|_2) \operatorname{erf} \epsilon_1 / 2\sqrt{2}\sigma$. We apply our center smoothing procedure on the autoencoder and image reconstruction models used in section 4.5.3 with ℓ_2 as the output metric and compare its certificates to the above bound. Since the minimum ℓ_2-norm of the output of these models can be zero and we keep $h = \epsilon_1/\sigma = 2$ for these experiments, the change in the output of $\mathbb{E}_\delta[f(x+\delta)]$ can be bounded by $\max \|f\|_2 \operatorname{erf} 1/\sqrt{2} \leq 0.68\sqrt{d}$, where d is the number of dimensions of the output space. For 28×28 gray-scale MNIST images and 32×32 RGB CIFAR-10 images, the corresponding bounds are 19.04 and 37.69 respectively. Figure 4.18 shows that the certificates obtained for center smoothing remain below the baseline for all the values of ϵ_1 used. Thus, by observing the neighborhood of an input point, center smoothing can yield better certificates for individual points in the input space than the baseline bound which is a global guarantee.

G Angular Distance

A common measure for similarity of two vectors A and B is the cosine similarity between them, defined as below:

$$\cos(A, B) = \frac{A \cdot B}{\|A\|_2 \|B\|_2} = \frac{\sum_i A_i B_i}{\sqrt{\sum_j A_j^2}\sqrt{\sum_k B_k^2}}.$$

In order to convert it into a distance, we can compute the angle between the two vectors by taking the cosine inverse of the above similarity measure, which is known as angular distance:

$$AD(A, B) = \cos^{-1}(\cos(A, B))/\pi.$$

Angular distance always remains between 0 and 1, and similar to the total variation distance, angular distance also defines a pseudometric on the output space. We repeat the same experiments with the same models and hyper-parameter settings as for total variation distance (figure 4.23). The results are similar in trend in all the experiments conducted, showing that center smoothing can be reliably applied to a vast range of output metrics to obtain similar robustness guarantees.

H Effect of Training with Noise

A common practice in the randomized smoothing literature is to train the base model with noise added to the training examples [36]. This helps the model to learn to ignore the smoothing noise and leads to better robustness certificates for classification tasks. For the total variation certificates in section 4.5.3, we train the autoencoders and

the reconstruction models using a Gaussian noise with the same variance as the one used for prediction and certification. In this section, we perform an ablation experiment to study the effect of the training noise in the certified output radius of the base model (figure 4.28). We observe that both the smoothing error and the certified output radius deteriorate in the absence of training noise. However, models trained without noise also produce non-trivial certificates. This shows that both center smoothing and training with noise contribute towards the robustness and performance of the smoothed models.

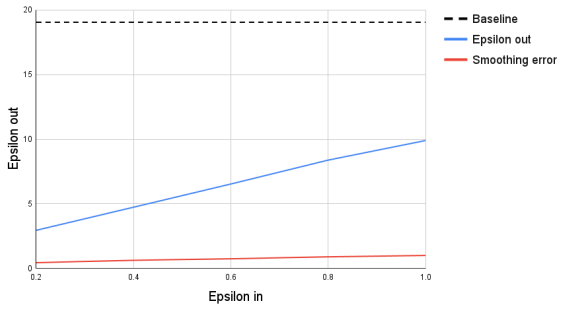


Figure 4.14: Dimensionality Reduction on MNIST

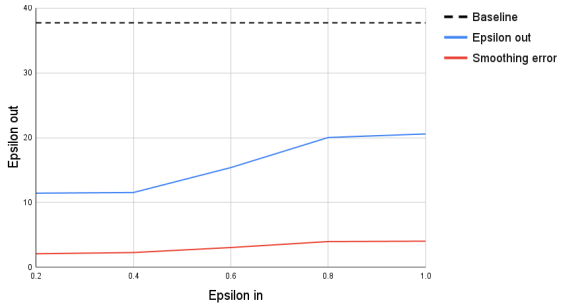


Figure 4.15: Dimensionality Reduction on CIFAR-10

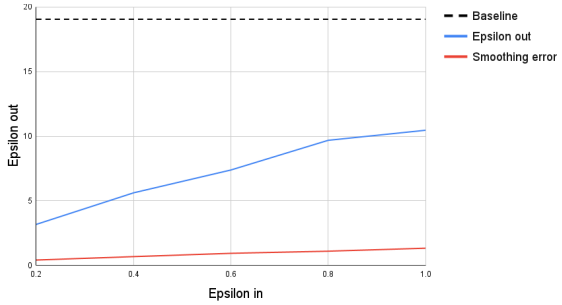


Figure 4.16: Image Reconstruction on MNIST

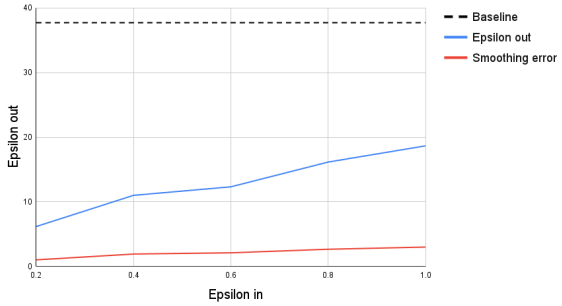


Figure 4.17: Image Reconstruction on CIFAR-10

Figure 4.18: Comparison with baseline ($h = 2$).

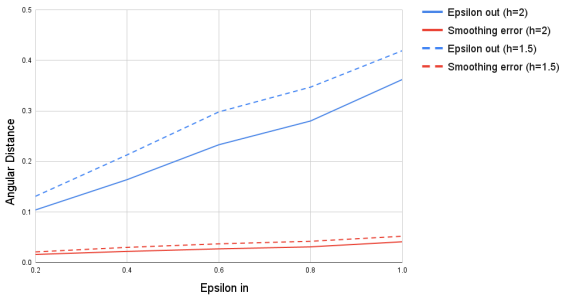


Figure 4.19: Dimensionality Reduction on MNIST

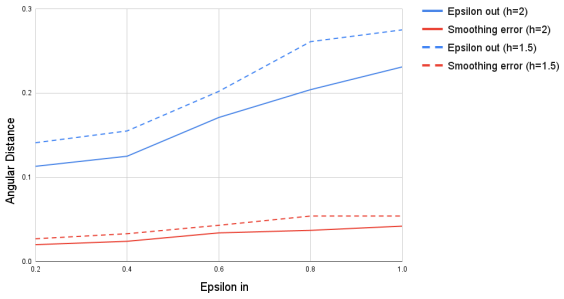


Figure 4.20: Dimensionality Reduction on CIFAR-10

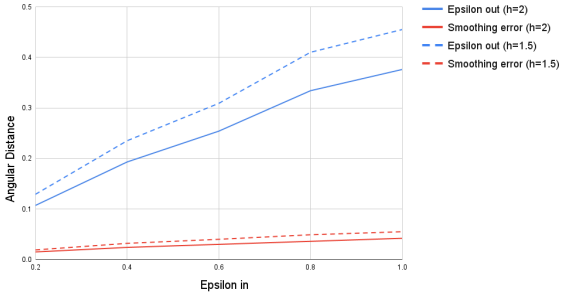


Figure 4.21: Image Reconstruction on MNIST

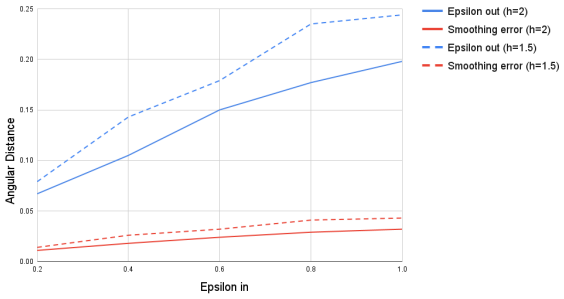


Figure 4.22: Image Reconstruction on CIFAR-10

Figure 4.23: Certifying Angular Distance

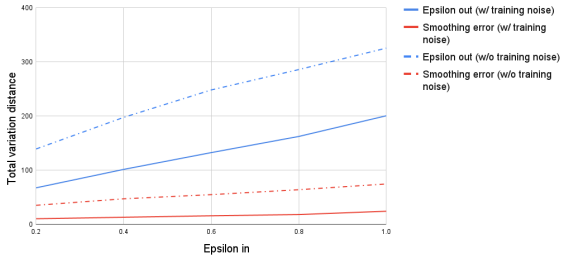


Figure 4.24: Dimensionality Reduction on MNIST

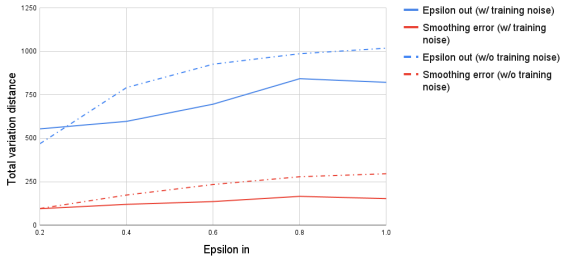


Figure 4.25: Dimensionality Reduction on CIFAR-10

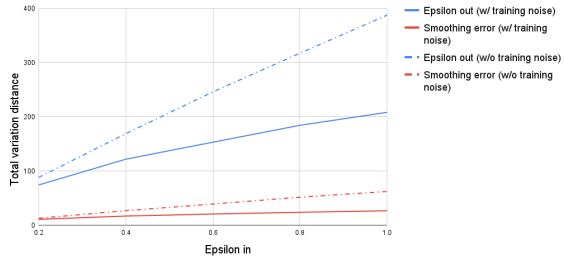


Figure 4.26: Image Reconstruction on MNIST

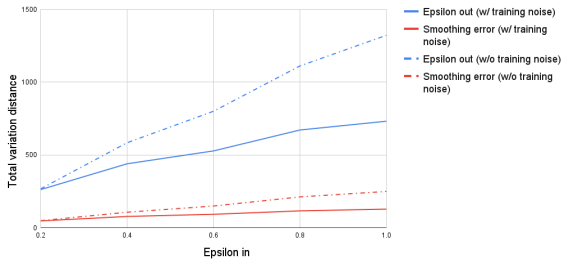


Figure 4.27: Image Reconstruction on CIFAR-10

Figure 4.28: Impact of training noise on the performance of the robust model and its certificates.

Chapter 5: Limitations of Randomized Smoothing

5.1 Introduction

Deep neural networks, especially in image classification tasks, have been shown to be vulnerable to adversarial perturbations of the input that are unnoticeable to a human observer but can alter the prediction of the model [1]. These examples are generated by optimizing a loss function for a trained network over the input features within a small neighborhood of an example input. Gradient based methods such as FGSM [3] and projected gradient descent [2] have been shown to be very effective for this purpose. In the last couple of years, several heuristic methods have been proposed to detect and/or defend against attacks from specific types of adversaries [18, 19, 20, 21, 22, 23]. Such defenses, however, have been shown to break down against more powerful attacks [4, 24, 25, 26]. For certain types of problems, adversarial examples might even be unavoidable [135].

This necessitates developing classifiers with robustness guarantees. Several convex relaxation-based techniques have been proposed to design *certifiably robust* classifiers [27, 28, 29, 30, 136] whose predictions are guaranteed to remain constant within a certified neighborhood around the input point, thereby eliminating the presence of any adversarial example in that region. However, the ever-increasing complexity of deep neural networks has made it difficult to scale these methods meaningfully to high-dimensional datasets

like ImageNet.

To deal with the scalability issue in certifiable robustness, a line of work has been introduced based on *randomized robustness* [36, 37, 38, 39, 68, 121, 124, 137, 138, 139] wherein an arbitrary base classifier is made more robust by averaging its prediction over random perturbations of the input point within its neighborhood. Cohen et al. (2019) proved the first tight robustness guarantee for Gaussian smoothing for an ℓ_2-norm bounded adversary.

In this work, however, we show that extending the smoothing technique to defend against higher-norm attacks, especially in the high-dimensional regime, can be challenging. In particular, for a general class of i.i.d. smoothing distributions, we show that, for $p > 2$, the largest ℓ_p-radius that can be certified (denoted by r_p^*) decreases with the number of dimensions d as $O(1/d^{\frac{1}{2}-\frac{1}{p}})$. Note that the special case of $p = 2$ does not suffer from such dependency on d. This makes smoothing-based robustness bounds weak against ℓ_p adversarial attacks for large p, especially, for ℓ_∞ because as $p \to \infty$ the dependence on d becomes $O(1/\sqrt{d})$. Moreover, we show that the dependence of the robustness certificate on d using a general i.i.d. smoothing distribution is similar to that of the standard Gaussian smoothing, even for $p > 2$. This implies that Gaussian smoothing essentially provides the best possible robustness certificate result in terms of the dependence on d even for $p > 2$.

To be more precise, suppose we smooth a classifier by randomly sampling points surrounding an image x, and observing the labels assigned to these points. Let $p_1(x)$ and $p_2(x)$ be the probabilities of the first and second most probable labels under the smoothing distribution. We prove the following bounds on the robustness certificate:

1. When points are sampled by adding i.i.d. noise to each dimension in x with σ^2 variance and continuous support, we prove the certified ℓ_p radius bound

$$r_p^* \leq \frac{\sigma}{2\sqrt{2}d^{\frac{1}{2}-\frac{1}{p}}}\left(\frac{1}{\sqrt{1-p_1(x)}}+\frac{1}{\sqrt{p_2(x)}}\right),$$

whenever $p_1(x) \geq 1/2$. See Theorem 7.

2. When smoothing with a generalized Gaussian distribution with variance σ^2 (which includes Laplacian, Gaussian, and uniform distributions), we prove that

$$r_p^* \leq \frac{2\sigma}{d^{\frac{1}{2}-\frac{1}{p}}}\left(\sqrt{\log\frac{1}{1-p_1(x)}}+\sqrt{\log\frac{1}{p_2(x)}}\right),$$

when $e^{-d/4} < p_2(x) \leq p_1(x) < 1 - e^{-d/4}$. When d is large, these bounds do not impact the range of values that $p_1(x)$ and $p_2(x)$ can take in a significant way. See Theorem 8.

3. We also study smoothing techniques where the distribution is uniform over a region around the input point. When smoothed over an ℓ_∞ ball of radius b, i.e. uniform i.i.d between $-b$ and b in each dimension, we show that

$$r_p^* < \frac{2b}{d^{1-\frac{1}{p}}} = 2\sqrt{3}\sigma/d^{1-\frac{1}{p}},$$

where $\sigma^2 = b^2/3$ is the variance in each dimension. See Theorem 9. Note that this bound is independent of $p_1(x)$ and $p_2(x)$.

4. For smoothing uniformly over an ℓ_1 ball of the same radius b, we achieve an even

stronger bound:

$$r_p^* < \frac{2b}{d}$$

See Theorem 10 for details. Along with being independent of $p_1(x)$ and $p_2(x)$, it is also independent of p. Thus, it holds for any p-norm bounded adversary. Note that, unlike the other smoothing distributions we have considered, the uniform ℓ_1 smoothing is not i.i.d. in every dimension.

These bounds hold for any $p > 0$, but are too weak to offer meaningful insights when $p < 2$ in the first two cases and for $p < 1$ in the third one. Moreover, it is straightforward to show that, for $p \geq 2$, the following ℓ_p-radius can be certified using Cohen et al.'s (2019) Gaussian smoothing:

$$r_p = \frac{\sigma}{2d^{\frac{1}{2}-\frac{1}{p}}} \left(\Phi^{-1}\left(p_1(x)\right) - \Phi^{-1}\left(p_2(x)\right) \right), \tag{5.1}$$

which has the same dependence on d as the upper bound obtained using i.i.d. smoothing. This radius is asymptotically only a constant factor away from the upper bound for the generalized Gaussian distribution, showing that this family of distributions fails to outperform standard Gaussian smoothing in high dimensions. To the best of our knowledge, these bounds form the first results on the limitations of randomized smoothing in the high dimensional regime that cover an extensive range of natural and commonly used smoothing distributions.[1] We provide empirical evidence to support our claims on the CIFAR-10 dataset.

[1] We have later come to know about a concurrent work which also illustrates the difficulty of extending randomized smoothing to defend against ℓ_∞ -attacks for high-dimensional data [140].

5.2 Preliminaries and Notation

Let h be a classifier that maps inputs from $\mathbb{R}^d$ to classes in $\mathcal{C}$. Let $\mathcal{P}$ be a (smoothing) probability distribution in $\mathbb{R}^d$. We define a *smoothed* classifier $\bar{h}$ as below:

$$\bar{h}(x) \triangleq \arg\max_{c \in \mathcal{C}} \mathop{\mathbb{P}}_{\Delta \sim \mathcal{P}}(h(x + \Delta) = c).$$

We refer to the process of smoothing using distribution $\mathcal{P}$ as $\mathcal{P}$-smoothing. Let $p_c(x)$ be the output probability of the base classifier for the class c. That is,

$$p_c(x) := \mathop{\mathbb{P}}_{\Delta \sim \mathcal{P}}(h(x + \Delta) = c).$$

Without loss of generality, we assume that $p_1(x)$ and $p_2(x)$ are the probabilities of the first and second most likely classes, respectively.

For $p > 0$, we say a smoothing distribution $\mathcal{P}$ achieves a *certified ℓ_p-norm radius* of r_p if, for a base classifier h and an input x,

$$\bar{h}(x + \delta) = \bar{h}(x), \quad \forall \delta \in \mathbb{R}^d, \|\delta\|_p \leq r_p.$$

For instance, as derived in [36], the Gaussian smoothing distribution $\mathcal{N}(0, \sigma^2 I)$ achieves a certified 2-norm radius of $\frac{\sigma}{2}(\Phi^{-1}(p_1(x)) - \Phi^{-1}(p_2(x)))$ where Φ^{-1} is the inverse of the standard Gaussian CDF.

For $p_1, p_2 \in (0, 1)$, such that, $p_1 \geq p_2$, let r_p^* denote the largest r_p that can be certified using $\mathcal{P}$-smoothing for all classifiers satisfying $p_1(x) = p_1$ and $p_2(x) = p_2$. If

we can show a classifier h in this class and two points $x, x' \in \mathbb{R}^d$, such that, $\bar{h}(x) \neq \bar{h}(x')$, then $r_p^* \leq \|x' - x\|_p$. We use this fact to show upper bounds on the largest p-norm radius that can be certified using a given class of distributions.

5.3 General i.i.d. Smoothing

We set the $\mathcal{P}$ to be a smoothing distribution $\mathcal{I}$ where each coordinate of Δ is sampled independently and identically from a symmetric distribution with zero mean, σ^2 variance with a continuous support. We prove the following theorem:

Theorem 7. *For distribution $\mathcal{I}$ and for $p_1, p_2 \in (0, 1)$, such that, $p_1 \geq 1/2$ and $p_1 + p_2 \leq 1$, the largest ℓ_p-radius r_p^* that can be certified for all classifiers satisfying $p_1(x) = p_1$ and $p_2(x) = p_2$ under $\mathcal{I}$-smoothing at input point x is bounded as:*

$$r_p^* \leq \frac{\sigma}{2\sqrt{2} d^{\frac{1}{2} - \frac{1}{p}}} \left(\frac{1}{\sqrt{1 - p_1(x)}} + \frac{1}{\sqrt{p_2(x)}} \right). \tag{5.2}$$

Proof. Let Z_i be the random variable modelling the i^{th} coordinate of Δ. Define a random variable $S = \sum_{i=1}^{d} Z_i$. It is straightforward to show that this random variable is distributed symmetrically with zero mean, $d\sigma^2$ variance and a continuous support. The key intuition behind this proof is that the random variable S, which is the sum of d identical and independent random variables, will tend towards a Gaussian distribution for large values of d, making the distribution $\mathcal{I}$ suffer from some of the same limitations as the Gaussian distribution.

To simplify our analysis, we move our frame of reference so that x is at the origin.

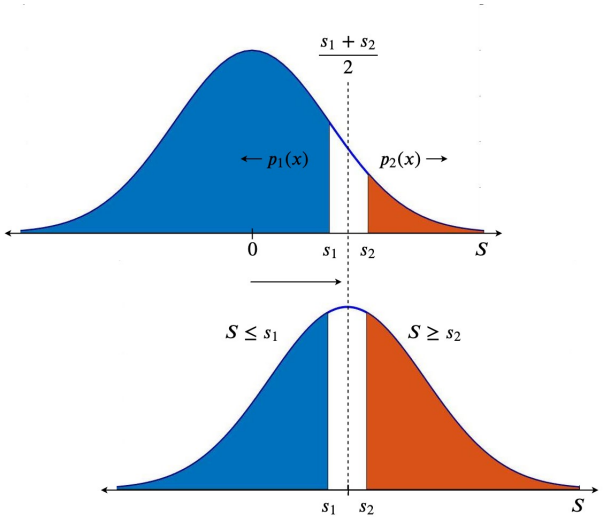


Figure 5.1: As the distribution of S moves from the origin to $\frac{s_1+s_2}{2}$ the probability for class one decreases and that of class two increases. They become equal at $\frac{s_1+s_2}{2}$ beyond which class two becomes more likely.

Therefore, $r_p^* \le \|x'\|_p$. Consider a classifier g that maps points in $\{w \in \mathbb{R}^d \mid \sum_{i=1}^d w_i \le s_1\}$ to class one and those in $\{w \in \mathbb{R}^d \mid \sum_{i=1}^d w_i \ge s_2\}$ to class two. We pick $s_1, s_2 \in \mathbb{R}^+$ such that, $\mathbb{P}(S \le s_1) = p_1(x)$ (this requires $p_1(x) \ge 1/2$) and $\mathbb{P}(S \ge s_2) = p_2(x)$. Let x' be the point with every coordinate equal to ϵ and so, $\sum_{i=1}^d x'_i = \epsilon d$. Since S is symmetric and has a continuous support, $\bar{g}(x') = \bar{g}(x)$ only if $\sum_{i=1}^d x'_i \le \frac{s_1+s_2}{2}$, which implies $\epsilon \le \frac{s_1+s_2}{2d}$. Therefore,

$$r_p^* \le \|x'\|_p = \epsilon d^{1/p} \le \frac{s_1+s_2}{2d^{1-\frac{1}{p}}}. \tag{5.3}$$

Figure 5.1 illustrates how the probabilities of the top two classes change as we move from x to x'.

Applying Chebyshev's inequality on S, we have:

$$P(S \ge s) = \frac{P(|S| \ge s)}{2} \le \frac{d\sigma^2}{2s^2}$$

The value of s for which $\frac{d\sigma^2}{2s^2} = p_2(x)$ must be an upper-bound on s_2.

$$s_2 \leq \frac{\sqrt{d}\sigma}{\sqrt{2p_2(x)}}$$

Similarly, since $\mathbb{P}(S \geq s_1) = 1 - p_1(x)$,

$$s_1 \leq \frac{\sqrt{d}\sigma}{\sqrt{2(1-p_1(x))}}$$

Substituting the above bounds for s_1 and s_2 in (5.3), proves Theorem (7):

$$r_p^* \leq \frac{\sigma}{2\sqrt{2}d^{\frac{1}{2}-\frac{1}{p}}} \left(\frac{1}{\sqrt{1-p_1(x)}} + \frac{1}{\sqrt{p_2(x)}} \right).$$

□

5.4 Generalized Gaussian Smoothing

We now restrict ourselves to the class of generalized Gaussian distributions that subsumes some commonly used and natural smoothing distributions such as Gaussian, Laplacian and uniform distributions. Using a similar approach as in the previous section, we obtain tighter upper bounds on r_p^* by restricting the smoothing distribution to generalized Gaussian. In this class of distributions, each coordinate is sampled independently from the following distribution:

$$p(z) = \frac{1}{C} e^{-(|z|/b)^q}$$

where $z \in \mathbb{R}$, $b > 0$ is the *scale parameter*, $q > 0$ is the *shape parameter* and C is the normalizing constant

$$\begin{aligned} C &= \int_{-\infty}^{\infty} e^{-(|z|/b)^q} dz \\ &= 2\int_{0}^{\infty} e^{-z^q/b^q} dz = \frac{2b\Gamma(1/q)}{q}, \end{aligned} \tag{5.4}$$

where $\Gamma(.)$ is the gamma function. The mean of this distribution is at zero and the variance σ^2 can be calculated as

$$\begin{aligned} \sigma^2 &= \frac{1}{C}\int_{-\infty}^{\infty} z^2 e^{-(|z|/b)^q} dz \\ &= \frac{2}{C}\int_{0}^{\infty} z^2 e^{-z^q/b^q} dz = \frac{2b^3\Gamma(3/q)}{Cq}. \end{aligned}$$

Substituting C from (5.4) leads to

$$\sigma^2 = \frac{b^2\Gamma(3/q)}{\Gamma(1/q)}.$$

Note that the class of generalised Gaussian distributions is a subset of the class of i.i.d. smoothing distributions considered in the previous section. The joint probability distribution over all the d dimensions can be expressed as:

$$p(z_1, z_2, \ldots, z_d) = \frac{1}{C^d} e^{-\sum_{i=1}^{d}(|z_i|/b)^q},$$

which for $q = 1, 2$ represents Laplace and Gaussian distributions, respectively. As $q \to \infty$, this distribution approximates the uniform distribution over $[-b, b]^d$. For a finite q, the level sets of the above p.d.f. define sets with constant ℓ_q-norm. Let $\mathcal{G}$ be a generalised Gaussian distribution with $q \geq 1$. The following theorem holds:

Theorem 8. *For distribution $\mathcal{G}$ and for $e^{-d/4} < p_2 \leq p_1 < 1 - e^{-d/4}$ and $p_1 + p_2 \leq 1$, the largest ℓ_p-radius r_p^* that can be certified for all classifiers satisfying $p_1(x) = p_1$ and $p_2(x) = p_2$ under $\mathcal{G}$-smoothing at input point x, is bounded as:*

$$r_p^* \leq \frac{2\sigma}{d^{\frac{1}{2}-\frac{1}{p}}} \left(\sqrt{\log(1/(1-p_1(x)))} + \sqrt{\log(1/p_2(x))} \right) \tag{5.5}$$

We provide a brief proof sketch for this theorem here. As before, define random variables Z_i and S, and assume x to be at the origin. Since the above distribution satisfies all the assumptions made in the previous section, we can directly conclude that the bound in (5.3) holds:

$$r_p^* \leq \frac{s_1 + s_2}{2d^{1-\frac{1}{p}}}$$

From here, we strengthen our analysis by replacing Chebyshev's inequality with Chernoff bound.

$$P(S \geq s) \leq \frac{E[e^{tS}]}{e^{ts}}$$

for any $t > 0$. Since S is a sum of independent random variables $Z_1, Z_2, \ldots, Z_d$ sampled from identical distributions,

$$P(S \geq s) \leq e^{-ts} \prod_{i=1}^{d} E[e^{tZ_i}] \leq e^{-ts} E[e^{tZ}]^d$$

where Z is sampled from $p(z)$.

Lemma 11. *For some constant* $c < 1.85$*,*

$$E[e^{tZ}] \leq \sum_{m=0}^{\infty} (c^2 t^2 \sigma^2)^m$$

Proof is presented in the appendix.

Setting $t = \frac{1}{\tau\sigma\sqrt{d}}$ for some $\tau > 0$ satisfying $\frac{c^2}{\tau^2 d} < 1$, we have:

$$\begin{aligned} P(S \geq s) &\leq e^{-s/\tau\sigma\sqrt{d}} \left(\sum_{m=0}^{\infty} (c^2/\tau^2 d)^m \right)^d \\ &= \frac{e^{-s/\tau\sigma\sqrt{d}}}{(1 - \frac{c^2}{\tau^2 d})^d} \leq e^{-s/\tau\sigma\sqrt{d}} e^{4/\tau^2} \end{aligned}$$

for $\tau^2 d \geq 16$. The value of s for which this expression is equal to $p_2(x)$ gives us the following upper-bound on s_2:

$$s_2 \leq \sigma\sqrt{d}(\tau \log(1/p_2(x)) + 4/\tau)$$

which for $\tau = 2/\sqrt{\log(1/p_2(x))}$ gives:

$$s_2 \leq 4\sigma\sqrt{d \log(1/p_2(x))}$$

and similarly, repeating the above analysis and setting $\tau = 2/\sqrt{\log(1/(1-p_1(x)))}$, we get:

$$s_1 \leq 4\sigma\sqrt{d \log(1/(1-p_1(x)))}$$

Both the above values for τ satisfy $\tau^2 d \geq 16$ due to the restrictions on p_1 and p_2. Substituting the above bounds for s_1 and s_2 in inequality (5.3), proves Theorem (8):

$$r_p^* \leq \frac{2\sigma}{d^{\frac{1}{2}-\frac{1}{p}}} \left(\sqrt{\log(1/(1-p_1(x)))} + \sqrt{\log(1/p_2(x))} \right)$$

When $p_1(x)$ is close to one and $p_2(x)$ is close to zero, this bound is within a constant factor of the Gaussian certificate in equation (5.1) because $\Phi^{-1}(p)$ can be lower bounded by $\alpha\sqrt{\log(1/(1-p)) + \beta}$ for some constants α and β. Figure (5.2) compares the behaviour of the two upper bounds, the one from i.i.d. smoothing $u_{\mathcal{I}}$ and the one from generalized Gaussian smoothing $u_{\mathcal{G}}$, with respect to the Gaussian certificate r_p obtained in equation (5.1). Assuming the binary classification case, for which $p_2(x) = 1 - p_1(x)$, we plot the ratios

$$\frac{u_{\mathcal{I}}}{r_p} = \frac{1}{\phi^{-1}(p_1(x))\sqrt{2(1-p_1(x))}},$$
$$\frac{u_{\mathcal{G}}}{r_p} = \frac{4\sqrt{\log \frac{1}{1-p_1(x)}}}{\phi^{-1}(p_1(x))}$$

which only depend on $p_1(x)$ and show that the generalized Gaussian bound is much tighter than the i.i.d. bound when $p_1(x)$ is close to one.

5.5 Uniform Smoothing

In this section, we analyse smoothing distributions that are uniform within a finite region around the input point x. We show stronger upper bounds for r_p^* when smoothed

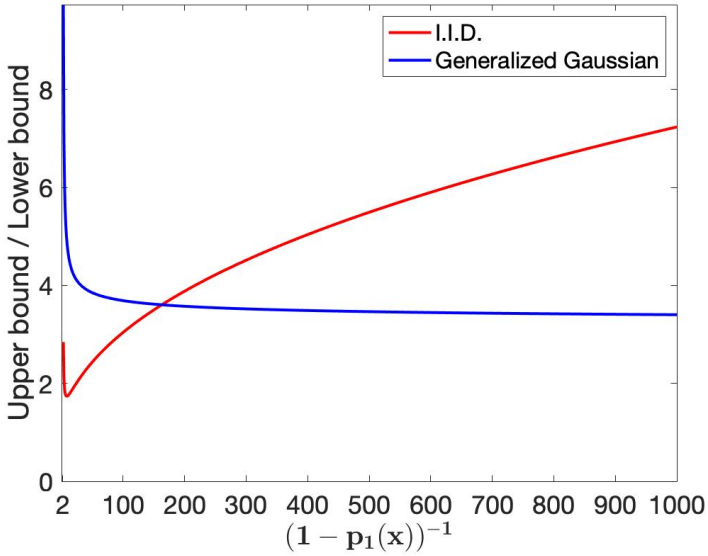


Figure 5.2: Comparison of the upper bounds from i.i.d. smoothing (5.2) and generalized Gaussian smoothing (5.5) w.r.t. the lower bound obtained from Gaussian smoothing (5.1). The x-axis represents $\frac{1}{1-p_1(x)}$ for $\frac{1}{2} \leq p_1(x) \leq 1$ and the y-axis represents the ratio of each upper bound to the Gaussian lower bound. At around $p_1(x) \approx 0.99$, the generalized Gaussian bound becomes tighter than the i.i.d. bound and gets within a constant factor of the Gaussian lower bound as $p_1(x)$ gets larger.

uniformly over ℓ_1 and ℓ_∞-norm balls. We first consider the ℓ_∞ smoothing distribution which is a limiting case for the generalized Gaussian distribution for $q = \infty$. We set $\mathcal{P}$ to be $\mathcal{U}([-b, +b]^d)$ which denotes a uniform distribution over the points in $[-b, +b]^d$.

Theorem 9. *For distribution $\mathcal{U}([-b, +b]^d)$, the largest ℓ_p-radius r_p^* that can be certified for all classifiers, is bounded as*

$$r_p^* < \frac{2b}{d^{1-\frac{1}{p}}} = 2\sqrt{3}\sigma/d^{1-\frac{1}{p}}.$$

where $\sigma^2 = b^2/3$ is the variance in each dimension.

Proof. Assume x is at origin and let x' be a point with every coordinate equal to ϵ. Let V_1 and V_2 denote the sets $[-b, +b]^d$ and $[-b + \epsilon, b + \epsilon]^d$. Consider a classifier g that maps

every point in $V_1 - V_2$ to class one and every point in $V_2 - V_1$ to class two. See figure 5.3. Let ρ denote the probability with which the smoothing distribution for $\bar{g}(x)$ samples from $V_1 - V_2$, which is equal to the probability with which the smoothing distribution for $\bar{g}(x')$ samples from $V_2 - V_1$, or

$$\begin{aligned}\rho &= \frac{(2b)^d - (2b-\epsilon)^d}{(2b)^d} \\ &= \left(1 - \left(1 - \frac{\epsilon}{2b}\right)^d\right).\end{aligned}$$

For $\bar{g}$ to classify x' into class one, we must have:

$$\begin{aligned} p_1(x') &> p_2(x') \\ p_1(x) - \rho &> p_2(x) + \rho \\ \rho &< \frac{p_1(x) - p_2(x)}{2} \\ \left(1 - \left(1 - \frac{\epsilon}{2b}\right)^d\right) &< \frac{1}{2} && p_1(x) - p_2(x) \leq 1 \\ \epsilon &< 2b\left(1 - 2^{-1/d}\right) < 2b/d && \left(1 - 2^{-1/d}\right) < 1/d \end{aligned}$$

Since $\|x'\|_p = \epsilon d^{1/p}$, the optimal radius,

$$r_p^* < 2b/d^{1-\frac{1}{p}} = 2\sqrt{3}\sigma/d^{1-\frac{1}{p}}$$

where σ^2 is the variance of $\mathcal{U}(-b, b)$. □

This shows that for $p > 1$, σ (or b) needs to grow with the number of dimensions d to certify for a meaningfully large p-norm radius. For instance, $p = 2$ and ∞, require σ to be

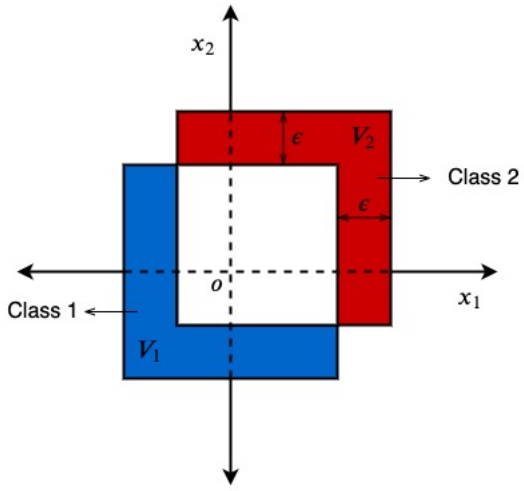


Figure 5.3: 2-D illustration of the ℓ_∞ smoothing case. The ℓ_∞ ball is shifted by ϵ along x_1 and x_2. The points in the blue region ($V_1 - V_2$) are mapped to class one and the points in the red region ($V_2 - V_1$) to class two.

$\Theta(\sqrt{d})$ and $\Theta(d)$ respectively. However, since inputs can be assumed to come from $[0, 1]^d$ (possibly after some scaling and shifting of images), smoothing over distributions with such large variance may significantly lower the performance of the smoothed classifier.

We now consider the uniform ℓ_1 smoothing distribution (denoted by $\mathcal{L}_1(b)$) where points are sampled uniformly from an ℓ_1-norm ball of radius b. Note that the noise in each dimension is no longer independent.

Theorem 10. *For distribution $\mathcal{L}_1(b)$, the largest ℓ_p-radius r_p^* that can be certified for all classifiers, is bounded as*

$$r_p^* < \frac{2b}{d}.$$

The following is a proof sketch of the above theorem. Let x be at the origin and x' be the point $(\epsilon, 0, 0, \ldots, 0)$, that is, ϵ in the first coordinate and zero everywhere else. Similar to before, let V_1 and V_2 be the sets defined by the ℓ_1 balls centered at x and x' respectively.

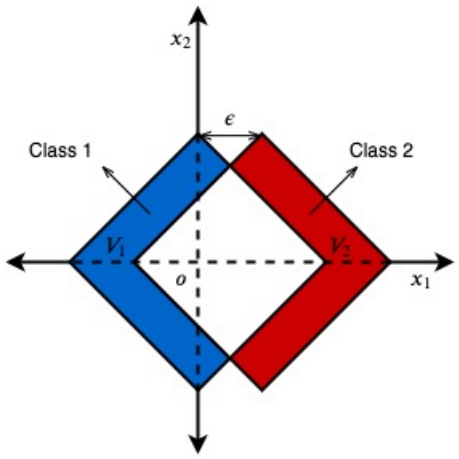


Figure 5.4: 2-D illustration of the ℓ_1 smoothing case. The ℓ_1 ball is shifted by ϵ along x_1. The points in the blue region ($V_1 - V_2$) are mapped to class one and the points in the red region ($V_2 - V_1$) to class two.

Lemma 12. *The set $V_1 \cap V_2$ is a subset of an ℓ_1 ball of radius $b - \frac{\epsilon}{2}$.*

The proof is presented in the appendix.

As before, let g be a classifier that maps every point in $V_1 - V_2$ to class one and every point in $V_2 - V_1$ to class two (figure 5.4). Let ρ denote the probability with which the smoothing distribution for $\bar{g}(x)$ samples from $V_1 - V_2$, which is equal to the probability with which the smoothing distribution for $\bar{g}(x')$ samples from $V_2 - V_1$, or

$$\rho \geq \frac{\frac{2^d}{d!}b^d - \frac{2^d}{d!}(b - \frac{\epsilon}{2})^d}{\frac{2^d}{d!}b^d} = \left(1 - \left(1 - \frac{\epsilon}{2b}\right)^d\right).$$

We us the formula $2^d R^d / d!$ as the volume of a d-dimensional ℓ_1 ball of radius R. The rest of the analysis is same as that for the ℓ_∞ case and since $\|x'\|_p = \epsilon$, we have,

$$r_p^* < \frac{2b}{d},$$

which proves Theorem 10.

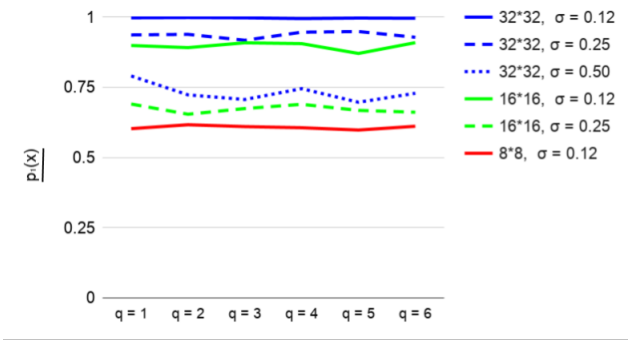


Figure 5.5: $\underline{p_1(x)}$ for CIFAR-10 images with median certified robustness for each classifier using Generalized Gaussian smoothing for different q. For a fixed standard deviation σ, the shape of the distribution, controlled by q, has almost no effect on the likelihood that the base classifier returns the correct class.

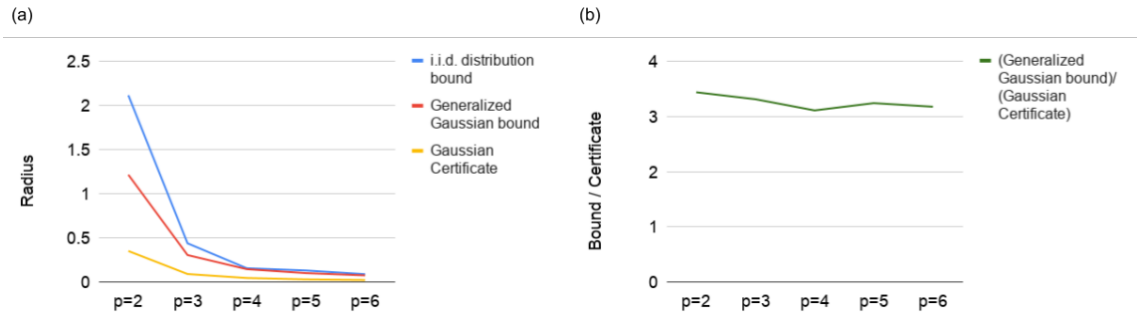


Figure 5.6: Upper bounds for certifying with Generalized Gaussian noise ($\sigma = .12$) on unaltered (32×32) CIFAR-10 images, with $q = p$, compared with certificates using Gaussian noise directly. At this noise level, $\underline{p_1(x)}$ is high enough for the Generalized Gaussian bound to be tighter than the i.i.d. distribution bound. Panel (a) shows the certificates and the bounds directly, while (b) shows the ratio between the tighter Generalized Gaussian bound and the certificate.

5.6 Experiments

In order to understand how our results apply to smoothing in practice, we tested the smoothed classification algorithm proposed by [36], using Generalized Gaussian noise in each dimension, rather than Gaussian noise. We specifically tested on CIFAR-10 (32×32 pixels), as well as scaled-down versions of this dataset (16×16 and 8×8 pixels), in order to study how our bounds behave as the dimension of the input changes. Although

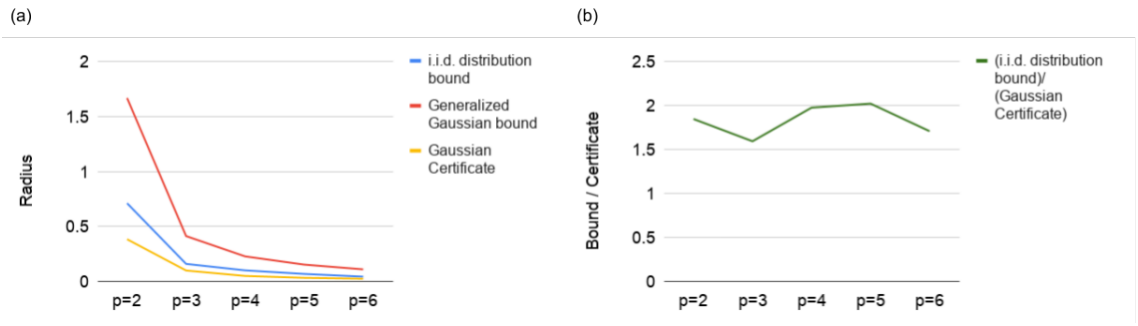


Figure 5.7: Repeating Figure 5.6 for $\sigma = .25$. At this level of noise, $\underline{p_1(x)}$ is low enough so that the i.i.d. distribution bound is tighter than the Generalized Gaussian bound (in contrast to the setup of Figure 5.6).

we do not have explicit certificates for these Generalized Gaussian distributions, we are able to compare the upper bounds derived in this work for any *possible* certificates to the *actual* certificates for Gaussian smoothing on the same images. Note that we re-trained the classifier on noisy images for each noise distribution and standard deviation σ.

Note also that our main results apply specifically to smoothing based certificates which are functions of only $p_1(x)$ and $p_2(x)$ (in theory, larger certificates could be derived if more information is available to the certification algorithm). In reporting the upper bounds on possible *empirical* certificates, we provide the same inputs to the upper bound as we would provide to the certificate: namely, an empirical lower bound $\underline{p_1(x)}$ on $p_1(x)$, estimated from samples, and an empirical upper bound $\overline{p_2(x)}$ on $p_2(x)$. We are *not* making claims about the "optimal possible" empirical estimation procedures required to derive the largest possible certificates. We instead regard these bounds, $\underline{p_1(x)}$ and $\overline{p_2(x)}$, as *inputs* to the empirical certificate: we are only claiming that, given estimates $\underline{p_1(x)}$ and $\overline{p_2(x)}$, no certificate will exceed the computed bound. In practice, we use the estimation procedure proposed by [36], which first selects a candidate top class label using a small number of samples, then uses a large number of samples ($100,000$ in our experiments) to compute

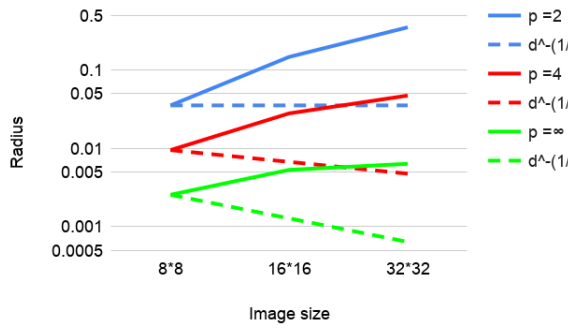


Figure 5.8: Certified Radius at different resolutions of CIFAR-10 using Gaussian noise ($\sigma = .12$). The increase in accuracy of the base classifier on higher-resolution images overcomes the inverse scaling with d in Eq. 5.1, achieving higher certified radii. Solid lines represent actual certificates and dashed lines represent how the certificates would scale if $\underline{p_1(x)}$ remained constant as resolution increased.

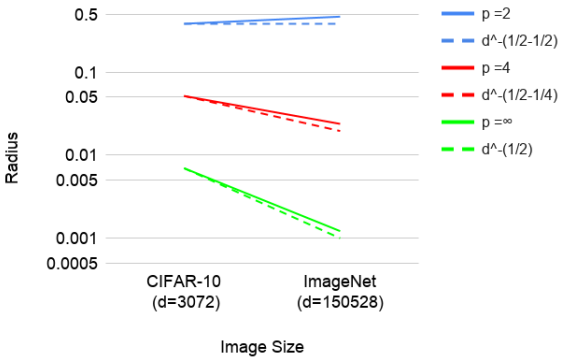


Figure 5.9: Certified Radius using Gaussian noise ($\sigma = .25$), for datasets of different image resolutions. We see that for $p > 2$, the certificates (solid lines) decrease with higher dimensionality almost as quickly as one would expect from the explicit dependence on d in Equation 5.1 (dashed lines).

$\underline{p_1(x)}$ based on a binomial distribution. $\overline{p_2(x)}$ is then taken as $1 - \underline{p_1(x)}$. Then, for the sake of our experiments, the only empirical input to our bound is the estimate of $\underline{p_1(x)}$.

One interesting result is that the distribution of noise added in each dimension seems to be largely irrelevant to determining $\underline{p_1(x)}$ (Figure 5.5). It is the variance of the noise added, *not* the specific choice of noise distribution, that determines $\underline{p_1(x)}$. This paints an even bleaker picture for the possibility of smoothing for high p-norm robustness than our theoretical results alone can: Theorems 7 and 8 still depend on $p_1(x)$ and $p_2(x)$ for the particular noise distribution used. This leaves open the possibility that certain choices of noise distributions could yield values of $p_1(x)$ large enough to counteract the scaling with p. However, empirically, we find that this is not the case: for a fixed σ, $p_1(x)$ does not depend on the shape of the smoothing distribution.

For example, one might attempt to use smoothing with $q = p$ in order to certify

for the ℓ_p norm, so that the level sets of the smoothing distribution correspond to ℓ_p balls around x. This is the technique used for ℓ_1 certification by [37], and for ℓ_2 certification by [36]. However, we find (Figures 5.6, 5.7) that, as anticipated by Figure 5.2, for $p > 2$, this can only achieve at best a constant factor improvement in certified robustness compared to simply using Gaussian smoothing with the certificate from [36] and applying equivalence of norms (Equation 5.1). Note that, as shown in Figure 5.5, it was *only* for the lowest level of noise tested ($\sigma = .12$) and the highest resolution images tested (32×32) that $\underline{p_1(x)}$ was sufficiently close to 1 for the Generalized Gaussian bound to be tighter than the i.i.d. distribution bound (Figure 5.6). For all other configurations (Figure 5.7, other plots are given in supplementary materials) the i.i.d. bound is tighter.

In the case of Gaussian smoothing, [36] makes an argument that, as image resolution increases, the base classifier will become more tolerant to noise, because information will be redundantly encoded in the additional pixels. This should allow us to increase the magnitude of the smoothing variance σ^2 proportionally to d. It is because by average-pooling back down a large image to a low-resolution one, the variance in each pixel of the smaller image will decrease proportionally with d. Then, if it is possible to classify noisy images at the lower resolution with a certain accuracy $p_1(x)$, it should be possible to classify images at the higher resolution with higher levels of noise. This increase in the amount of noise that can be added to high resolution images (to obtain roughly the same accuracy to that of low resolution ones) will cancel out the decrease in the robustness radius due to the curse of dimensionality explained in this paper. It is because based on Equation 5.1, if σ is allowed to scale with $\sqrt{d}$ with $p_1(x)$ and $p_2(x)$ unchanged, then the certified radius should even remain constant with d in the ℓ_∞ case.

For image datasets that are *identical* except for a scaling factor, we observe a related phenomenon: for a fixed noise variance, $p_1(x)$ tends to increase with the resolution of the image (i.e., the dimensionality of the input), and therefore the certified radii tend to increase with d in the $p = 2$ case. In Figure 5.8, we show that, for $p > 2$, this increase is enough to counteract the *inverse* scaling with d in Equation 5.1, at least in the case of low-resolution CIFAR-10 images. In other words, we still get larger certificates for larger-resolution images, simply because our base classifier becomes more accurate on noisy images as resolution increases. We emphasize that this is using the standard Gaussian noise: we have demonstrated that other i.i.d distributions will not give significantly better certificates.

The above setup, however, is an artificial scenario. In the real world, higher-resolution datasets are typically used for classification tasks which could *not* be accomplished with high accuracy at a lower resolution. As shown in Figure 5.9, if we compare, for a fixed σ, a real-world low dimensional classification task (CIFAR-10, $d = 3072$) to a high dimensional classification task (ImageNet, $d = 150528$), we see that the certified radius (and therefore $p_1(x)$), does *not* substantially increase with higher resolution. Therefore, for higher p-norms, the certified radius decreases with dimension with a scaling nearly as extreme as the explicit $d^{(1/2-1/p)}$ factor in Equation 5.1. Therefore, in practice, the curse of dimensionality can be observed as explained in this paper and it cannot be overcome using a novel choice of i.i.d. smoothing distribution.

5.7 Conclusion

In this work, we demonstrated some limitations of common smoothing distributions for ℓ_p-norm bounded adversaries when $p > 2$. We partially answer the question, raised in [36], whether smoothing techniques similar to Gaussian smoothing can be employed to achieve certifiable robustness guarantees for a general ℓ_p-norm bounded adversary. Most i.i.d. smoothing distributions fail to yield good robustness guarantees in the high-dimensional regime against ℓ_p-norm bounded attacks when $p > 2$. Their performance is no better than that of Gaussian smoothing up to a constant factor. While a constant factor improvement in performance could be critical in certain applications, the focus of this work is on the effect of dimensionality on certified robustness. We note that, in our analysis, we focus on i.i.d. and symmetric smoothing distributions. Our analysis highlights the importance of developing input-dependent smoothing techniques rather than the current smoothing methods based on i.i.d. distributions.

5.8 Appendices

A Proof for lemma 11

Proof. Applying the series expansion of e^{tZ}, we get,

$$
\begin{aligned}
E[e^{tZ}] &= \sum_{n=0}^{\infty} \frac{t^n E[Z^n]}{n!} \\
E[Z^n] &= \frac{1}{C} \int_{-\infty}^{\infty} z^n e^{-(|z|/b)^q} dz \\
&= \frac{1}{C} \int_{0}^{\infty} (1 + (-1)^n) z^n e^{-z^q/b^q} dz \\
&= \begin{cases} 0, & n \text{ is odd} \\ \frac{2}{C} \int_0^{\infty} z^n e^{-z^q/b^q} dz, & n \text{ is even} \end{cases}
\end{aligned}
$$

When n is even:

$$
\begin{aligned}
E[Z^n] &= \frac{2}{C} \int_{0}^{\infty} z^n e^{-z^q/b^q} dz \\
&= \frac{2b^{n+1}\Gamma\left(\frac{n+1}{q}\right)}{Cq}
\end{aligned}
$$

Substituting C,

$$
\begin{aligned}
E[Z^n] &= \frac{b^n \Gamma\left(\frac{n+1}{q}\right)}{\Gamma(1/q)} \leq b^n \Gamma(n+1) && \text{for } q \geq 1 \\
E[Z^n] &\leq b^n n!
\end{aligned}
$$

Therefore, keeping only the terms with even n in the expansion of $E[e^{tZ}]$, we get:

$$\begin{aligned} E[e^{tZ}] &\leq \sum_{m=0}^{\infty} (t^2 b^2)^m \\ &= \sum_{m=0}^{\infty} \left(\frac{t^2 \sigma^2 \Gamma(1/q)}{\Gamma(3/q)} \right)^m && \text{using } \sigma^2 = \tfrac{b^2 \Gamma(3/p)}{\Gamma(1/p)} \\ &\leq \sum_{m=0}^{\infty} (c^2 t^2 \sigma^2)^m \end{aligned}$$

for some positive constant $c < 1.85$, because,

$$\begin{aligned} \frac{\Gamma(1/q)}{\Gamma(3/q)} &= \frac{3q\Gamma(1+1/q)}{q\Gamma(1+3/q)} && \text{(using } \Gamma(z+1) = z\Gamma(z)\text{)} \\ &= \frac{3\Gamma(1+1/q)}{\Gamma(1+3/q)} \\ &< 1.85^2 && \text{(for } q \geq 1, \Gamma(1+1/q) \leq 1 \text{ and } \Gamma(1+3/q) > 0.88\text{)} \end{aligned}$$

□

B Proof for lemma 12

Proof. The points in V_1 satisfy the following 2^d constraints:

$$\begin{aligned} x_1 + x_2 + \cdots + x_d &\leq b \\ -x_1 + x_2 + \cdots + x_d &\leq b \\ x_1 - x_2 + \cdots + x_d &\leq b \\ -x_1 - x_2 + \cdots + x_d &\leq b \\ &\vdots \end{aligned}$$

$$-x_1 - x_2 - \cdots - x_d \leq b$$

Similarly, points in V_2 satisfy,

$$\begin{aligned}
(x_1 - \epsilon) + x_2 + \cdots + x_d &\leq b \\
-(x_1 - \epsilon) + x_2 + \cdots + x_d &\leq b \\
(x_1 - \epsilon) - x_2 + \cdots + x_d &\leq b \\
-(x_1 - \epsilon) - x_2 + \cdots + x_d &\leq b \\
&\vdots \\
-(x_1 - \epsilon) - x_2 - \cdots - x_d &\leq b
\end{aligned}$$

Then, the points in $V_1 \cap V_2$ must satisfy the following set of constraints constructed by picking constraints that have a $+$ sign for x_1 in the first set of constraints and a $-$ sign for x_1 in the second set.

$$\begin{aligned}
x_1 + x_2 + \cdots + x_d &\leq b \\
-(x_1 - \epsilon) + x_2 + \cdots + x_d &\leq b \\
x_1 - x_2 + \cdots + x_d &\leq b \\
-(x_1 - \epsilon) - x_2 + \cdots + x_d &\leq b \\
&\vdots
\end{aligned}$$

$$-(x_1 - \epsilon) - x_2 - \cdots - x_d \leq b$$

They may be rewritten as,

$$\begin{aligned}
(x_1 - \epsilon/2) + x_2 + \cdots + x_d &\leq b - \epsilon/2 \\
-(x_1 - \epsilon/2) + x_2 + \cdots + x_d &\leq b - \epsilon/2 \\
(x_1 - \epsilon/2) - x_2 + \cdots + x_d &\leq b - \epsilon/2 \\
-(x_1 - \epsilon/2) - x_2 + \cdots + x_d &\leq b - \epsilon/2 \\
&\vdots \\
-(x_1 - \epsilon/2) - x_2 - \cdots - x_d &\leq b - \epsilon/2
\end{aligned}$$

which define an ℓ_1 ball of radius $b - \epsilon/2$ centered at $(\epsilon/2, 0, \ldots, 0)$, that is, $\epsilon/2$ in the first coordinate and zero everywhere else. □

C Additional Plots of Certificate Upper Bounds

See Figure 5.10.

D Experimental Details

Our experiments are adapted from the released code for ℓ_2 smoothing from [36]. In particular, for each Generalized Gaussian distribution with varying parameter q and

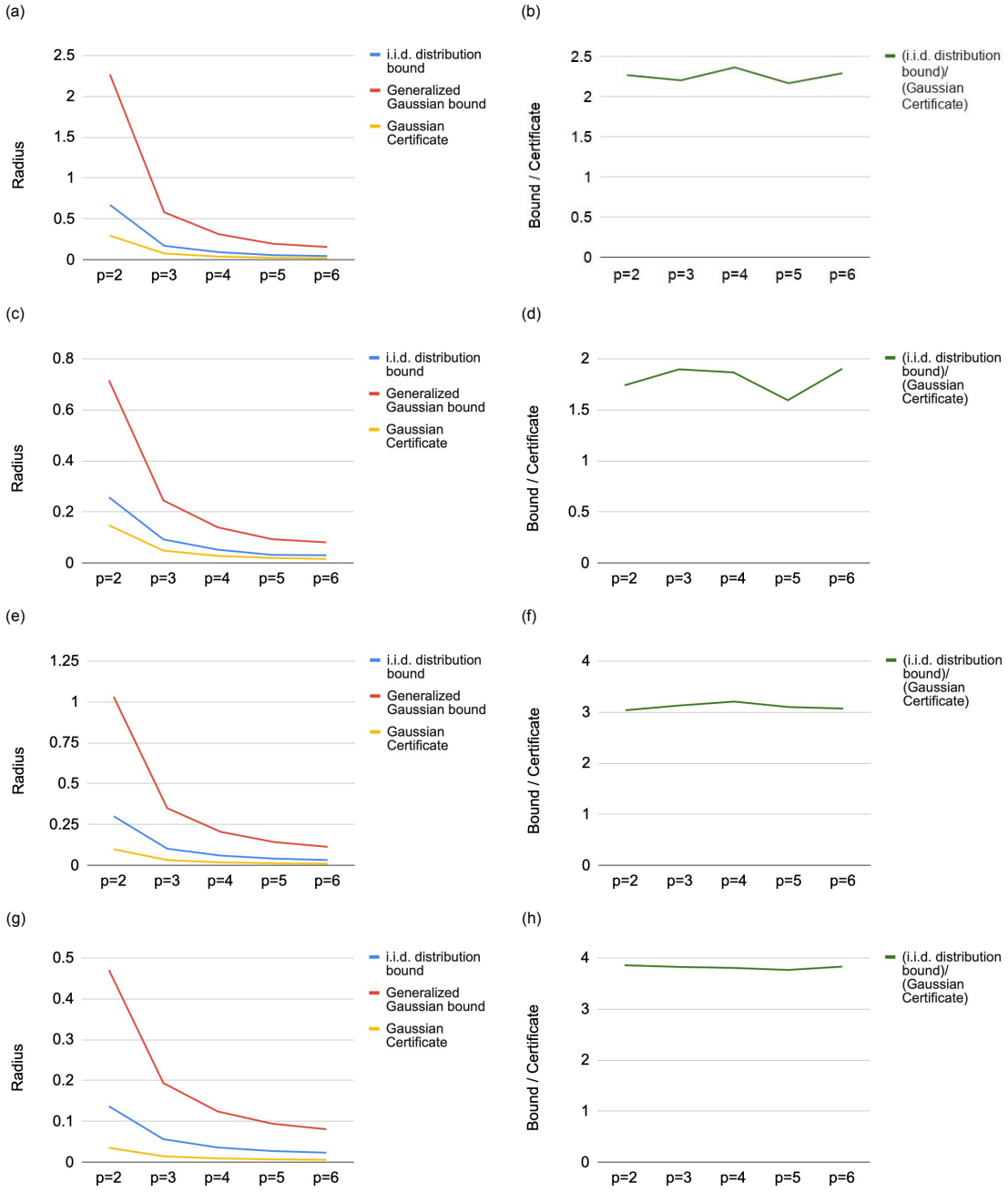


Figure 5.10: Upper bounds for certifying with Generalized Gaussian noise on CIFAR-10 images, with $q = p$, compared with certificates using Gaussian noise directly.Left panels show the certificates and the bounds directly, while right panels show the ratios between the i.i.d. distribution bounds (tighter in each case) and the certificates. Panels (a,b) use unaltered CIFAR-10 images with $\sigma = 0.5$ noise. Panels (c,d) and (e,f) use CIFAR-10 images at 16×16 scale with $\sigma = 0.12$ and $\sigma = 0.25$ respectively. Panels (g,h) use CIFAR-10 images at 8×8 scale with $\sigma = 0.12$.

standard deviation σ, we trained a ResNet-110 classifier on CIFAR-10 for 90 epochs, with the training under the same noise distribution as used for certification. All training and certification parameters are identical to those used in [36] unless otherwise specified. In particular, all certificates are reported to 99.9% confidence, and we tested using a 500-image subset of the CIFAR-10 test set. For lower-resolution versions of CIFAR-10, we again trained separate models for each resolution used, with the resolution at training time matching the resolution at test time. We first reduced the image resolutions before adding noise, then, once the noise was added, scaled the images back to the original 32×32 resolution (by repeating pixel values) before classifying with ResNet-110: this ensured that the number of parameters did not vary between classifiers.

We trained with $\sigma = 0.12, 0.25, 0.50, 1.00$ for resolutions 32×32, 16×16 and 8×8. At higher levels of noise for each scale ($\sigma = 0.25$ for 8×8, $\sigma = 0.5$ for 8×8 and 16×16, $\sigma = 1.00$ on all scales) the resulting classifiers could not correctly certify the median image ($\underline{p_1(x)} < .5$), so we do not report any certificates.

Values for ImageNet for the median certificate under Gaussian noise are adapted from the released certificate data from [36].

Chapter 6: Certifying Neural Network Confidence

6.1 Introduction

Deep neural networks have been shown to be vulnerable to adversarial attacks, in which a nearly imperceptible perturbation is added to an input image to completely alter the network's prediction [1, 2, 3, 4]. Several empirical defenses have been proposed over the years to produce classifiers that are robust to such perturbations [18, 19, 20, 21, 22, 23, 98]. However, without robustness guarantees, it is often the case that these defenses are broken by stronger attacks [24, 25, 26, 141]. Certified defenses, such as those based on convex-relaxation [27, 28, 29, 30, 31] and interval-bound propagation [32, 33, 34, 35], address this issue by producing robustness guarantees within a neighborhood of an input point. However, due to the complexity of present-day neural networks, these methods have seen limited use in high-dimensional datasets such as ImageNet.

Randomized smoothing has recently emerged as the state-of-the-art technique for certifying adversarial robustness with the scalability to handle datasets as large as ImageNet [36, 37, 38, 39]. This defense uses a base classifier, e.g. a deep neural network, to make predictions. Given an input image, a smoothing method queries the top class label at a large number of points in a Gaussian distribution surrounding the image, and returns the label with the majority vote. If the input image is perturbed slightly, the new voting

population overlaps greatly with the smoothing distribution around the original image, and so the vote outcome can change only a small amount.

Conventional smoothing throws away a lot of information about class labels, and has limited capabilities that make its outputs difficult to use for decision making. Conventional classification networks with a softmax layer output a confidence score that can be interpreted as the degree of certainty the network has about the class label [142]. This is a crucial piece of information in real world decision-making applications such as self-driving cars [143] and disease-diagnosis networks [144], where safety is paramount.

In contrast, standard Gaussian smoothing methods take binary votes at each randomly sampled point – i.e., each point votes either for or against the most likely class, without conveying any information about how confident the network is in the class label. This may lead to scenarios where a point has a large certified radius but the underlying classifier has a low confidence score. For example, imagine a 2-way classifier for which a large portion, say 95%, of the sampled points predict the same class. In this case, the certified radius will be very large (indicating that this image is not an ℓ_2-bounded adversarial example). However, it could be that each point predicts the top class with very low confidence. In this case, one should have very low confidence in the class label, despite the strength of the adversarial certificate. A Gaussian smoothing classifier counts a 51% confidence vote exactly the same way as a 99% confidence vote, and this important information is erased.

In this work, we restore confidence information in certified classifiers by proposing a method that produces class labels with a *certified confidence score*. Instead of taking a vote at each Gaussian sample around the input point, we average the confidence scores from the underlying base classifier for each class. The prediction of our smoothed classifier

is given by the argmax of the expected scores of all the classes. Using the probability distribution of the confidence scores under the Gaussian, we produce a lower bound on how much the expected confidence score of the predicted class can be manipulated by a bounded perturbation to the input image. To do this, we adapt the Neyman-Pearson lemma, the fundamental theorem that characterizes the worst-case behaviour of the classifier under regular (binary) voting, to leverage the distributional information about the confidence scores. The lower bound we obtain is monotonically decreasing with the ℓ_2-norm of the perturbation and can be expressed as a linear combination of the Gaussian CDF at different points. This allows us to design an efficient binary search based algorithm to compute the radius within which the expected score is guaranteed to be above a given threshold. Our method endows smoothed classifiers with the new and important capability of producing confidence scores.

We study two notions of measuring confidence: the *average prediction score* of a class, and the *margin* by which the average prediction score of one class exceeds that of another. The average prediction score is the expected value of the activations in the final softmax-layer of a neural network under the smoothing distribution. A class is guaranteed to be the predicted class if its average prediction score is greater than one half (since softmax values add up to one) or it maintains a positive margin over all the other classes. For both these measures, along with the bounds described in the previous paragraph, we also derive naive lower bounds on the expected score at a perturbed input point that do not use the distribution of the scores. We perform experiments on CIFAR-10 and ImageNet datasets which show that using information about the distribution of the scores allows us to achieve better certified guarantees than the naive method.

Related work: Randomized smoothing as a technique to design certifiably robust machine learning models has been studied amply in recent years. It has been used to produce certified robustness against additive threat models, such as, ℓ_1 [37, 121], ℓ_2 [36, 122, 123] and ℓ_0-norm [69, 124] bounded adversaries, as well as non-additive threat models, such as, Wasserstein Adversarial attacks [82]. A derandomized version has been shown to provide robustness guarantees for patch attacks [68]. Smoothed classifiers that use the average confidence scores have been studied in [39] to achieve better certified robustness through adversarial training. A recent work uses the median score to generated certified robustness for regression models [125]. Differential privacy based defense method studied in [37] is capable of providing a guarantee on the test accuracy of a robust model under adversarial attack. Various limitations of randomized smoothing, like its inapplicability to high-dimensional problems for ℓ_∞-robustness, have been studied in [50, 140, 145].

6.2 Background and Notation

Gaussian smoothing, introduced by Cohen et al. in 2019, relies on a "base classifier," which is a mapping $f : \mathbb{R}^d \to \mathcal{Y}$ where $\mathbb{R}^d$ is the input space and $\mathcal{Y}$ is a set of k classes. It defines a smoothed classifier $\bar{f}$ as

$$\bar{f}(x) = \underset{c \in \mathcal{Y}}{\operatorname{argmax}}\ \mathbb{P}(f(x + \delta) = c)$$

where $\delta \sim \mathcal{N}(0, \sigma^2 I)$ is sampled from an isometric Gaussian distribution with variance σ^2. It returns the class that is most likely to be sampled by the Gaussian distribution

centered at point x. Let p_1 and p_2 be the probabilities of sampling the top two most likely classes. Then, $\bar{f}$ is guaranteed to be constant within an ℓ_2-ball of radius

$$R = \frac{\sigma}{2}\left(\Phi^{-1}(p_1) - \Phi^{-1}(p_2)\right)$$

where Φ^{-1} is the inverse CDF of the standard Gaussian distribution [36]. For a practical certification algorithm, a lower bound $\underline{p_1}$ on p_1 and an upper bound $\overline{p_2} = 1 - \underline{p_1}$ on p_2, with probability $1 - \alpha$ for a given $\alpha \in (0, 1)$, are obtained and the certified radius is given by $R = \sigma\Phi^{-1}(\underline{p_1})$. This analysis is tight for ℓ_2 perturbations; the bound is achieved by a worst-case classifier in which all the points in the top-class are restricted to a half-space separated by a hyperplane orthogonal to the direction of the perturbation.

In our discussion, we diverge from the standard notation described above, and assume that the base classifier f maps points in $\mathbb{R}^d$ to a k-tuple of confidence scores. Thus, $f : \mathbb{R}^d \to (a, b)^k$ for some $a, b \in \mathbb{R}$ and $a < b$[1]. We define the smoothed version of the classifier as

$$\bar{f}(x) = \mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}[f(x + \delta)],$$

which is the expectation of the class scores under the Gaussian distribution centered at x. The final prediction is made by taking an argmax of the expected scores. This definition has been studied by Salman et al. in [39] to develop an attack against smoothed classifiers which when used in an adversarial training setting helps boost the performance of conventional smoothing. The goal of this work is to identify a radius around an image x within which the expected confidence score of the predicted class i, i.e. $\bar{f}_i(x) =$

[1] (a, b) denotes the open interval between a and b.

$\mathbb{E}[f_i(x+\delta)]$, remains above a given threshold $c \in (a,b)$[2].

We measure confidence using two different notions. The first measure is the average prediction score of a class as output by the final softmax layer. We denote the prediction score function with $h : \mathbb{R}^d \to (0,1)^k$ and define the average for class i as $\bar{h}_i(x) = \mathbb{E}[h_i(x+\delta)]$. The second one is the margin $m_i(x) = h_i(x) - \max_{j \neq i} h_j(x)$ by which class i beats every other class in the softmax prediction score. In section 6.4, we show that the expected margin $\bar{m}_i(x) = \mathbb{E}[m_i(x+\delta)]$ for the predicted class is a lower-bound on the gap in average prediction scores of the top two class labels. Thus, $\bar{m}_i(x) > 0$ implies that i is the predicted class.

6.3 Certifying Confidence Scores

Standard Gaussian smoothing for establishing certified class labels essentially works by averaging binary (0/1) votes from every image in a Gaussian cloud around the input image, x. It then establishes the worst-case class boundary given the recorded vote, and produces a certificate. The same machinery can be applied to produce a naive certificate for confidence score; rather than averaging binary votes, we simply average scores. We then produce the worst-case class distribution, in which each class lives in a separate half-space, and generate a certificate for this worst case.

However, the naive certificate described above throws away a lot of information. When continuous-values scores are recorded, we obtain not only the average score, but also the *distribution* of scores around the input point. By using this distributional information, we can potentially create a much stronger certificate.

[2] $f_i(x)$ denotes the ith component of $f(x)$

To see why, consider the extreme case of a "flat" classifier function for which every sample in the Gaussian cloud around x returns the same top-class prediction score of 0.55. In this case, the average score is 0.55 as well. For a function where the *distribution* of score votes is concentrated at 0.55 (or any other value great than ½), the average score will always remain at 0.55 for *any* perturbation to x, thus yielding an infinite certified radius. However, when using the naive approach that throws away the distribution, the worst-case class boundary with average vote 0.55 is one with confidence score 1.0 everywhere in a half-space occupying 0.55 probability, and 0.0 in a half-space with 0.45 probability. This worst-case, which uses only the average vote, produces a very small certified radius, in contrast to the infinite radius we could obtain from observing the distribution of votes.

Below, we first provide a simple bound that produces a certificate by averaging scores around the input image, and directly applying the framework from [36]. Then, we describe a more refined method that uses distributional information to obtain stronger bounds.

6.3.1 A baseline method using Gaussian means

In this section, we describe a method that uses only the average confidence over the Gaussian distribution surrounding x, and not the distribution of values, to bound how much the expected score can change when x is perturbed with an ℓ_2 radius of R units. This is a straightforward extension of Cohen et al.'s [36] work to our framework. It shows that regardless the behaviour of the base classifier f, its smoothed version $\bar{f}$ changes slowly which is similar to the observation of bounded Lipschitz-ness made by Salman et al.

in [39] (Lemma 2). The worst-case classifier in this case assumes value a in one half space and b in other, with a linear boundary between the two as illustrated in figure 6.1. The following theorem formally states the bounds, the proof of which is deferred to the appendix[3].

Theorem 11. *Let $\underline{e_i}(x)$ and $\overline{e_i}(x)$ be a lower-bound and an upper-bound respectively on the expected score $\bar{f}_i(x)$ for class i and, let $\underline{p_i}(x) = \frac{\underline{e_i}(x)-a}{b-a}$ and $\overline{p_i}(x) = \frac{\overline{e_i}(x)-a}{b-a}$. Then, for a perturbation x' of the input x, such that, $\|x' - x\|_2 \leq R$,*

$$\bar{f}_i(x') \geq b\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R) + a(1 - \Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R)) \tag{6.1}$$

and

$$\bar{f}_i(x') \leq b\Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_i}(x)) + R) + a(1 - \Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_i}(x)) + R))$$

where Φ_σ is the CDF of the univariate Gaussian distribution with σ^2 variance, i.e., $\mathcal{N}(0, \sigma^2)$.

6.3.2 Proposed certificate

The bounds in section 6.3.1 are a simple application of the Neyman-Pearson lemma to our framework. But this method discards a lot of information about how the class scores are distributed in the Gaussian around the input point. Rather than consolidating the confidence scores from the samples into an expectation, we propose a method that uses the cumulative distribution function of the confidence scores to obtain improved bounds

[3] A separate proof, using Lemma 2 from Salman et al. in [39], for this theorem for $\sigma = 1$ is also included in the appendix.

on the expected class scores.

Given an input x, we draw m samples from the Gaussian distribution around x. We use the prediction of the base classifier f on these points to generate bounds on the distribution function of the scores for the predicted class. These bounds, in turn, allow us to bound the amount by which the expected score of the class will decrease under an ℓ_2 perturbation. Finally, we apply binary search to compute the radius for which this lower bound on the expected score remains above c.

Consider the sampling of scores around an image x using a Gaussian distribution. Let the probability with which the score of class i is above s be

$$p_{i,s}(x) = \mathop{\mathbb{P}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}(f_i(x+\delta) \geq s).$$

For point x and class i, consider the random variable $Z = -f_i(x+\delta)$ where $\delta \sim \mathcal{N}(0, \sigma^2 I)$. Let $F(s) = \mathbb{P}(Z \leq s)$ be the cumulative distribution function of Z and $F_m(s) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\{Z_j \leq s\}$ be its empirical estimate. For a given $\alpha \in (0,1)$, the Dvoretzky–Kiefer–Wolfowitz inequality [146] says that, with probability $1-\alpha$, the true CDF is bounded by the empirical CDF as follows:

$$F_m(s) - \epsilon \leq F(s) \leq F_m(s) + \epsilon, \forall s,$$

where $\epsilon = \sqrt{\frac{\ln 2/\alpha}{2m}}$. Thus, $p_{i,s}(x)$ is also bounded within $\pm\epsilon$ of its empirical estimate $\sum_{j=1}^{m} \mathbf{1}\{f_i(x+\delta_j) \geq s\}$.

The following theorem bounds the expected class score under an ℓ_2 perturbation

using bounds on the cumulative distribution of the scores.

Theorem 12. *Let, for class i, $a < s_1 \leq s_2 \leq \cdots \leq s_n < b$ be n real numbers and let $\overline{p_{i,s_j}}(x)$ and $\underline{p_{i,s_j}}(x)$ be upper and lower bounds on $p_{i,s_j}(x)$ respectively derived using the Dvoretzky–Kiefer–Wolfowitz inequality, with probability $1-\alpha$, for a given $\alpha \in (0,1)$. Then, for a perturbation x' of the input x, such that, $\|x'-x\|_2 \leq R$,*

$$\bar{f}_i(x') \geq a+(s_1-a)\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_{i,s_1}}(x))-R)+\sum_{j=2}^{n}(s_j-s_{j-1})\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_{i,s_j}}(x))-R) \quad (6.2)$$

and

$$\bar{f}_i(x') \leq s_1+(b-s_n)\Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_{i,s_n}}(x))+R)+\sum_{j=1}^{n-1}(s_{j+1}-s_j)\Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_{i,s_j}}(x))+R)$$

where Φ_σ is the CDF of the univariate Gaussian distribution with σ^2 variance, i.e., $\mathcal{N}(0,\sigma^2)$.

The above bounds are tight for ℓ_2 perturbations. The worst-case classifier for the lower bound is one in which the class score decreases from b to a in steps, taking values $s_n, s_{n-1}, \ldots, s_1$ at each level. Figure 6.2 illustrates this case for three intermediate levels. A similar worst-case scenario can be constructed for the upper bound as well where the class score increases from a to b along the direction of the perturbation. Even though our theoretical results allow us to derive both upper and lower bounds for the expected scores, we restrict ourselves to the lower bound in our experimental results. We provide a proof sketch for this theorem in section 6.3.3. Our experimental results show that the CDF-based approach beats the naive bounds in practice by a significant margin, showing

that having more information about the classifier at the input point can help achieve better guarantees.

Computing the certified radius Both the bounds in theorem 12 monotonic in R. So, in order to find a certified radius, up to a precision τ, such that the lower (upper) bound is above (below) a certain threshold we can apply binary search which will require at most $O(\log(1/\tau))$ evaluations of the bound.

6.3.3 Proof of Theorem 12

We present a brief proof for theorem 12. We use a slightly modified version of the Neyman-Pearson lemma (stated in [36]) which we prove in the appendix.

Lemma 13 (Neyman & Pearson, 1933)**.** *Let X and Y be random variables in $\mathbb{R}^d$ with densities μ_X and μ_Y. Let $h : \mathbb{R}^d \to (a, b)$ be a function. Then:*

1. *If $S = \left\{ z \in \mathbb{R}^d \mid \frac{\mu_Y(z)}{\mu_X(z)} \leq t \right\}$ for some $t > 0$ and $\mathbb{P}(h(X) \geq s) \geq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) \geq s) \geq \mathbb{P}(Y \in S)$.*

2. *If $S = \left\{ z \in \mathbb{R}^d \mid \frac{\mu_Y(z)}{\mu_X(z)} \geq t \right\}$ for some $t > 0$ and $\mathbb{P}(h(X) \geq s) \leq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) \geq s) \leq \mathbb{P}(Y \in S)$.*

Set X to be the smoothing distribution at an input point x and Y to be that at $x + \epsilon$ for some perturbation vector ϵ. For a class i, define sets $\underline{S}_{i,j} = \{z \in \mathbb{R}^d \mid \mu_Y(z)/\mu_X(z) \leq t_{i,j}\}$ for some $t_{i,j} > 0$, such that, $\mathbb{P}(X \in \underline{S}_{i,j}) = \underline{p_{i,s_j}}(x)$. Similarly, define sets $\overline{S}_{i,j} = \{z \in \mathbb{R}^d \mid \mu_Y(z)/\mu_X(z) \geq t'_{i,j}\}$ for some $t'_{i,j} > 0$, such that, $\mathbb{P}(X \in \overline{S}_{i,j}) = \overline{p_{i,s_j}}(x)$. Since, $\mathbb{P}(f_i(X) \geq s_j) \geq \mathbb{P}(X \in \underline{S}_{i,j})$, using lemma 13 we can say that $\mathbb{P}(f_i(Y) \geq s_i) \geq$

$\mathbb{P}(Y \in \underline{S}_{i,j})$. Therefore,

$$\begin{aligned}
\mathbb{E}[f_i(Y)] &\geq s_n \mathbb{P}(f_i(Y) \geq s_n) + s_{n-1}(\mathbb{P}(f_i(Y) \geq s_{n-1}) - \mathbb{P}(f_i(Y) \geq s_n)) \\
&\quad + \cdots + s_1(\mathbb{P}(f_i(Y) \geq s_1) - \mathbb{P}(f_i(Y) \geq s_2)) + a(1 - \mathbb{P}(f_i(Y) \geq s_1)) \\
&= a + (s_1 - a)\mathbb{P}(f_i(Y) \geq s_1) + \sum_{j=2}^{n}(s_j - s_{j-1})\mathbb{P}(f_i(Y) \geq s_j) \\
&\geq a + (s_1 - a)\mathbb{P}(Y \in \underline{S}_{i,1}) + \sum_{j=2}^{n}(s_j - s_{j-1})\mathbb{P}(Y \in \underline{S}_{i,j}).
\end{aligned}$$

Similarly, $\mathbb{P}(f_i(X) \geq s_j) \leq \mathbb{P}(X \in \overline{S}_{i,j})$ implies $\mathbb{P}(f_i(Y) \geq s_j) \leq \mathbb{P}(Y \in \overline{S}_{i,j})$ as per lemma 13. Therefore,

$$\begin{aligned}
\mathbb{E}[f_i(Y)] &\leq b\mathbb{P}(f_i(Y) \geq s_n) + s_n(\mathbb{P}(f_i(Y) \geq s_{n-1}) - \mathbb{P}(f_i(Y) \geq s_n)) \\
&\quad + \cdots + s_1(1 - \mathbb{P}(f_i(Y) \geq s_1)) \\
&= (b - s_n)\mathbb{P}(f_i(Y) \geq s_n) + \sum_{j=1}^{n-1}(s_{j+1} - s_j)\mathbb{P}(f_i(Y) \geq s_j) + s_1 \\
&\leq s_1 + (b - s_n)\mathbb{P}(Y \in \overline{S}_{i,n}) + \sum_{j=1}^{n-1}(s_{j+1} - s_j)\mathbb{P}(Y \in \overline{S}_{i,j}).
\end{aligned}$$

Since, we are smoothing using an isometric Gaussian distribution with σ^2 variance, $\mu_X = \mathcal{N}(x, \sigma^2 I)$ and $\mu_Y = \mathcal{N}(x + \epsilon, \sigma^2 I)$. Then, for some t and β

$$\begin{aligned}
\frac{\mu_Y(z)}{\mu_Y(z)} \leq t &\iff \epsilon^T z \leq \beta \\
\frac{\mu_Y(z)}{\mu_Y(z)} \geq t &\iff \epsilon^T z \geq \beta.
\end{aligned}$$

Thus, each of the sets $\underline{S}_{i,j}$ and $\overline{S}_{i,j}$ is a half space defined by a hyper-plane orthogonal to

the direction of the perturbation. This simplifies our analysis to one dimension, namely, the one along the perturbation. For each of the sets $\underline{S}_{i,j}$ and $\overline{S}_{i,j}$, we can find a point on the real number line $\Phi_\sigma^{-1}(\underline{p_{i,s_j}}(x))$ and $\Phi_\sigma^{-1}(\overline{p_{i,s_j}}(x))$ respectively such that the probability of a Gaussian sample to fall in that set is equal to the Gaussian CDF at that point. Therefore,

$$\bar{f}_i(x+\epsilon) \geq a + (s_1 - a)\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_{i,s_1}}(x)) - R) + \sum_{j=2}^{n}(s_j - s_{j-1})\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_{i,s_j}}(x)) - R)$$

and

$$\bar{f}_i(x+\epsilon) \leq s_1 + (b - s_n)\Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_{i,s_n}}(x)) + R) + \sum_{j=1}^{n-1}(s_{j+1} - s_j)\Phi_\sigma(\Phi_\sigma^{-1}(\overline{p_{i,s_j}}(x)) + R)$$

which completes the proof of theorem 12. We would like to note here that although we use the Gaussian distribution for smoothing, the modified Neyman-Pearson lemma does not make any assumptions on the shape of the distributions which allows for this proof to be adapted for other smoothing distributions as well.

6.4 Confidence measures

We study two notions of confidence: average prediction score of a class and the margin of average prediction score between two classes. Usually, neural networks make their predictions by outputting a prediction score for each class and then taking the argmax of the scores. Let $h : \mathbb{R}^d \to (0,1)^k$ be a classifier mapping input points to prediction scores between 0 and 1 for each class. We assume that the scores are generated by a softmax-like layer, i.e., $0 < h_i(x) < 1, \forall i \in \{1, \ldots, k\}$ and $\sum_i h_i(x) = 1$. For $\delta \sim$

$\mathcal{N}(0, \sigma^2 I)$, we define average prediction score for a class i as

$$\bar{h}_i(x) = \mathbb{E}[h_i(x + \delta)].$$

The final prediction for the smoothed classifier is made by taking an argmax over the average prediction scores of all the classes, i.e., $\text{argmax}_i\ \bar{h}_i(x)$. Thus, if for a class j, $\bar{h}_j(x) \geq 0.5$, then $j = \text{argmax}_i\ \bar{h}_i(x)$.

Now, we define margin m at point x for a class i as

$$m_i(x) = h_i(x) - \max_{j \neq i} h_j(x).$$

Thus, if i is the class with the highest prediction score, $m_i(x)$ is the lead it has over the second highest class (figure 6.4). And, for any other class $m_i(x)$ is the negative of the difference of the scores of that class with the highest class. We define average margin at point x under smoothing distribution $\mathcal{P}$ as

$$\bar{m}_i(x) = \mathbb{E}[m_i(x + \delta)].$$

For a pair of classes i and j, we have,

$$\begin{aligned}
\bar{h}_i(x) - \bar{h}_j(x) &= \mathbb{E}[h_i(x + \delta)] - \mathbb{E}[h_j(x + \delta)] \\
&= \mathbb{E}[h_i(x + \delta) - h_j(x + \delta)] \\
&\geq \mathbb{E}[h_i(x + \delta) - \max_{j \neq i} h_j(x + \delta)]
\end{aligned}$$

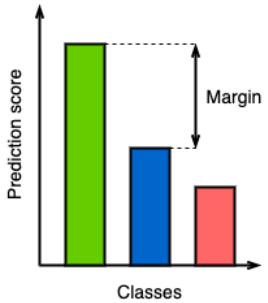


Figure 6.4: Margin

$$= \mathbb{E}[m_i(x+\delta)] = \bar{m}_i(x)$$

$$\bar{h}_i(x) \geq \bar{h}_j(x) + \bar{m}_i(x).$$

Thus, if $\bar{m}_i(x) > 0$, then class i must have the highest average prediction score making it the predicted class under this notion of confidence measure.

6.5 Experiments

We conduct several experiments to motivate the use of certified confidence, and to validate the effectiveness of our proposed CDF-based certificate.

6.5.1 Does certified radius correlate with confidence score?

A classifier can fail because of an adversarial attack, or because of epistemic uncertainty – a class label may be uncertain or wrong because of lack of useful features, or because the model was not trained on sufficient representative data. The use of certified confidence is motivated by the observation that the original Gaussian averaging, which certifies the *security* of class labels, does not convey whether the user should be *confident* in the label because it neglects epistemic uncertainty. We demonstrate this with a simple experiment. In figure 6.5, we show plots of softmax prediction score vs. certified radius obtained using smoothed ResNet-110 and ResNet-50 classifiers trained by Cohen et al. in [36] for CIFAR-10 and ImageNet respectively. The noise level σ used for this experiment was 0.25. For both models, the certified radii correlate very little with the prediction scores for the input images. The CIFAR-10 plot has points with high scores but small radii.

While, for ImageNet, we see a lot of points with low scores but high radii. This motivates the need for certifying confidence; high radius does not imply high confidence of the underlying classifier. This lack of correlation is visualized in figure 6.6.

In the plots, CIFAR-10 images tend to have a higher prediction score than ImageNet images which is potentially due to the fact that the ImageNet dataset has a lot more classes than the CIFAR-10 dataset, driving the softmax scores down. There is a hard limit (~ 0.95 for ImageNet) on the largest radius that can be generated by Cohen et al.'s certifying algorithm which causes a lot of the ImageNet points to accumulate at this radius value. This limit comes from the fact that even if all the samples around an input image vote for the same class, the lower-bound on the top-class probability is strictly less than one, which keeps the certified radius within a finite value.

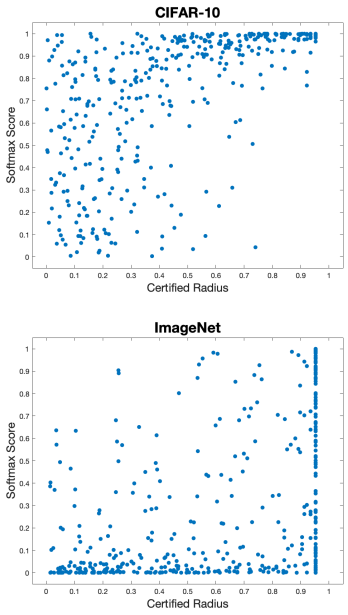


Figure 6.5: Prediction Score vs. Certified Radius.

6.5.2 Evaluating the strength of bounds

We use the ResNet-110 and ResNet-50 models trained by Cohen et al. in [36] on CIFAR-10 and ImageNet datasets respectively to generate confidence certificates. These models have been pre-trained with varying Gaussian noise level σ in the training data. We use the same σ for certifying confidences as well. We use the same number of samples $m = 100,000$ and value of $\alpha = 0.001$ as in [36]. We set $s_1, s_2 \ldots, s_n$

in theorem 12 such that the number of confidence score values falling in each of the intervals $(a, s_1), (s_1, s_2), \ldots, (s_n, b)$ is the same. We sort the scores from the m samples in increasing order and set s_i to be the element at position $1 + (i-1)m/n$ in the order. We chose this method of splitting the range (a, b), instead of at regular steps, to keep the intervals well-balanced. We present results for both notions of confidence measure: average prediction score and margin. Figure 6.11 plots certified accuracy, using the naive bound and the CDF-based method, for different threshold values for the top-class average prediction score and the margin at various radii for $\sigma = 0.25$. The same experiments for $\sigma = 0.50$ have been included in the appendix.

Each line is for a given threshold for the confidence score. The solid lines represent certificates derived using the CDF bound and the dashed lines are for ones using the naive bound. For the baseline certificate (6.1), we use Hoeffding's inequality to get a lower-bound on the expected top-class confidence score $e_i(x)$, that holds with probability $1-\alpha$, for a given $\alpha \in (0, 1)$.

$$\underline{e_i}(x) = \frac{1}{m}\sum_{j=1}^{m} f_i(x + \delta_j) - (b-a)\sqrt{\frac{\ln(1/\alpha)}{2m}}$$

This bound is a reasonable choice because $\underline{p_i}(x)$ differs from the empirical estimate by the same amount $\sqrt{\ln(1/\alpha)/2m}$ as $\underline{p_{i,s}}(x)$ in the proposed CDF-based certificate. In the appendix, we also show that the baseline certificate, even with the best-possible lower-bound for $e_i(x)$, cannot beat our method for most cases.

We see a significant improvement in certified accuracy (e.g. at radius = 0.25) when certification is done using the CDF method instead of the naive bound. The confidence

measure based on the margin between average prediction scores yields slightly better certified accuracy when thresholded at zero than the other measure.

6.6 Conclusion

While standard certificates can guarantee that a decision is *secure*, they contain little information about how *confident* the user should be in the assigned label. We present a method that certifies the confidence scores, rather than the labels, of images. By leveraging information about the distribution of confidence scores around an input image, we produce certificates that beat a naive bound based on a direct application of the Neyman-Pearson lemma. The results in this work show that certificates can be strengthened by incorporating more information into the worst-case bound than just the average vote. We hope this line of research leads to methods for strengthening smoothing certificates based on other information sources, such as properties of the base classifier or the spatial distribution of votes.

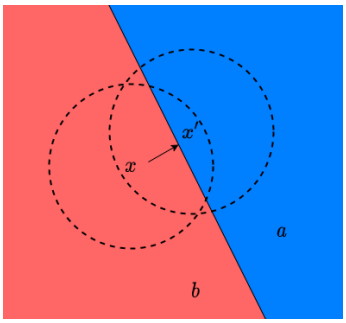


Figure 6.1: Naive classifier

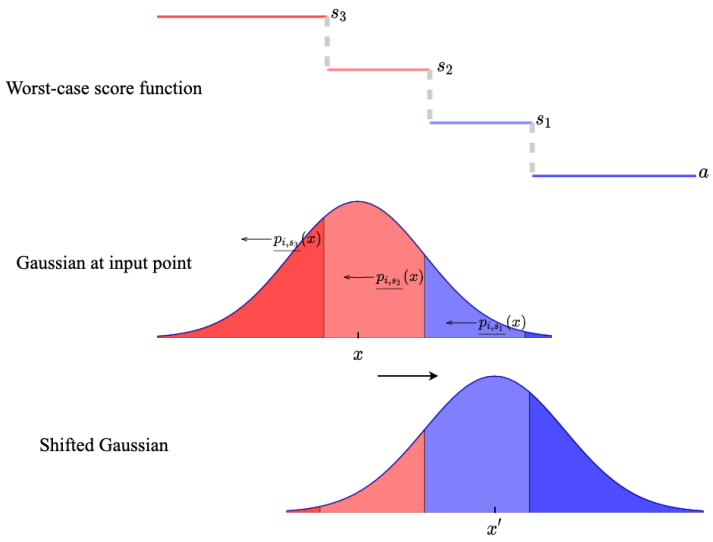


Figure 6.2: CDF-based classifier

Figure 6.3: Worst case classifier behaviour using (a) naive approach and (b) CDF-based method. As the center of the distribution moves from x to x', the probability mass of the higher values of the score function (indicated in red) decreases and that of the lower values (indicated in blue) increases, bringing down the value of the expected score.

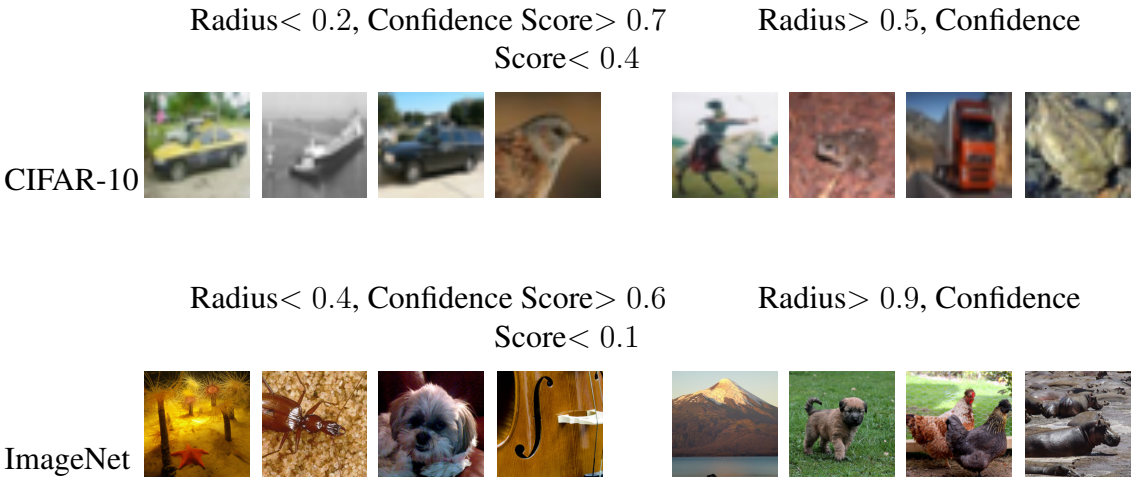


Figure 6.6: Certified radius does not correlate well with human visual confidence or network confidence score. Low radius images on the left have high confidence scores, while the high radius images on the right all have low confidence scores. There is not a pronounced visual difference between low- and high-radius images.

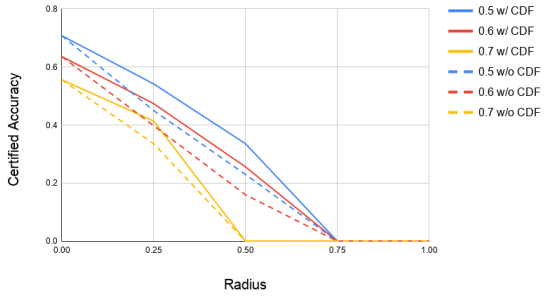


Figure 6.7: Average Prediction Score (CIFAR-10)

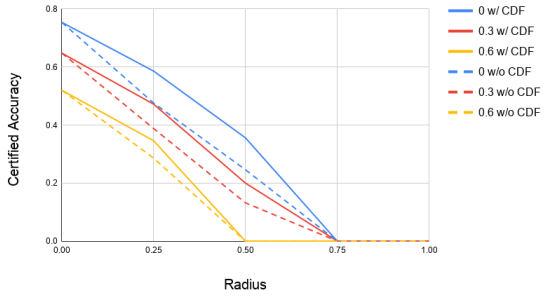


Figure 6.8: Margin (CIFAR-10)

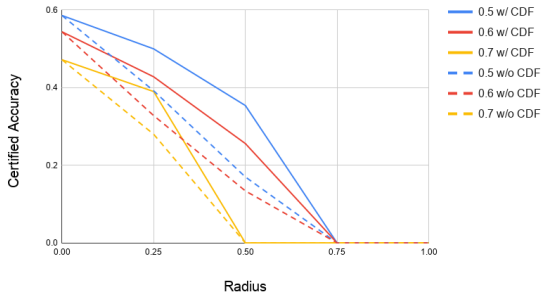


Figure 6.9: Average Prediction Score (ImageNet)

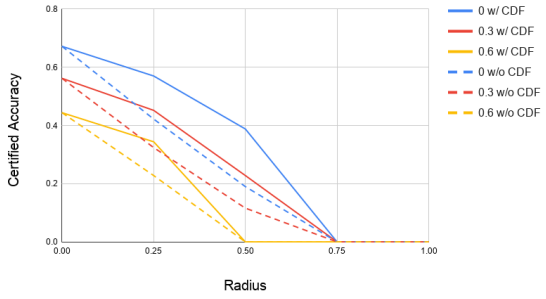


Figure 6.10: Margin (ImageNet)

Figure 6.11: Certified accuracy vs. radius (CIFAR-10 & ImageNet) at different cutoffs for average confidence score with $\sigma = 0.25$. Solid and dashed lines represent certificates computed with and without CDF bound respectively.

6.7 Appendices

A Proof of Theorem 11

We first prove a slightly modified version of the Neyman-Pearson lemma.

Lemma 14 (Neyman & Pearson, 1933)**.** *Let X and Y be random variables in $\mathbb{R}^d$ with densities μ_X and μ_Y. Let $h : \mathbb{R}^d \to (a, b)$ be a function. Then:*

1. *If $S = \left\{ z \in \mathbb{R}^d \mid \frac{\mu_Y(z)}{\mu_X(z)} \leq t \right\}$ for some $t > 0$ and $\mathbb{E}[h(X)] \geq (b-a)\mathbb{P}(X \in S) + a$, then $\mathbb{E}[h(Y)] \geq (b-a)\mathbb{P}(Y \in S) + a$.*

2. *If $S = \left\{ z \in \mathbb{R}^d \mid \frac{\mu_Y(z)}{\mu_X(z)} \geq t \right\}$ for some $t > 0$ and $\mathbb{E}[h(X)] \leq (b-a)\mathbb{P}(X \in S) + a$, then $\mathbb{E}[h(Y)] \leq (b-a)\mathbb{P}(Y \in S) + a$.*

Proof. Let S^c be the complement set of S.

$$
\begin{aligned}
\mathbb{E}[h(Y)] - (b-a)\mathbb{P}(Y \in S) - a &= \mathbb{E}[h(Y)] - b\mathbb{P}(Y \in S) - a(1 - \mathbb{P}(Y \in S)) \\
&= \mathbb{E}[h(Y)] - b\mathbb{P}(Y \in S) - a\mathbb{P}(Y \notin S) \\
&= \int_{\mathbb{R}^d} h(z)\mu_Y(z)dz - b\int_S \mu_Y(z)dz - a\int_{S^c} \mu_Y(z)dz \\
&= \left[\int_{S^c} h(z)\mu_Y(z)dz + \int_S h(z)\mu_Y(z)dz\right] \\
&\qquad - b\int_S \mu_Y(z)dz - a\int_{S^c} \mu_Y(z)dz \\
&= \int_{S^c} (h(z) - a)\mu_Y(z)dz - \int_S (b - h(z))\mu_Y(z)dz \\
&\geq t\left[\int_{S^c} (h(z) - a)\mu_X(z)dz - \int_S (b - h(z))\mu_X(z)dz\right] \\
&\qquad\qquad \text{(since } a < h(z) < b\text{)}
\end{aligned}
$$

$$= t\left[\int_{\mathbb{R}^d} h(z)\mu_X(z)dz - b\int_S \mu_X(z)dz - a\int_{S^c} \mu_X(z)dz\right]$$

$$= t\left[\mathbb{E}[h(X)] - b\mathbb{P}(X \in S) - a\mathbb{P}(X \notin S)\right]$$

$$= t\left[\mathbb{E}[h(X)] - b\mathbb{P}(X \in S) - a(1 - \mathbb{P}(X \in S))\right]$$

$$= t\left[\mathbb{E}[h(X)] - (b-a)\mathbb{P}(X \in S) - a\right] \geq 0$$

The second statement can be proven similarly by switching $\geq$ and $\leq$. $\square$

In the first statement of the lemma, set h to f_i, μ_X to $\mathcal{N}(x, \sigma^2 I)$ and μ_Y to $\mathcal{N}(x', \sigma^2 I)$, and find a t, such that, $\mathbb{P}(X \in S) = \underline{p_i}(x)$. Now, since μ_X and μ_Y are isometric Gaussians with the same variance,

$$\frac{\mu_Y(z)}{\mu_X(z)} \leq t \iff (x' - x)^T z \leq \beta$$

for some $\beta \in \mathbb{R}$. Therefore, the set S is a half-space defined by a hyper-plane orthogonal to the perturbation $x' - x$. So, if $\|x' - x\|_2 \leq R$, then $\mathbb{P}(Y \in S) \geq \Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R)$.

$$\begin{aligned}
\bar{f_i}(x') &= \mathbb{E}[f_i(Y)] \\
&\geq (b-a)\mathbb{P}(Y \in S) + a && \text{(from the above lemma)} \\
&\geq (b-a)\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R) + a \\
&= b\Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R) + a(1 - \Phi_\sigma(\Phi_\sigma^{-1}(\underline{p_i}(x)) - R))
\end{aligned}$$

The upper bound on $\bar{f_i}(x')$ can be derived similarly by applying the second statement of the above lemma.

A.1 Alternate proof

Theorem 11 can also be proved for $\sigma = 1$ using Lemma 2 from Salman et al. in [39]. This lemma states that for any function $g : \mathbb{R}^d \to (0, 1)$, $\Phi^{-1}(\bar{g})$ is 1-Lipschitz, where Φ^{-1} is the inverse CDF of the standard Gaussian distribution. Set $g(.)$ to be $\frac{f_i(.)-a}{b-a}$ for an arbitrary class i. Then, $\bar{g}(x) = \frac{\bar{f}_i(x)-a}{b-a}$ is upper and lower bounded by $\overline{p_i}(x)$ and $\underline{p_i}(x)$ respectively. Due to the Lipschitz condition, we have,

$$\Phi^{-1}(\bar{g}(x)) - \Phi^{-1}(\bar{g}(x')) \leq \|x - x'\|_2 \leq R$$

$$\Phi^{-1}(\bar{g}(x')) \geq \Phi^{-1}(\bar{g}(x')) - R \geq \Phi^{-1}(\underline{p_i}(x)) - R$$

$$\bar{g}(x') \geq \Phi(\Phi^{-1}(\underline{p_i}(x)) - R)$$

Substituting $\bar{g}(x) = \frac{\bar{f}_i(x)-a}{b-a}$ and rearranging terms appropriately gives us the first bound in theorem 11. The second bound can be derived similarly.

B Proof of Lemma 13

Let S^c be the complement set of S.

$$\begin{aligned}
\mathbb{P}(h(Y) \geq s) - \mathbb{P}(Y \in S) &= \int_{\mathbb{R}^d} \mathbf{1}\{h(z) \geq s\}\mu_Y(z)dz - \int_S \mu_Y(z)dz \\
&= \left[\int_{S^c} \mathbf{1}\{h(z) \geq s\}\mu_Y(z)dz + \int_S \mathbf{1}\{h(z) \geq s\}\mu_Y(z)dz\right] - \int_S \mu_Y(z)dz \\
&= \int_{S^c} \mathbf{1}\{h(z) \geq s\}\mu_Y(z)dz - \int_S (1 - \mathbf{1}\{h(z) \geq s\})\mu_Y(z)dz
\end{aligned}$$

$$\geq t\left[\int_{S^c} \mathbf{1}\{h(z) \geq s\}\mu_X(z)dz - \int_S (1 - \mathbf{1}\{h(z) \geq s\})\mu_X(z)dz\right]$$

(since $0 \leq \mathbf{1}\{h(z) \geq s\} \leq 1$)

$$= t\left[\int_{\mathbb{R}^d} \mathbf{1}\{h(z) \geq s\}\mu_X(z)dz - \int_S \mu_X(z)dz\right]$$

$$= t\left[\mathbb{P}(h(X) \geq s) - \mathbb{P}(X \in S)\right] \geq 0$$

The second statement of the lemma can be proven similarly by switching $\geq$ and $\leq$.

C Additional Experiments

In section 6.5.2, we compared the two methods, using Hoeffding's inequality and Dvoretzky–Kiefer–Wolfowitz inequality to derive the required lower bounds, for the certificates. We repeat the same experiments in figure 6.16 for $\sigma = 0.50$. Then, in figure 6.21, we show that the CDF-based method (using the DKW inequality) outperforms the baseline approach regardless of how tight a lower-bound for $e_i(x)$ is used in the baseline certificate (6.1). We replace $\underline{e_i}(x)$ with the empirical estimate of the expectation $\hat{e}_i(x) = \sum_{j=1}^{m} f_i(x + \delta_j)/m$, which is an upper bound on $\underline{e_i}(x)$. And since bound (6.1) is an increasing function of $\underline{e_i}(x)$, any valid lower bound $\underline{e_i}(x)$ on the expectation cannot yield a certified accuracy better than that obtained using $\hat{e}_i(x)$. We compare our certificate with the best-possible baseline certificate for some of Cohen et al. [36]'s ResNet-110 models trained on the CIFAR-10 dataset using the same value of α as in section 6.5.2. The baseline mostly stays below the CDF-based method for both types of confidence measures under the noise levels considered.

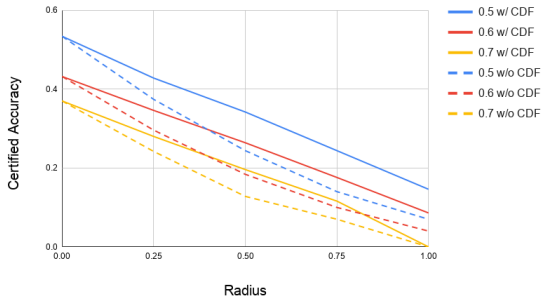


Figure 6.12: Average Prediction Score (CIFAR-10)

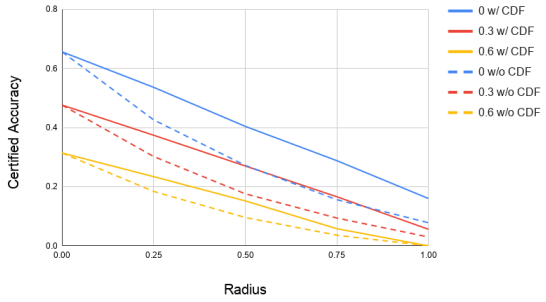


Figure 6.13: Margin (CIFAR-10)

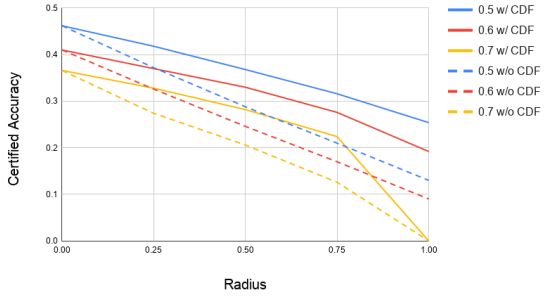


Figure 6.14: Average Prediction Score (ImageNet)

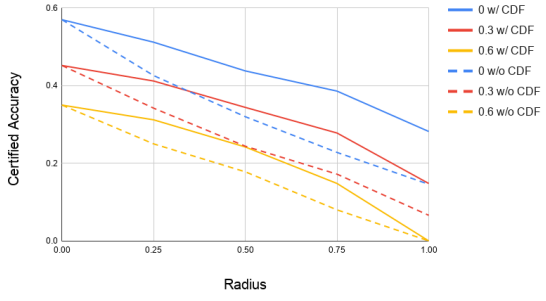


Figure 6.15: Margin (ImageNet)

Figure 6.16: Certified accuracy vs. radius (CIFAR-10 & ImageNet) at different cutoffs for average confidence score with $\sigma = 0.50$. Solid and dashed lines represent certificates computed with and without CDF bound respectively.

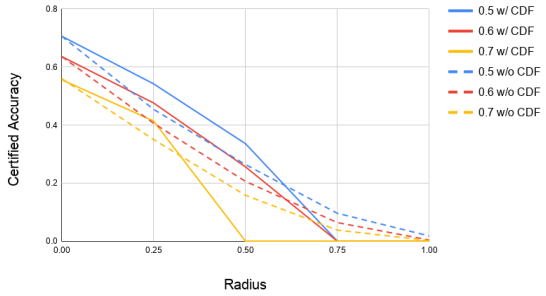


Figure 6.17: Average Prediction Score at $\sigma = 0.25$

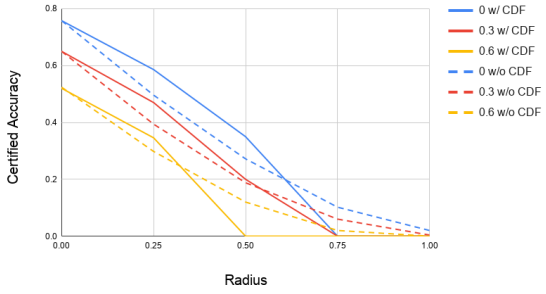


Figure 6.18: Margin at $\sigma = 0.25$

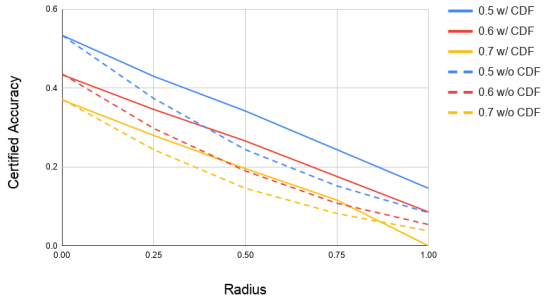


Figure 6.19: Average Prediction Score at $\sigma = 0.50$

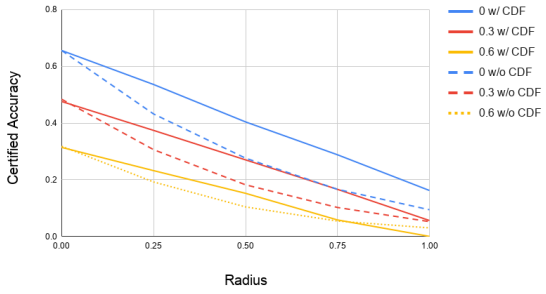


Figure 6.20: Margin at $\sigma = 0.50$

Figure 6.21: Certified accuracy vs. radius (CIFAR-10 only) at different cutoffs for average confidence score. Solid lines represent certificates computed with the CDF bound and dashed lines represent the best-possible baseline certificate.

Chapter 7: Streaming Models with a Sliding Window

7.1 Introduction

Deep neural network (DNN) models are increasingly being adopted for real-time decision-making tasks. They are often required to make predictions on an evolving stream of inputs in applications like algorithmic trading [147, 148, 149, 150, 151], human action recognition [152, 153, 154] and speech detection [155, 156, 157]. However, DNNs are known to malfunction under tiny perturbations of the input, such as an imperceptible noise added to an image, designed to fool them into making incorrect predictions [1, 2, 3, 24, 56]. This vulnerability is not limited just to static models like classifiers and has been demonstrated for streaming models as well [158, 159, 160, 161]. Such input corruptions, commonly known as adversarial attacks, make DNNs especially risky for safety-critical applications such as health monitoring [162, 163, 164, 165] and autonomous driving [97, 166, 167].

Over the years, a long line of research has been dedicated to mitigating this weakness of DNNs. These methods seek to improve the robustness of a model by introducing input corruptions during training [18, 19, 20, 21, 22, 23, 98, 168]. However, such empirical defenses have been shown to break down under newer and stronger attacks [24, 25, 26, 141]. This motivated the study of provable robustness in machine learning (ML)

which seeks to obtain verifiable guarantees on the adversarial performance of a DNN. Several certified robustness techniques have been developed over the years, most notable of which are based on convex relaxation [27, 28, 29, 30, 31], interval-bound propagation [32, 33, 34, 35] and randomized smoothing [36, 37, 38, 39, 67]. However, most of this work focuses on static tasks with independently generated inputs and the adversarial streaming setting still remains open. What makes the streaming setting more challenging is that the adversary can choose the attack budget based on previous inputs. For instance, it could wait for a critical decision-making point, such as a trading algorithm making a buy/sell recommendation or an autonomous vehicle approaching a stop sign, before generating an adversarial perturbation.

In this work, we derive provable robustness guarantees for the streaming setting, where inputs are presented as a sequence of potentially correlated items. We design certificates that produce guarantees on the average model performance over long, potentially infinite, data streams. Our threat model is defined as a man-in-the-middle adversary present between the DNN and the data stream. It can perturb the input items before they are observed by the DNN. The adversary is constrained by a limit on the average size of the perturbation added to the inputs. We show that a DNN that randomizes the inputs before making predictions is guaranteed to achieve a certain performance level for any adversary within this threat model. Unlike existing randomized smoothing-based approaches that aggregate predictions over several noised samples ($\sim 10^6$) of the input, our procedure only requires one sample of the randomized input, keeping the computational complexity of the DNN unchanged. Our certificates are independent of the stream length, making them suitable for large streams.

Technical Challenges: Provable robustness procedures developed for static tasks like classification assume that the inputs are sampled independently from the data distribution. Robustness certificates are derived for individual input instances assuming that the DNN is evaluated on each instance separately. The adversarial perturbation added to one input does not affect the DNN's output for another. However, in the streaming ML setting, the prediction at a given time-step is dependent on past input items in the data stream. A worst-case adversary can exploit this dependence to adapt and strengthen its attack. A robustness certificate derived under the assumption of independence of input samples may not hold for such correlated inputs. Thus, there is a need to design provable robustness techniques tailored specifically for the streaming ML setting.

Out of the existing certified robustness techniques, randomized smoothing has become prominent due to its model-agnostic nature, scalability for high-dimensional problems [37], and flexibility to adapt to different machine learning paradigms like reinforcement learning and structured outputs [49, 70, 71]. This makes randomized smoothing a suitable candidate for provable robustness in streaming ML. However, conventional randomized smoothing approaches require several evaluations ($\sim 10^6$) of the prediction model on different noise vectors in order to produce a robust output. This significantly increases the computational requirements of the model making them infeasible for real-world streaming applications which require decisions to be made in a short time frame such as high-frequency trading and autonomous driving. Our goal is to obtain robustness guarantees for a simple technique that only adds a single noise vector to the DNN's input.

Existing works on provable robustness in reinforcement learning [70, 71] indicate that if the prediction at a given time-step is a function of the entire stream till that step,

the robustness guarantees worsen with the length of the stream and become vacuous for large stream sizes. The tightness analysis of these certificates suggests that it is difficult to achieve robustness guarantees that are independent of the stream size. However, many practical streaming models use only a bounded number of past input items in order to make predictions at a given time step. Recent work has also shown that near-optimal performance can be achieved by only observing a small number of past inputs for several real-world sequential decision-making problems [169]. This raises the natural question:

> Can we obtain better certificates if the DNN only used a fixed number of inputs from the stream?

Our Contributions: We design a robustness certificate for streaming models that use a fixed-sized sliding window over the data stream to make predictions (see Figure 7.1). In our setting, the DNN only uses the part of the data stream inside the window at any given time step. We certify the average performance Z of the model over a stream of size t:

$$Z = \frac{\sum_{i=1}^{t} f_i}{t},$$

where each f_i measures the performance of the DNN at time-step i as a value in the range $[0, 1]$.

The adversary is allowed to perturb the input items inside the window at every time step separately. The strength of the adversary is limited by a bound ϵ on the average size of the perturbation added:

$$\frac{\sum_{i=1}^{t} \sum_{k=1}^{w} d(x_i, x_i^k)}{wt} \leq \epsilon,$$

where x_i and x_i^k are the input item at time-step i and its kth adversarial perturbation respectively, w is the window size and d is a distance function to measure the size of the adversarial perturbations, e.g., $d(x_i, x_i^k) = \|x_i - x_i^k\|_2$. Our adversarial threat model is general enough to subsume the scenario where the attacker only perturbs each stream element only once as a special case where all x_i^ks are set to some x_i'.

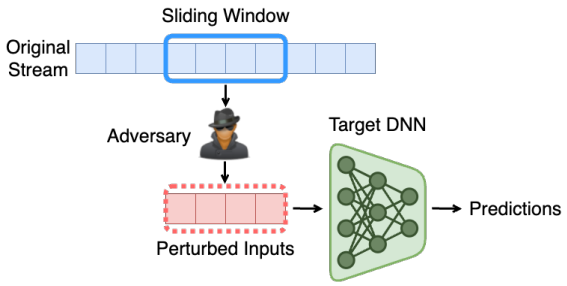


Figure 7.1: Adversarial Streaming Threat Model.

Our main theoretical result shows that the difference between the clean performance $\tilde{Z}$ of a robust streaming model and its performance $\tilde{Z}_\epsilon$ in the presence of an adversarial attack is bounded as follows:

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon), \tag{7.1}$$

where $\psi(.)$ is a concave function that bounds the total variation between the smoothing distributions at two input points as a function of the distance between them (condition (7.4) in Section 7.3). Such an upper bound always exists for any smoothing distribution. For example, when the distance between the points is measured using the ℓ_2-norm and the smoothing distribution is a Gaussian $\mathcal{N}(0, \sigma^2 I)$ with variance σ^2, then the concave upper bound is given by $\psi(\cdot) = \text{erf}(\cdot/2\sqrt{2}\sigma)$. Our robustness certificate is independent of the length of the stream and depends only on the window size w and average perturbation size ϵ. This suggests that streaming ML models with smaller window sizes are provably more robust to adversarial attacks.

We perform experiments on two real-world applications – human activity recognition

and speech keyword detection. We use the UCI HAR datset [170] for human activity recognition and the Speech commands dataset [171] for speech keyword detection. We train convolutional networks that take sliding windows as inputs and provide robustness guarantees for their performance. In our experiments, we consider two different scenarios for the adversary. In the first case, the adversary can perturb an input only once. In the more general second scenario, the adversary can perturb each sliding window separately, making it a powerful attacker. We develop strong adversaries for both of these scenarios and show their effectiveness in our experiments. We then show that our certificates provide meaningful robustness guarantees in the presence of such strong adversaries. Consistent with our theory, our experiments also demonstrate that a smaller window size w gives a stronger certificate.

7.2 Related Work

The adversarial streaming setup has been studied extensively in recent years. Mladenovic et al. [159] designed an attack for transient data streams that do not allow the adversary to re-attack past input items. In their setting, the adversary only has partial knowledge of the target DNN and the perturbations applied in previous time steps are irrevocable. Their objective is to produce an adversarial attack with minimal access to the data stream and the target model. Our goal, on the other hand, is to design a provably robust method that can defend against as general and strong an adversary as possible. We assume that the adversary has full knowledge of the parameters of the target DNN and can change the adversarial perturbations added in previous time steps. Our threat model includes

transient data streams as a special case and applies even to adversaries that only have partial access to the DNN.

Streaming adversarial attacks have also been studied for sampling algorithms such as Bernoulli sampling and reservoir sampling [161]. Here, the goal of the adversary is to create a stream that is unrepresentative of the actual data distribution. Other works have studied the adversarial streaming setup for specific data analysis problems like frequency moment estimation [160], submodular maximization [172], coreset construction and row sampling [158]. In this work, we focus on a robustness certificate for general DNN models in the streaming setting under the conventional notion of adversarial attacks in machine learning literature. We use a sliding-window computational model which has been extensively studied over several years for many streaming applications [173, 174, 175]. Recently Efroni et al. [169] also showed that a short-term memory is sufficient for several real-world reinforcement learning tasks.

A closely related setting is that of adversarial reinforcement learning. Adversarial attacks have been designed that either directly corrupt the observations of the agent [6, 7, 8] or introduce adversarial behavior in a competing agent [5]. Robust training methods, such as adding adversarial noise [104, 105] and training with a learned adversary in an online alternating fashion [106], have been proposed to in improve the robustness of RL agents. Several certified defenses have also been developed over the years. For instance, Zhang et al. [100] developed a method that can certify the actions of an RL agent at each time step under a fixed adversarial perturbation budget. It can certify the total reward obtained at the end of an episode if each of the intermediate actions is certifiably robust. Our streaming formulation allows the adversary to choose the budget at each time step

as long as the average perturbation size remains below ϵ over time. Our framework also does not require each prediction to be robust in order to certify the average performance of the DNN. More recent works in certified RL can produce robustness guarantees on the total reward without requiring every intermediate action to be robust or the adversarial budget to be fixed [70, 71]. However, these certificates degrade for longer streams and the tightness analysis of these certificates indicates that this dependence on stream size may not be improved. Our goal is to keep the robustness guarantees independent of stream size so that they are suitable even for large streams.

The literature on provable robustness has primarily focused on static prediction problems like image classification. One of the most prominent techniques in this line of research is randomized smoothing. For a given input image, this technique aggregates the output of a DNN on several noisy versions of the image to produce a robust class label [36, 37]. This is the first approach that scaled up to high-dimensional image datasets like ImageNet for ℓ_2-norm bounded adversaries.. It does not make any assumptions on the underlying neural network such as Lipschitz continuity or a specific architecture, making it suitable for conventional DNNs that are several layers deep. However, randomized smoothing also suffers some fundamental limitations for higher norms such as the ℓ_∞-norm [50]. Due to its flexible nature, randomized smoothing has also been adapted for tasks beyond classification, such as segmentation and deep generative modeling, with multi-dimensional and structured outputs like images, segmentation masks, and language [49]. For such outputs, robustness certificates are designed in terms of a distance metric in the output space such as LPIPS distance, intersection-over-union and total variation distance. However, provable robustness in the static setting assumes a fixed budget on

the size of the adversarial perturbation for each input instance and does not allow the adversary to choose a different budget for each instance. In our streaming threat model, we allow the adversary the flexibility of allocating the adversarial budget to different time steps in an effective way, attacking more critical input items with a higher budget and conserving its budget at other time steps. Recent work on provable robustness against Wasserstein shifts of the data distribution allows the adversary to choose the attack budget for each instance differently [176]. However, unlike our streaming setting, the input instances are drawn independently from the data distribution and the adversarial perturbation applied to one instance does not impact the performance of the DNN on another.

7.3 Preliminaries and Notation

Streaming ML Setting: We define a data stream of size t as a sequence of input items $x_1, x_2, \ldots, x_i, \ldots, x_t$ generated one-by-one from an input space $\mathcal{X}$ over discrete time steps. At each time step i, a DNN model μ makes a prediction that may depend on no more than w of the previous inputs. We refer to the contiguous block of past input items as a window $W_i \in \mathcal{X}^{\min(i,w)}$ of size w defined as follows:

$$W_i = \begin{cases} (x_1, x_2, \ldots, x_i) & \text{for } i \leq w \\ (x_{i-w+1}, x_{i-w+2}, \ldots, x_i) & \text{otherwise.} \end{cases}$$

The performance of the model μ at time step i is given by a function $f_i : \mathcal{X}^{\min(i,w)} \rightarrow [0, 1]$ that passes the window W_i through the model μ, compares the prediction with the ground truth and outputs a value in the range $[0, 1]$. For instance, in speech recognition,

the window W_i would represent the audio from the past few seconds which gets fed to the model μ. The function $f_i = \mathbf{1}\{\mu(W_i) = y_i\}$ could indicate whether the prediction of μ matches the ground truth y_i. Similarly, in autonomous driving, we can define a performance function $f_i = \text{IoU}(\mu(W_i), y_i)$ that measures the average intersection-over-union of the segmentation mask of the surrounding environment. We define the overall performance Z of the model μ as an average over the t time-steps:

$$Z = \frac{\sum_{i=1}^{t} f_i}{t}.$$

Threat Model: An adversary A is present between the DNN and the data stream which can perturb the inputs with the objective of minimizing the average performance Z of the DNN (see Figure 7.1). Let x_i' be the perturbed input at step i. We define a constraint on the amount by which the adversary can perturb the inputs as a bound on the average distance between the original input items x_i and their perturbed versions x_i':

$$\frac{\sum_{i=1}^{t} d(x_i, x_i')}{t} \leq \epsilon, \tag{7.2}$$

where d is a function that measures the distance between a pair of input items from $\mathcal{X}$, e.g., $d(x_i, x_i') = \|x_i - x_i'\|_2$. The adversary seeks to minimize the overall performance Z of the model without violating the above constraint, i.e.,

$$\min_{A \in \mathcal{A}_\epsilon} \sum_{i=1}^{t} f_i(A(x_i), A(x_{i-1}), \ldots, A(x_{i-w+1}))/t,$$

where $\mathcal{A}_\epsilon$ is the set of all adversaries satisfying constraint (7.2). We also study another

threat model where the adversary is allowed to attack an input item x_i in every window that it appears in. We denote the k-th attack of x_i as x_i^k and redefine the above constraint as follows:

$$\frac{\sum_{i=1}^{t}\sum_{k=1}^{w} d(x_i, x_i^k)}{wt} \leq \epsilon \tag{7.3}$$

This threat model is more general than the one defined by constraint (7.2) because it subsumes this constraint as a special case when all x_i^k are equal to x_i'. Thus, any robustness guarantee that holds for this stronger threat model must also hold for the previous one.

Robustness Procedure: Our goal is to design a procedure that has provable robustness guarantees against the above threat models. We define a robust prediction model $\tilde{\mu}$: Given an input $x_i \in \mathcal{X}$, we sample a point $\tilde{x}_i$ from a probability distribution $\mathcal{S}(x_i)$ around x_i (e.g., $\mathcal{N}(x_i, \sigma^2 I)$) and evaluate the model μ on $\tilde{x}_i$. Define the performance of $\tilde{\mu}$ at time-step i to be the expected value of f_i under the randomized inputs, i.e.,

$$\tilde{f}_i = \mathbb{E}_{\tilde{x}_i \sim \mathcal{S}(x_i)}[f_i(\tilde{x}_i, \tilde{x}_{i-1}, \ldots, \tilde{x}_{i-w+1})]$$

and the overall performance as $\tilde{Z} = \sum_{i=1}^{t} \tilde{f}_i / t$.

Let $\psi(\cdot)$ be a concave function bounding the total variation between the distributions $\mathcal{S}(x_i)$ and $\mathcal{S}(x_i')$ as a function of the distance between them, i.e.,

$$\mathsf{TV}(\mathcal{S}(x_i), \mathcal{S}(x_i')) \leq \psi(d(x_i, x_i')). \tag{7.4}$$

Such a bound always exists regardless of the shape of the smoothing distribution because as the distance between the points x_i and x_i' goes from 0 to ∞, the total variation goes from 0 to 1. A trivial concave bound could be obtained by simply taking the convex hull of the region under the total variation curve (see Figure 7.2). However, to find a closed-form expression for ψ, we need to analyze different smoothing distributions and distance functions separately. If the smoothing distribution is a Gaussian $\mathcal{N}(0, \sigma^2 I)$ with variance σ^2 and the distance is measured using the ℓ_2-norm, as in all of our experiments, then $\psi(\|x_i - x_i'\|_2) = \text{erf}(\|x_i - x_i'\|_2 / 2\sqrt{2}\sigma)$, where erf is the Gauss error function. For a uniform smoothing distribution within an interval of size b in each dimension of x_i and the ℓ_1-distance metric, $\psi(\|x_i - x_i'\|_1) = \|x_i - x_i'\|_1 / b$. See Appendix G for proof.

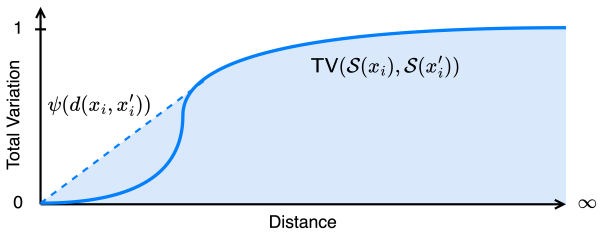


Figure 7.2: Constructing a concave upper bound $\psi(\cdot)$ for any smoothing distribution $\mathcal{S}$.

7.4 Robustness Certificate

In this section, we prove robustness guarantees for the simpler threat model defined by constraint (7.2) where each input item is allowed to be attacked only once. In the following lemma, we bound the change in the performance function $\tilde{f}_i$ at each time-step i using the function ψ and the size of the adversarial perturbation added at each step. For the proof, we first decompose the change in the value of this function into components

for each input item. Since each of these components can be expressed as the difference of the expected value of a function in the range $[0, 1]$ under two probability distributions, they can be bounded by the total variation of these distributions. A complete proof of the following lemma is available in Appendix A.

Lemma 15. *The change in each $\tilde{f}_i$ under an adversary in $\mathcal{A}_\epsilon$ is bounded as*

$$|\tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})| \leq \sum_{j=i}^{i-s+1} \psi(d(x_j, x'_j)),$$

where $s = \min(i, w)$.

Now we use the above lemma to prove the main robustness guarantee. We first decompose the change in the average performance into the average of the differences at each time step. Then we apply lemma 15 to bound each difference with the function ψ of the per-step perturbation size. We then utilize the convex nature of ψ to convert this average over the performance differences to an average of perturbation sizes, which completes the proof.

Theorem 13. *Let $\tilde{Z}_\epsilon$ to be the minimum $\tilde{Z}$ for an adversary in $\mathcal{A}_\epsilon$. Then,*

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon).$$

Proof. Let $\tilde{Z}'$ be the overall performance of $\tilde{M}$ under an adversary. Then,

$$|\tilde{Z} - \tilde{Z}'| = \left| \frac{\sum_{i=1}^{t} \tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1})}{t} - \frac{\sum_{i=1}^{t} \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})}{t} \right|$$

(where $s = \min(i, w)$)

$$
\begin{aligned}
&\leq \frac{1}{t}\sum_{i=1}^{t}\left|\tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})\right| \\
&\leq \sum_{i=1}^{t}\sum_{j=i}^{i-s+1} \psi(d(x_j, x'_j))/t && \text{(from lemma 15)} \\
&\leq w\sum_{i=1}^{t} \psi(d(x_i, x'_i))/t && \text{(since each term appears at most } w \text{ times)} \\
&\leq w\psi\left(\sum_{i=1}^{t} d(x_i, x'_i)/t\right) && (\psi \text{ is concave and Jensen's inequality})
\end{aligned}
$$

Therefore, for the worst-case adversary in $\mathcal{A}_\epsilon$, we have

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon)$$

from constraint (7.2) on the average distance between the original and perturbed inputs.

□

Although the above certificate is designed for the sliding-window computational model for streaming applications, it may also be applied to static tasks like classification with a fixed adversarial budget for all inputs by setting $w = 1$. In Appendix E, we compare our bound with that obtained by Cohen et al. [36] for an ℓ_2-norm bounded adversary and a Gaussian smoothing distribution. While the above bound is not tight, our analysis shows that the gap with static ℓ_2-certificate is small for meaningful robustness guarantees.

7.5 Attacking Each Window

Now, we consider the case where the adversary is allowed to attack each window seen by the target DNN separately. The threat model in this section is defined using constraint (7.3). It is able to re-attack an input item x_i in each new window. Similar to the definition of a window in Section 7.3, define an adversarially corrupted window W'_i as:

$$W'_i = \begin{cases} (x_1^i, x_2^{i-1}, \ldots, x_i^1) & \text{for } i \leq w \\ (x_{i-w+1}^w, x_{i-w+2}^{w-1}, \ldots, x_i^1) & \text{otherwise,} \end{cases}$$

where x_i^k is the k^{th} perturbed instance of x_i.

Similar to the certificate derived in Section 7.4, we first bound the change in the per-step performance function and then use that result to prove the final robustness guarantee. We formulate the following lemma similar to Lemma 15 but accounting for the fact that each input item can be perturbed multiple times.

Lemma 16. *The change in each $\tilde{f}_i$ under an adversary in $\mathcal{A}_\epsilon$ is bounded as*

$$|\tilde{f}_i(W_i) - \tilde{f}_i(W'_i)| \leq \sum_{j=i-s+1}^{i} \psi(d(x_j, x_j^{i+1-j})),$$

where $s = \min(i, w)$.

The proof is available in Appendix B.

We prove the same certified robustness bound as in Section 7.4 but the ϵ here is defined according to constraint (7.3).

Theorem 14. *Let $\tilde{Z}_\epsilon$ to be the minimum $\tilde{Z}$ for an adversary in $\mathcal{A}_\epsilon$. Then,*

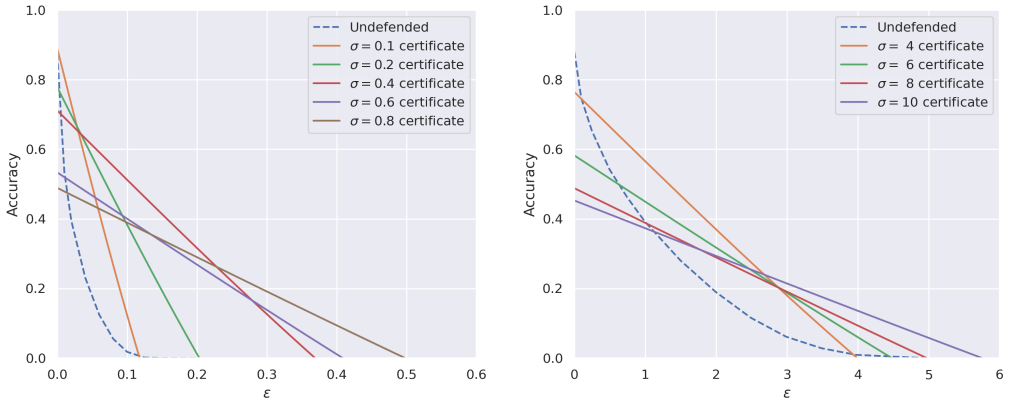
$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon).$$

The proof is available in Appendix C.

7.6 Experiments

We test our certificates for two streaming tasks – speech keyword detection and human activity recognition. We use a subset of the Speech commands dataset [171] for our speech keyword detection task. This subset contains ten keyword classes, corresponding to utterances of numbers from zero to nine recorded at a sample rate of 16 kHz. This dataset also contains noise clips such as audio of running tap water and exercise bike. We add these noise clips to the speech audio to simulate real-world scenarios and stitch them together to generate longer audio clips. We use the UCI HAR dataset [170] for human activity recognition. This contains a 6-D triaxial accelerometer and gyroscope readings measured with human subjects. The objective in HAR is to recognize various human activities based on sensor readings. The UCI HAR dataset contains signals recorded at 50 Hz that correspond to six human activities such as standing, sitting, laying, walking, walking up, and walking down.

We use the M5 network described in [177] with an SGD optimizer and an initial learning rate of 0.1, which we anneal using a cosine scheduler. For the speech detection task, we train a M5 network with 128 channels for 30 epochs with a batch size of 128. For the human activity recognition task, we use a M5 network with 32 channels for 30 epochs



(a) Speech keyword detection

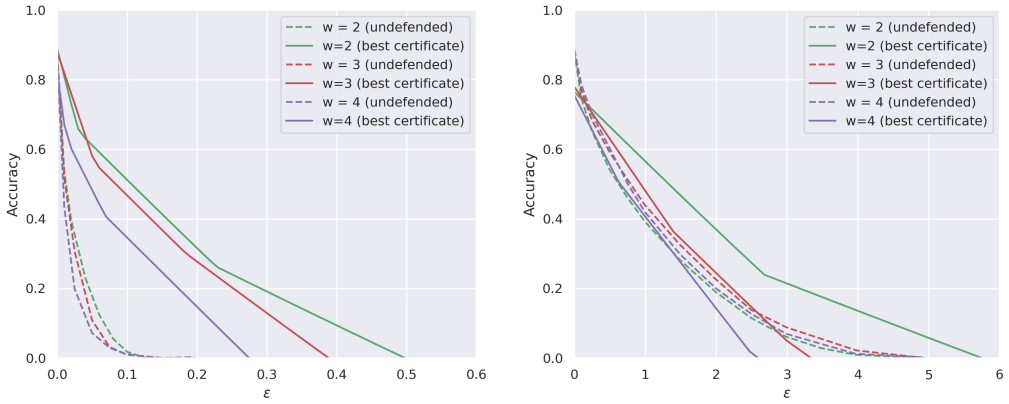
(b) Human activity recognition

Figure 7.3: Certificates (solid lines) against online adversarial attacks for varying smoothing noises. Here the adversary is allowed to perturb each input only once. The dashed lines represent the performance of an undefended model under an adversarial attack.

with a batch size of 256. We apply isotropic Gaussian noise for smoothing and use the ℓ_2-norm to define the average distance measure d. For the speech keyword detection task, we use smoothing noises with standard deviations of 0.1, 0.2, 0.4, 0.6, and 0.8. For the human activity recognition task, we use smoothing noises with standard deviations of 4, 6, 8, and 10. See Appendix F for more details on the experiments. We compute certificates for both scenarios, where the input is attacked only once and where each window can be attacked with the ability to re-attack inputs. These experiments show that our certificates provide meaningful guarantees against adversarial perturbations.

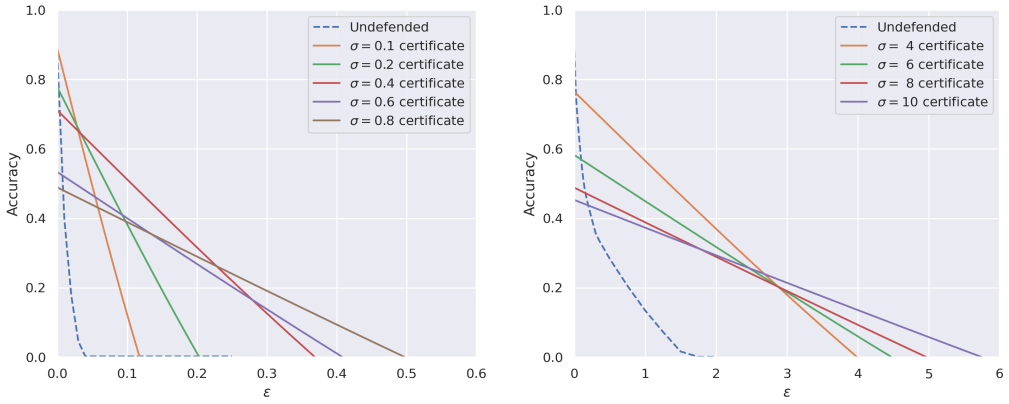
7.6.1 Attacking each input only once

We evaluate the robustness of undefended models using a custom-made attack that is constrained by the ℓ_2-norm budget, as described in equation 7.2. To adhere to this constraint at each time-step j, the attacker must only perturb the input x_j, since the previous inputs $(x_{j-w+1}, ..., x_{j-1})$ have already been perturbed. This creates a significant



(a) Speech keyword detection (b) Human activity recognition

Figure 7.4: Best certificates across varying smoothing noises for different window sizes. Streaming models with smaller window sizes are more robust to adversarial perturbations.



(a) Speech keyword detection (b) Human activity recognition

Figure 7.5: Certificates against online adversarial attacks for varying smoothing noises. Here we attack each window, allowing input items to be attacked multiple times The average size of perturbation is computed as per equation 7.3.

challenge in creating a strong adversary. We design an adversary that only perturbs the last input x_j at every time-step j using projected gradient descent to minimize f_j. In our experiments, we set $f_j = 1$ if the model outputs the correct class and $f_j = 0$ when the model misclassifies. We linearly search using grid search parameter α for the smallest distance $d(x_j, x'_j)$ such that the input $(x'_{j-w+1}, ..., x'_j)$ leads to a misclassification at time-step j. We perturb x_j if $(x'_{j-w+1}, ..., x'_j)$ leads to misclassification and the average distance budget at time-step j is less than ϵ. Else, we do not perturb x_j. In this manner, our attack

Algorithm 8: Our streaming attack

Input: time-step j, clean inputs $x_j, x_{j-1}, ..., x_{j-w+1}$, perturbed inputs $x'_{j-1}, ..., x'_{j-w+1}$, attack budget ϵ, search parameter $\alpha \in \mathbb{N}$.

$d_{j-1} = \sum_{i=1}^{j-1} d(x_i, x'_i)$

$budget_j = j\epsilon - d_{j-1}$

for $i = 0$ **to** α **do**

  $\epsilon' = \frac{i}{\alpha} \cdot budget_j$

  $x = \arg\min_x f_j(x, ..., x'_{j-w+1})$ s.t. $d(x, x_j) \leq \epsilon'$

  if $f_j(x'_j, ..., x'_{j-w+1}) = 0$ **then**

    $x'_j = x$

    break

  else

    $x'_j = x_j$

  end if

end for

perturbs the streaming input in a greedy fashion. See Algorithm 8 for details.

We conduct our streaming attack on the keyword recognition task with a window size of $w = 2$, where each input x_j is a 4000-dimensional vector in the range [0,1]. We also perform the attack on the human activity recognition task with $w = 2$, where each input x_j is a 250x6-dimensional matrix. We use search parameter $\alpha = 15$. We plot the results of our certificates for various smoothing noises (see Figure 7.3). Note that the attack budget ϵ is calculated as per the definition in equation 7.2. In Figure 7.4, we also plot our best certificates across various smoothing noises for different window sizes w. This plot supports our theory that streaming models with smaller window sizes are more robust to adversarial perturbations. Figures 7.7 and 7.8 in Appendix G show that the empirical performance of smooth models after the online adversarial attack is lower bound by our certificates.

7.6.2 Attacking each window

Now, we perform experiments for the attack setting described in Section 7.5. Note that here we need to calculate the attack budget ϵ based on equation 7.3. In this setting, we can re-attack an input for every window, making it a stronger attack. To attack the undefended models, we search for window perturbations that lead to misclassification using a minimum distance budget. Similar to our previous attack in Section 7.6.1, we only perturb a window at time-step j if the average window distance at time-step j is less than ϵ. Also, we do not perturb a window if the window can not be perturbed to reduce the performance f_j. In Figure 7.5, we plot our certificates for this attack setting along with the accuracy of the undefended model for different attack budgets. These experiments show that our certificates produce meaningful performance guarantees against adversarial perturbations even if an attacker has the ability to re-attack the inputs. Figure 7.9 in Appendix G shows that the empirical performance of smooth models after the online adversarial attack is lower bound by our certificates.

7.7 Conclusion

In this work, we design provable robustness guarantees for streaming ML models with a sliding window. Our certificates provide a lower bound on the average performance of a streaming DNN model in the presence of an adversary. The adversarial budget in our threat model is defined in terms of the average size of the perturbations added to the input items across the entire stream. This allows the adversary to allocate a different budget to each input item and leads to a more general threat model than the static setting. Our

certificates are independent of the stream length and can handle long, potentially infinite, streams. They are also applicable for adversaries that are allowed to re-attack past inputs leading to strong robustness guarantees covering a wide range of attack strategies.

To the best of our knowledge, this is the first attempt at designing adversarial robustness certificates for the streaming setting. We note that our robustness guarantees are not proven to be tight and could be improved upon by future work. We hope our work inspires further investigations into provable robustness for streaming ML models.

7.8 Appendices

A Proof of Lemma 15

Statement: *The change in each $\tilde{f}_i$ under an adversary in $\mathcal{A}_\epsilon$ is bounded as*

$$|\tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})| \leq \sum_{j=i}^{i-s+1} \psi(d(x_j, x'_j)),$$

where $s = \min(i, w)$.

Proof. The left-hand side of the above inequality can be re-written as:

$$\begin{aligned}
&|\tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})| \\
&\qquad = |\tilde{f}_i(x_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x_{i-1}, \ldots, x_{i-s+1}) \\
&\qquad\quad + \tilde{f}_i(x'_i, x_{i-1}, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, x'_{i-1}, \ldots, x'_{i-s+1})| \\
&\qquad = \left| \sum_{j=i}^{i-s+1} \tilde{f}_i(x'_i, \ldots x_j, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, \ldots, x'_j, \ldots, x_{i-s+1}) \right| \\
&\qquad \leq \sum_{j=i}^{i-s+1} \left| \tilde{f}_i(x'_i, \ldots x_j, \ldots, x_{i-s+1}) - \tilde{f}_i(x'_i, \ldots, x'_j, \ldots, x_{i-s+1}) \right|
\end{aligned}$$

The two terms in each summand differ only in the jth input. Thus, the jth term in the above summation can be written as the difference of the expected value of some $[0, 1]$-function q_j under the distributions $\mathcal{S}(x_j)$ and $\mathcal{S}(x'_j)$, i.e., $|\mathbb{E}_{\tilde{\chi} \sim \mathcal{S}(x_j)}[q_j(\tilde{\chi})] - \mathbb{E}_{\tilde{\chi} \sim \mathcal{S}(x'_j)}[q_j(\tilde{\chi})]|$, which can be upper bounded by the total variation between $\mathcal{S}(x_j)$ and $\mathcal{S}(x'_j)$. Here, q_j is given by:

$$q_j(\chi) = \mathbb{E}[f_i(\tilde{x}'_i, \ldots, \tilde{x}'_{j-1}, \chi, \tilde{x}_{j+1} \ldots, \tilde{x}_{i-s+1})],$$

where $\chi \in \mathcal{X}$ is the jth input item, the inputs before χ are drawn from the corresponding adversarially shifted smoothing distributions and the inputs after χ are drawn from the original distributions, i.e., $\tilde{x}'_i \sim \mathcal{S}(x'_i), \ldots, \tilde{x}'_{j-1} \sim \mathcal{S}(x'_{j-1})$ and $\tilde{x}_{j+1} \sim \mathcal{S}(x_{j+1}), \ldots, \tilde{x}_{i-s+1} \sim \mathcal{S}(x_{i-s+1})$.

Without loss of generality, assume $\mathbb{E}_{\tilde{\chi}\sim\mathcal{S}(x_j)}[q_j(\tilde{\chi})] \geq \mathbb{E}_{\tilde{\chi}\sim\mathcal{S}(x'_j)}[q_j(\tilde{\chi})]$. Then,

$$
\begin{aligned}
&\left|\mathbb{E}_{\tilde{\chi}\sim\mathcal{S}(x_j)}[q_j(\tilde{\chi})] - \mathbb{E}_{\tilde{\chi}\sim\mathcal{S}(x'_j)}[q_j(\tilde{\chi})]\right| \\
&= \int_{\mathcal{X}} q_j(x)\mu_1(x)dx - \int_{\mathcal{X}} q_j(x)\mu_2(x)dx \quad (\mu_1 \text{ and } \mu_2 \text{ are the PDFs of } \mathcal{S}(x_j) \text{ and } \mathcal{S}(x'_j)) \\
&= \int_{\mathcal{X}} q_j(x)(\mu_1(x) - \mu_2(x))dx \\
&= \int_{\mu_1>\mu_2} q_j(x)(\mu_1(x) - \mu_2(x))dx - \int_{\mu_2>\mu_1} q_j(x)(\mu_2(x) - \mu_1(x))dx \\
&\leq \int_{\mu_1>\mu_2} \max_{x'\in\mathcal{X}} q_j(x')(\mu_1(x) - \mu_2(x))dx - \int_{\mu_2>\mu_1} \min_{x'\in\mathcal{X}} q_j(x')(\mu_2(x) - \mu_1(x))dx \\
&\leq \int_{\mu_1>\mu_2} (\mu_1(x) - \mu_2(x))dz \qquad (\text{since } \max_{x'\in\mathcal{X}} q_j(x') \leq 1 \text{ and } \min_{x'\in\mathcal{X}} q_j(x') \geq 0) \\
&= \frac{1}{2}\int_{\mathcal{X}} |\mu_1(x) - \mu_2(x)|dx = \mathsf{TV}(\mathcal{S}(x_1), \mathcal{S}(x_2)).
\end{aligned}
$$

The equality in the last line follows from the fact that $\int_{\mu_1>\mu_2}(\mu_1(x) - \mu_2(x))dx = \int_{\mu_2>\mu_1}(\mu_2(x) - \mu_1(x))dx = \frac{1}{2}\int_{\mathcal{X}} |\mu_1(x) - \mu_2(x)|dx$.

Therefore, from condition (7.4), we have:

$$
|\tilde{f}_i(x'_i, \ldots x_j, \ldots, x_{i-w+1}) - \tilde{f}_i(x'_i, \ldots, x'_j, \ldots, x_{i-w+1})| \leq \mathsf{TV}(\mathcal{S}(x_j), \mathcal{S}(x'_j)) \leq \psi(d(x_j, x'_j)).
$$

This proves the statement of the lemma. □

B Proof of Lemma 16

Statement: *The change in each $\tilde{f}_j$ under an adversary in $\mathcal{A}_\epsilon$ is bounded as*

$$|\tilde{f}_j(W_j) - \tilde{f}_j(W'_j)| \leq \sum_{i=j-w+1}^{j} \psi(d(x_i, x_i^{j+1-i})).$$

Proof. The left-hand side of the above inequality can be re-written as:

$$\begin{aligned}
|\tilde{f}_j(W_j) - \tilde{f}_j(W'_j)| &= |\tilde{f}_j(x_{j-w+1}, \ldots, x_j) - \tilde{f}_j(x^w_{j-w+1}, \ldots, x^1_j)| \\
&= |\tilde{f}_j(x_{j-w+1}, \ldots, x_{j-1}, x_j) - \tilde{f}_j(x_{j-w+1}, \ldots, x_{j-1}, x^1_j) \\
&+ \tilde{f}_j(x_{j-w+1}, \ldots, x_{j-1}, x^1_j) - \tilde{f}_j(x^w_{j-w+1}, \ldots, x^2_{j-1}, x^1_j)| \\
&= \left| \sum_{k=1}^{w} \tilde{f}_j(x_{j-w+1}, \ldots x_{j-k+1}, x^{k-1}_{j-k+2}, \ldots, x^1_j) \right. \\
&\qquad \left. - \tilde{f}_j(x_{j-w+1}, \ldots x^k_{j-k+1}, x^{k-1}_{j-k+2}, \ldots, x^1_j) \right| \\
&\leq \sum_{k=1}^{w} \left| \tilde{f}_j(x_{j-w+1}, \ldots x_{j-k+1}, x^{k-1}_{j-k+2}, \ldots, x^1_j) \right. \\
&\qquad \left. - \tilde{f}_j(x_{j-w+1}, \ldots x^k_{j-k+1}, x^{k-1}_{j-k+2}, \ldots, x^1_j) \right|
\end{aligned}$$

The two terms in each summand differ only in the $(j-k+1)$-th input. Thus, it can be written as the difference of the expected value of some $[0,1]$-function q under the distributions $\mathcal{S}(x_{j-k+1})$ and $\mathcal{S}(x^k_{j-k+1})$, i.e., $|\mathbb{E}_{\tilde{x}_{j-k+1} \sim \mathcal{S}(x_{j-k+1})}[q(\tilde{x}_{j-k+1})] - \mathbb{E}_{\tilde{x}^k_{j-k+1} \sim \mathcal{S}(x^k_{j-k+1})}[q(\tilde{x}^k_{j-k+1})]|$ which can be upper bounded by the total variation between $\mathcal{S}(x_{j-k+1})$ and $\mathcal{S}(x^k_{j-k+1})$.

Therefore, from condition (7.4), we have:

$$|\tilde{f}_j(x_{j-w+1},\ldots,x_{j-k+1},x_{j-k+2}^{k-1},\ldots,x_j^1) - \tilde{f}_j(x_{j-w+1},\ldots x_{j-k+1}^k,x_{j-k+2}^{k-1},\ldots,x_j^1)|$$

$$\leq \mathsf{TV}(\mathcal{S}(x_{j-k+1}),\mathcal{S}(x_{j-k+1}^k)) \leq \psi(d(x_{j-k+1},x_{j-k+1}^k)).$$

This proves the statement of the lemma. □

C Proof of Theorem 14

Statement: *Let $\tilde{Z}_\epsilon$ to be the minimum $\tilde{Z}$ for an adversary in $\mathcal{A}_\epsilon$. Then,*

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon).$$

Proof. Let $\tilde{Z}'$ be the overall performance of $\tilde{M}$ under an adversary. Then,

$$\begin{aligned}
|\tilde{Z} - \tilde{Z}'| &= \left| \frac{\sum_{j=1}^t \tilde{f}_j(W_j)}{t} - \frac{\sum_{j=1}^t \tilde{f}_j(W_j')}{t} \right| \\
&\leq \frac{\sum_{j=1}^t |\tilde{f}_j(W_j)) - \tilde{f}_j(W_j')|}{t} \\
&\leq \sum_{j=1}^t \sum_{k=1}^w \psi(d(x_{j-k+1},x_{j-k+1}^k))/t && \text{(from lemma 16)} \\
&\leq \sum_{j=1}^t \sum_{k=1}^w \psi(d(x_j,x_j^k))/t \\
&= w\sum_{j=1}^t \sum_{k=1}^w \psi(d(x_j,x_j^k))/wt \\
&\leq w\psi\left(\sum_{j=1}^t \sum_{k=1}^w d(x_j,x_j^k))/wt\right) && (\psi \text{ is concave and Jensen's inequality})
\end{aligned}$$

Therefore, for the worst-case adversary in $\mathcal{A}_\epsilon$, we have

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq w\psi(\epsilon)$$

from constraint (7.2) on the average distance between the original and perturbed inputs.

□

D Function ψ for Different Smoothing Distributions

For an isometric Gaussian distribution,

$$\mathsf{TV}(\mathcal{N}(x_i, \sigma^2 I), \mathcal{N}(x_i', \sigma^2 I)) = \operatorname{erf}(\|x_i - x_i'\|_2 / 2\sqrt{2}\sigma).$$

Proof. Due to the isometric symmetry of the Gaussian distribution and the ℓ_2-norm, the total variation between the two distributions is the same as when they are separated by the same ℓ_2-distance but only in the first coordinate. It is equivalent to shifting a univariate normal distribution by the same amount. Therefore, the total variation between the two distributions is equal to the difference in the probability of a normal random variable with variance σ^2 being less than $\|x_i - x_i'\|_2/2$ and $-\|x_i - x_i'\|_2/2$, i.e., $\Phi(\|x_i - x_i'\|_2/2\sigma) - \Phi(-\|x_i - x_i'\|_2/2\sigma)$ where Φ is the standard normal CDF.

$$\begin{aligned}
\mathsf{TV}(\mathcal{N}(x_i, \sigma^2 I), \mathcal{N}(x_i', \sigma^2 I)) &= \Phi(\|x_i - x_i'\|_2/2\sigma) - \Phi(-\|x_i - x_i'\|_2/2\sigma) \\
&= 2\Phi(\|x_i - x_i'\|_2/2\sigma) - 1 \\
&= 2\left(\frac{1 + \operatorname{erf}(\|x_i - x_i'\|_2/2\sqrt{2}\sigma)}{2}\right) - 1
\end{aligned}$$

$$= \text{erf}(\|x_i - x_i'\|_2 / 2\sqrt{2}\sigma).$$

□

For a uniform smoothing distribution $\mathcal{U}(x_i, b)$ between $x_{ij} - b/2$ and $x_{ij} + b/2$ in each dimension j of x_i for some $b \geq 0$, $\mathsf{TV}(\mathcal{U}(x_i, b), \mathcal{U}(x_i', b)) \leq \|x_i - x_i'\|_1 / b$. When $\|x_i - x_i'\|_1$ is constrained, the overlap between $\mathcal{U}(x_i, b)$ and $\mathcal{U}(x_i', b)$ is minimized when the shift is only along one dimension.

E Comparison with Existing Certificates for Static Tasks

In this section, we compare our bound when applied to the static setting of classification, i.e., window size $w = 1$ in bound (7.1), to that obtained by Cohen et al. [36] for an ℓ_2 adversary and a Gaussian smoothing distribution. As discussed in Appendix D, the ψ function for this case takes the form of the Gauss error function erf. Thus our bound on the drop in the smoothed model's performance against an ℓ_2 adversary is given by:

$$|\tilde{Z} - \tilde{Z}_\epsilon| \leq \text{erf}(\epsilon / 2\sqrt{2}\sigma).$$

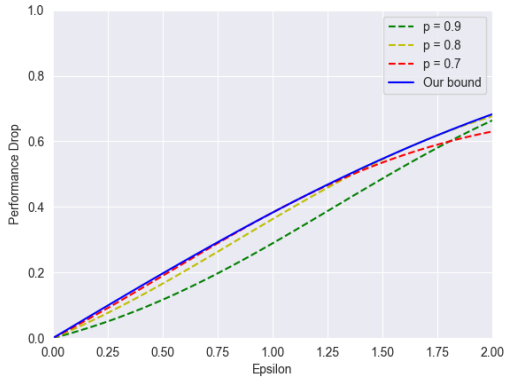


Figure 7.6: Comparison between our bound and [36]'s certificate for an ℓ_2 adversary and a Gaussian smoothing distribution. The solid blue curve corresponds to our bound and the dashed curves represent bound (7.5) for different values of p. We keep $\sigma = 1$ as it only has a scaling effect along the x-axis.

Cohen et al. [36]'s certificate bounds the worst-case adversarial performance as a function of the clean performance. If the probability of predicting the correct class is p on the original input, the probability of that in the presence of an adversary is bounded by $\Phi(\Phi^{-1}(p) - \epsilon/\sigma)$. Therefore, the performance drop Δp is bounded by:

$$\Delta p \leq p - \Phi\left(\Phi^{-1}(p) - \frac{\epsilon}{\sigma}\right). \tag{7.5}$$

Figure 7.6 compares the two bounds for different values of p. We keep $\sigma = 1$ as it only has a scaling effect along the x-axis. The bound from the ℓ_2 certificate by Cohen et al. [36] is tighter than ours, mainly because it takes the clean performance p of the smoothed model into account. However, the gap between the two bounds is small in the range where ϵ goes from 0 to 2, by which point the certified performance drops by more than 60%. Thus for most meaningful robustness guarantees, our certificates are almost at par with the best-known ℓ_2 certificates. The key advantage of our certificates over those for the static setting is that they are applicable for an adaptive adversary that can allocate different attack budgets for different input items in the stream.

F Experimental details

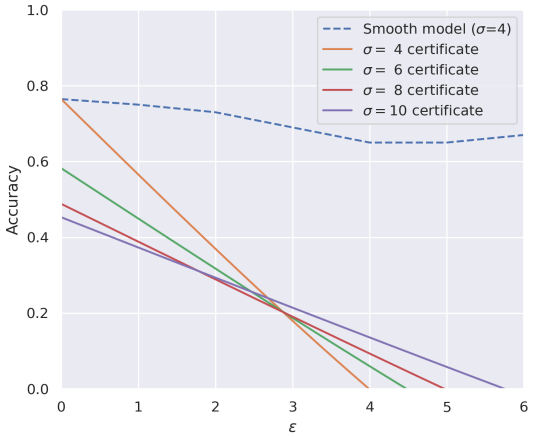
We use a single NVIDIA RTX A4000 GPU with four AMD EPYC 7302P Processors. For our main experiments with UCI HAR and Speech Commands datasets, we use window size $w = 2$ with inputs belonging to $\mathbb{R}^{250\times 6}$ and $\mathbb{R}^{4000}$. The UCI HAR dataset consists of long streaming inputs with sample-level annotations. For a window W_j, the label is the majority class that is present in that window. The signals in the HAR dataset are

standardized to have mean 0 and variance 1. For the speech keyword detection task, we use a subset of the Speech commands dataset that consists of long noise clips and one-second-long speech keyword clips. The labels for each audio clip are available. We utilize all the long noise clips and clips belonging to the classes belonging speech utterances of numbers from zero to nine to make longer clips for our streaming case. We add noise clips to the keyword audios to make them more similar to real-world scenarios. Each clip is stitched together [178] with arbitrarily long noise between each keyword clip. To make transitions between the audio smooth, we use exponential decays to overlap keyworrd audio clips for stitching, with noise in the background. Hence, for the speech keyword detection, we have 11 classes for labels – zero to nine and a noise class. A window is labeled to be the majority class in that window.

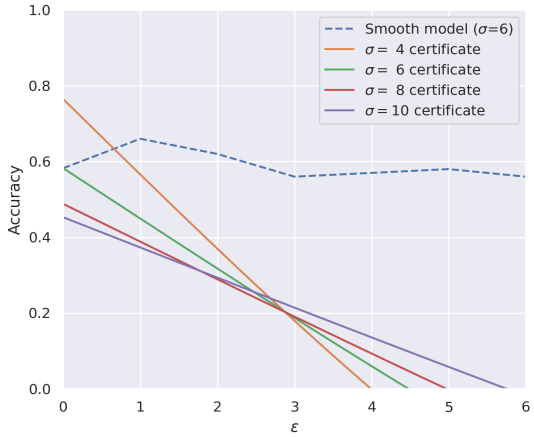
For training, we use M5 networks with 32 channels for HAR. We train for 30 epochs with a bath-size of 256 using SGD with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. We use a cosine annealing learning rate scheduler. For training the robust models, we use different smoothing noises with standard deviations 4, 6, 8, and 10. For training on the keyword detection data, we use M5 networks with 128 channels for HAR. We train for 30 epochs with a bath-size of 128 using SGD with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. We use a cosine annealing learning rate scheduler. For training the robust models, we use different smoothing noises with standard deviations 0.1, 0.2, 0.4, 0.6, and 0.8. For attacking the trained models, we use PGD ℓ_2 attacks for both the datasets. PGD is run for 100 steps with a step size of $2\epsilon'/100$ where ϵ' is the ℓ_2 attack budget.

G Attacking the Smooth Models

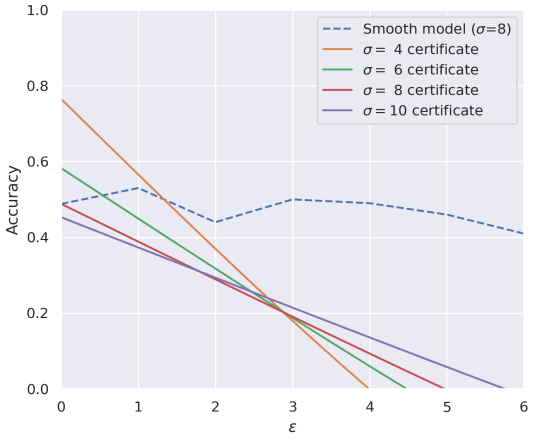
In this section, we empirically validate our certificates by showing that the performance of the smoothed models in the presence of an adversary is lower-bounded by our certificates. For the first set of experiments (Figures 7.7 and 7.8), we consider an adversary that is allowed to attack an input item only once, as in Section 7.6.1. We show our results on the Human Activity Recognition dataset in Figure 7.7 and the keyword detection task in Figure 7.8 for a window size of $w = 2$. In Figure 7.9, we show our results on the HAR dataset where the adversary can attack each window separately as per equation 7.3. As seen in the plots, the empirical performance of the smooth models after the online adversarial attacks is always better than the performance guaranteed by our certificates. By comparing Figures 7.7 and 7.9, we observe that allowing the adversary to attack each window separately makes it significantly stronger and brings the adversarial performance of the smoothed model closer to the certified performance.



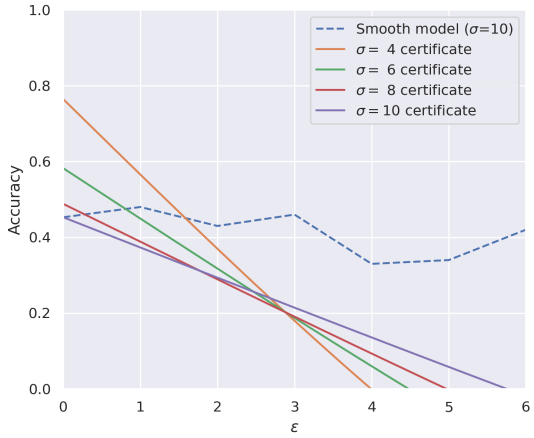
(a) Attacking model with smoothing noise $\sigma = 4$



(b) Attacking model with smoothing noise $\sigma = 6$

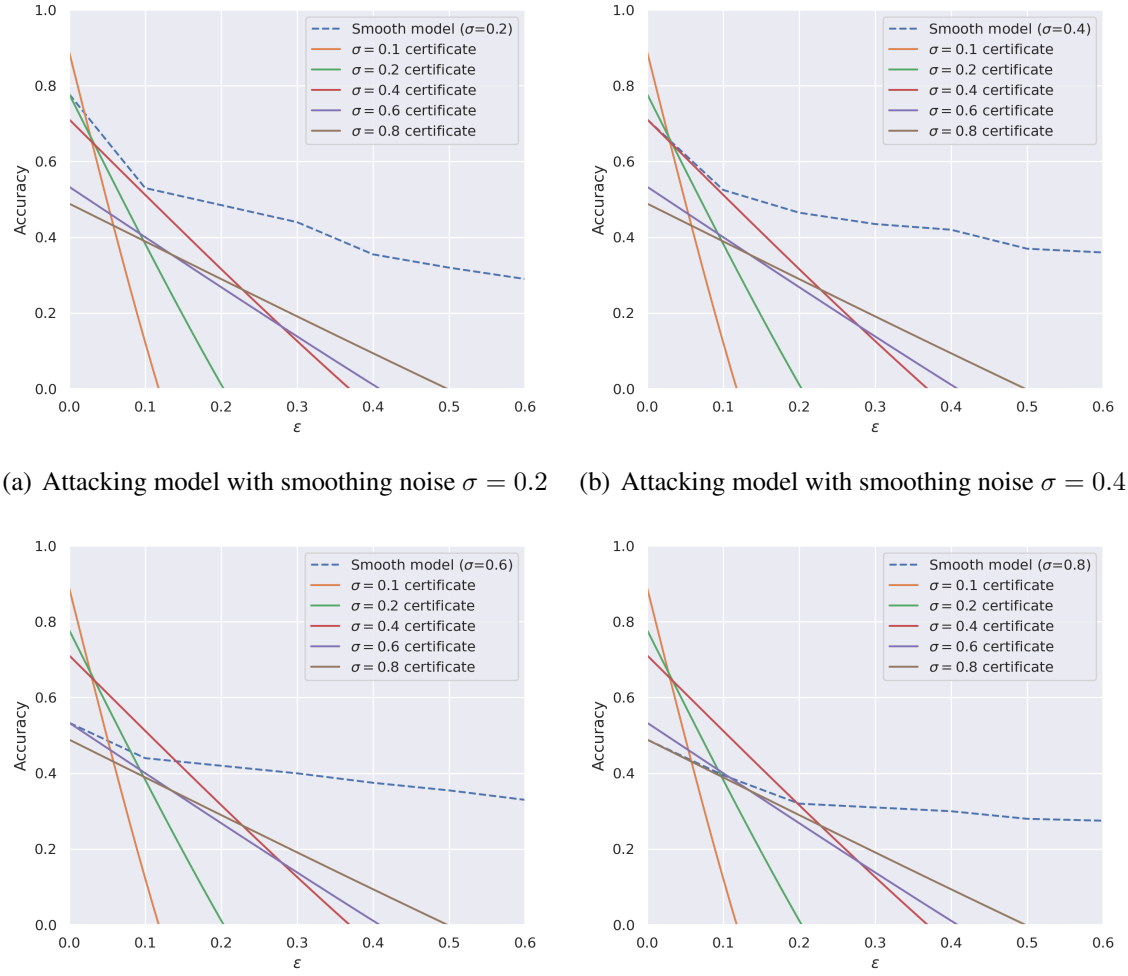


(c) Attacking model with smoothing noise $\sigma = 8$



(d) Attacking model with smoothing noise $\sigma = 10$

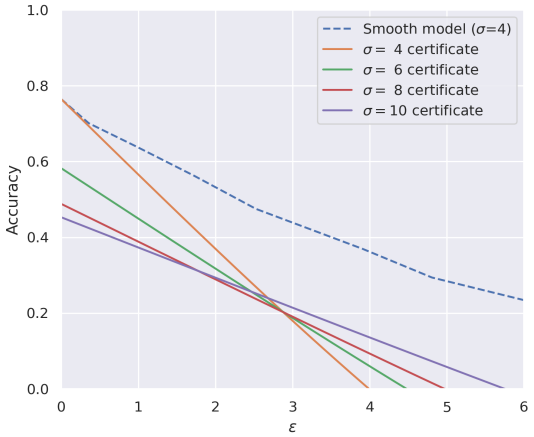
Figure 7.7: Certificates against online adversarial attacks for varying smoothing noises for the human activity recognition task. We attack smooth models trained with different smoothing noises in these plots. Here we can perturb each input only once. The average size of perturbation is computed as per equation 7.2.



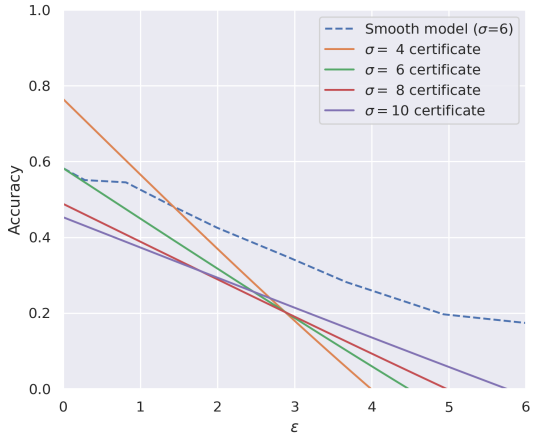
(a) Attacking model with smoothing noise $\sigma = 0.2$ (b) Attacking model with smoothing noise $\sigma = 0.4$

(c) Attacking model with smoothing noise $\sigma = 0.6$ (d) Attacking model with smoothing noise $\sigma = 0.8$

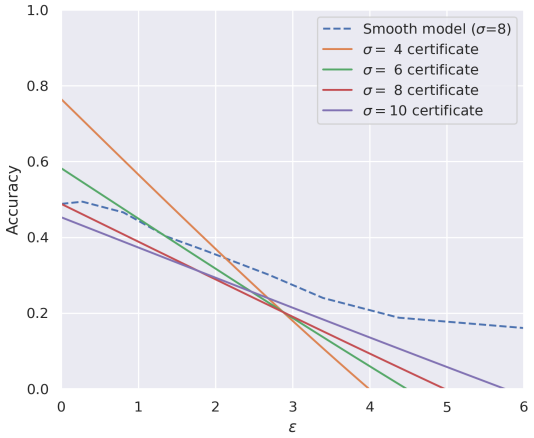
Figure 7.8: Certificates against online adversarial attacks for varying smoothing noises for the speech keyword detection task. We attack smooth models trained with different smoothing noises in these plots. Here we can perturb each input only once. The average size of perturbation is computed as per equation 7.2.



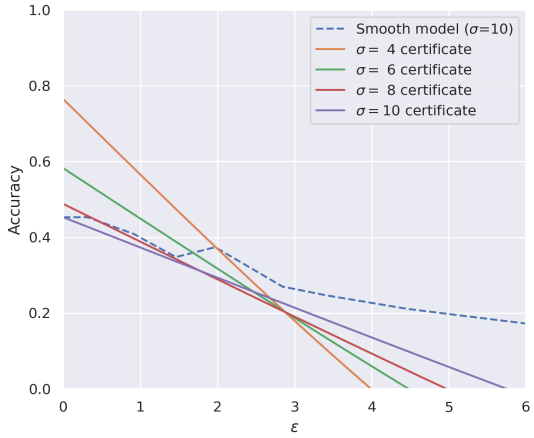
(a) Attacking model with smoothing noise $\sigma = 4$



(b) Attacking model with smoothing noise $\sigma = 6$



(c) Attacking model with smoothing noise $\sigma = 8$



(d) Attacking model with smoothing noise $\sigma = 10$

Figure 7.9: Certificates against online adversarial attacks for varying smoothing noises for the human activity recognition task. We attack smooth models trained with different smoothing noises in these plots. Here we can attack each window separately. The average size of perturbation is computed as per equation 7.3.

Chapter 8: Conclusion

8.1 Contributions

We study several approaches for extending provable robustness to real-world scenarios. The literature on provable robustness in machine learning mainly focuses on static tasks such as image classification. Certificates are designed with a fixed adversarial budget for each input instance and with the assumption that inputs are sampled independently. In this work, we develop certifiable methods that can defend against dynamic and adaptive adversaries as in reinforcement learning and streaming tasks. We also design robustness certificates for tasks with complex outputs such as images, language, segmentation masks, etc., and for distribution shifts caused by natural perturbations like changes in the color balance of an image. We also study the limitations of extending randomized smoothing-based approaches for the ℓ_∞-threat model for high-dimensional inputs.

Our first contribution is a robustness certificate for the accuracy of a model under bounded Wasserstein shifts of the data distribution. We show that a simple procedure that randomizes the input of the model within a transformation space is provably robust to distributional shifts under the transformation. Our framework allows the datum-specific perturbation size to vary across different points in the input distribution and is general enough to include fixed-sized perturbations as well. Our certificates produce guaranteed

lower bounds on the performance of the model for any (natural or adversarial) shift of the input distribution within a Wasserstein ball around the original distribution.

In our second work, we present robustness guarantees in the reinforcement learning (RL) setting. We present an efficient procedure, designed specifically to defend against an adaptive RL adversary, that can directly certify the total reward without requiring the policy to be robust at each time-step. Our main theoretical contribution is to prove an adaptive version of the Neyman-Pearson Lemma – a key lemma for randomized smoothing-based certificates – where the adversarial perturbation at a particular time can be a stochastic function of current and previous observations and states as well as previous actions. Our robustness certificates guarantee that the final total reward remains above a certain threshold, even though the actions at intermediate time-steps may change under the attack.

Next, we design a procedure for certifying models with complex outputs such as images, text, and segmentation masks. For an adversarial perturbation of bounded ℓ_2 size, our Center Smoothing algorithm can certify the change in the output under commonly used distance metrics like perceptual distance, intersection-over-union (IoU), cosine distance, etc. Given a general neural network model, our method can make it provably robust by evaluating it on several noisy versions of the input and aggregating the predictions by computing the center of the ball that encloses at least half of the output points. The robustness certificate guarantees that the change in the output as measured by the distance metric remains bounded for an adversarial perturbation of the input.

We also study a fundamental limitation of randomized smoothing-based methods for certifying against the ℓ_∞-threat model. We show that for high-dimensional inputs like images, randomized smoothing suffers from a curse of dimensionality for a vast class of

smoothing distribution. The best possible ℓ_∞-radius obtained by randomized smoothing decreases rapidly with the number of dimensions in the input. In particular, for a general class of i.i.d. smoothing distributions, we show that, for $p > 2$, the largest ℓ_p-radius that can be certified decreases with the number of dimensions d as $O(1/d^{\frac{1}{2}-\frac{1}{p}})$. In an asymptotic sense, this dependence on dimensionality is no better than certifying using an isometric Gaussian smoothing distribution, essentially putting a matching lower bound on the robustness radius.

We also propose a method to certify the confidence of a neural network in its predictions. Most conventional neural networks output a score in the range of 0 to 1 (typically at the final softmax layer) which can be interpreted as the confidence that a model has in its prediction. This information can be crucial for several real world decision-making applications such as self-driving cars and disease-diagnosis networks, where safety is paramount. Our approach uses the distribution of the scores under several noisy versions of the input to certify the confidence of the model. We adapt the Neyman-Pearson lemma (a key theorem in randomized smoothing) for functions with bounded real-valued outputs in order to certify the expected value of the confidence over the smoothing distribution.

Another setting where we study certified robustness is that of streaming applications, such as online content recommendation and stock market analysis, where models use historical data to make predictions. In this setting, inputs are presented as a sequence of potentially correlated items, and an adversarial perturbation added to one input item could affect the predictions on subsequent input items. The adversarial threat model we consider allows the attacker to allocate different perturbation budgets to different inputs

in a dynamic fashion to optimize its attack. We derive robustness certificates for models that use a fixed-size sliding window over the input stream. Our guarantees hold for the average model performance across the entire stream and are independent of stream size, making them suitable for large and potentially infinite data streams.

8.2 Future Work

There are several ways in which our methods can be improved upon. For instance, the distance functions used in our distributional robustness certificates, such as ℓ_2-norm and parameterized transformations, are predefined non-learnable functions that may not be suitable for modeling sophisticated data shifts. A future direction of research could be to refine our distributional certificates for more complex domains such as weather patterns, user preferences, facial expressions, etc. Similarly, in our robustness certificates for reinforcement learning, the performance guarantee degrades with the length of the episodes. Our tightness result shows that this dependence could not be improved via Gaussian smoothing. Designing provable methods that have a better dependence on the length of the episodes could be an interesting direction for future research.

Our work shows fundamental limitations in designing certificates for the ℓ_∞ threat model using i.i.d. smoothing distributions. It would be an interesting direction of research to investigate smoothing distributions outside the class of i.i.d. distributions that could provide meaningful ℓ_∞ certificates. Another interesting direction could be to study certifiable robustness under non-ℓ_p threat models, such as edit distance, cosine similarity, and perceptual distance, that are capable of capturing changes in the semantic content of the input.

Bibliography

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

[4] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 10408–10418, 2019. URL http://papers.nips.cc/paper/9228-functional-adversarial-attacks.

[5] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgEMpVFwB.

[6] Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *5th International*

Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net, 2017. URL https://openreview.net/forum?id=ryvlRyBKl.

[7] Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition - 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings*, volume 10358 of *Lecture Notes in Computer Science*, pages 262–275. Springer, 2017. doi: 10.1007/978-3-319-62416-7_19. URL https://doi.org/10.1007/978-3-319-62416-7_19.

[8] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 2040–2042. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. URL http://dl.acm.org/citation.cfm?id=3238064.

[9] Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. Show-and-fool: Crafting adversarial examples for neural image captioning. *CoRR*, abs/1712.02051, 2017. URL http://arxiv.org/abs/1712.02051.

[10] Nicholas Carlini and David A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *CoRR*, abs/1801.01944, 2018. URL http://arxiv.org/abs/1801.01944.

[11] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C. Hansen. On instabilities of deep learning in image reconstruction - does AI come at a cost? *CoRR*, abs/1902.05300, 2019. URL http://arxiv.org/abs/1902.05300.

[12] Ankit Raj, Yoram Bresler, and Bo Li. Improving robustness of deep-learning-based image reconstruction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7932–7942. PMLR, 2020. URL http://proceedings.mlr.press/v119/raj20a.html.

[13] Francesco Calivá, Kaiyang Cheng, Rutwik Shah, and Valentina Pedoia. Adversarial robust training in mri reconstruction. *arXiv preprint arXiv:2011.00070*, 2020.

[14] Kaiyang Cheng, Francesco Calivá, Rutwik Shah, Misung Han, Sharmila Majumdar, and Valentina Pedoia. Addressing the false negative problem of deep learning mri reconstruction models by adversarial attacks and robust training. In

Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal, editors, *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 121–135, Montreal, QC, Canada, 06–08 Jul 2020. PMLR. URL http://proceedings.mlr.press/v121/cheng20a.html.

[15] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 36–42. IEEE Computer Society, 2018. doi: 10.1109/SPW.2018.00014. URL https://doi.org/10.1109/SPW.2018.00014.

[16] Jun-Ho Choi, Huan Zhang, Jun-Hyuk Kim, Cho-Jui Hsieh, and Jong-Seok Lee. Evaluating robustness of deep image super-resolution against adversarial attacks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 303–311. IEEE, 2019. doi: 10.1109/ICCV.2019.00039. URL https://doi.org/10.1109/ICCV.2019.00039.

[17] Minghao Yin, Yongbing Zhang, Xiu Li, and Shiqi Wang. When deep fool meets deep prior: Adversarial attack on super-resolution network. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1930–1938, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356657. doi: 10.1145/3240508.3240603. URL https://doi.org/10.1145/3240508.3240603.

[18] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian J. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=S18Su--CW.

[19] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

[20] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=H1uR4GZRZ.

[21] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5775–5783, 2017. doi: 10.1109/ICCV.2017.615. URL https://doi.org/10.1109/ICCV.2017.615.

[22] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick D. McDaniel. On the (statistical) detection of adversarial examples. *CoRR*, abs/1702.06280, 2017.

[23] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *CoRR*, abs/1704.04960, 2017. URL http://arxiv.org/abs/1704.04960.

[24] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 3–14, 2017. doi: 10.1145/3128572.3140444. URL https://doi.org/10.1145/3128572.3140444.

[25] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[26] Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aäron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5032–5041, 2018. URL http://proceedings.mlr.press/v80/uesato18a.html.

[27] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5283–5292, 2018. URL http://proceedings.mlr.press/v80/wong18a.html.

[28] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10900–10910, Red Hook, NY, USA, 2018. Curran Associates Inc.

[29] Sahil Singla and Soheil Feizi. Robustness certificates against adversarial examples for relu networks. *CoRR*, abs/1902.01235, 2019. URL http://arxiv.org/abs/1902.01235.

[30] Ping-yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studer, and Tom Goldstein. Certified defenses for adversarial patches. In *8th International Conference on Learning Representations*, 2020.

[31] Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8981–8991. PMLR, 2020. URL http://proceedings.mlr.press/v119/singla20a.html.

[32] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models, 2018. URL http://arxiv.org/abs/1810.12715.

[33] Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4081–4091, 2019. doi: 10.18653/v1/D19-1419. URL https://doi.org/10.18653/v1/D19-1419.

[34] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers, 2018.

[35] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3578–3586. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/mirman18b.html.

[36] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[37] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672, 2019. doi: 10.1109/SP.2019.00044. URL https://doi.org/10.1109/SP.2019.00044.

[38] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 9459–9469, 2019. URL http://papers.nips.cc/paper/9143-certified-adversarial-robustness-with-additive-noise.

[39] Hadi Salman, Jerry Li, Ilya P. Razenshteyn, Pengchuan Zhang, Huan Zhang, Sébastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 11289–11300, 2019.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[41] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20 (184):1–25, 2019. URL http://jmlr.org/papers/v20/19-519.html.

[42] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1802–1811. PMLR, 2019. URL http://proceedings.mlr.press/v97/engstrom19a.html.

[43] Alhussein Fawzi and Pascal Frossard. Manitest: Are classifiers really invariant? *CoRR*, abs/1507.06535, 2015. URL http://arxiv.org/abs/1507.06535.

[44] Michael A. Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4845–4854. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00498.

[45] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *CoRR*, abs/1611.05760, 2016. URL http://arxiv.org/abs/1611.05760.

[46] Yiren Zhou, Sibo Song, and Ngai-Man Cheung. On classification of distorted images with deep convolutional neural networks. *CoRR*, abs/1701.01924, 2017. URL http://arxiv.org/abs/1701.01924.

[47] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Provable robustness against wasserstein distribution shifts via input randomization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HJFVrpCaGE.

[48] Aounon Kumar, Alexander Levine, and Soheil Feizi. Policy smoothing for provably robust reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=mwdfai8NBrJ.

[49] Aounon Kumar and Tom Goldstein. Center smoothing: Certified robustness for networks with structured outputs. *Advances in Neural Information Processing Systems*, 34, 2021.

[50] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5458–5467. PMLR, 2020. URL http://proceedings.mlr.press/v119/kumar20b.html.

[51] Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[52] Aounon Kumar, Vinu Sankar Sadasivan, and Soheil Feizi. Provable robustness for streaming models with a sliding window, 2023.

[53] Samuel F. Dodge and Lina J. Karam. Understanding how image quality affects deep neural networks. *CoRR*, abs/1604.04004, 2016. URL http://arxiv.org/abs/1604.04004.

[54] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7549–7561, 2018.

[55] David J. B. Pearce and Hans-Günter Hirsch. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *INTERSPEECH*, 2000.

[56] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, volume 8190 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2013. doi: 10.1007/978-3-642-40994-3_25. URL https://doi.org/10.1007/978-3-642-40994-3_25.

[57] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[58] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

[59] Fanny Yang, Zuowen Wang, and Christina Heinze-Deml. Invariance-inducing regularization using worst-case transformations suffices to boost accuracy and spatial robustness. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14757–14768, 2019.

[60] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of*

Machine Learning Research, pages 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

[62] Songwei Ge, Shlok Mishra, Haohan Wang, Chun-Liang Li, and David W. Jacobs. Robust contrastive learning using negative samples with diminished semantics. *CoRR*, abs/2110.14189, 2021. URL https://arxiv.org/abs/2110.14189.

[63] Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5858–5868, 2019.

[64] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3353–3364, 2019.

[65] Pratyush Maini, Eric Wong, and J. Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6640–6650. PMLR, 2020. URL http://proceedings.mlr.press/v119/maini20a.html.

[66] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=dFwBosAcJkN.

[67] Alexander Levine and Soheil Feizi. Improved, deterministic smoothing for L1 certified robustness. *CoRR*, abs/2103.10834, 2021. URL https://arxiv.org/abs/2103.10834.

[68] Alexander Levine and Soheil Feizi. (de)randomized smoothing for certifiable defense against patch attacks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[69] Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. In *The Thirty-Fourth AAAI Conference on*

Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 4585–4593. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5888.

[70] Aounon Kumar, Alexander Levine, and Soheil Feizi. Policy smoothing for provably robust reinforcement learning. *CoRR*, abs/2106.11420, 2021. URL https://arxiv.org/abs/2106.11420.

[71] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. CROP: certifying robust policies for reinforcement learning through functional smoothing. *CoRR*, abs/2106.09292, 2021. URL https://arxiv.org/abs/2106.09292.

[72] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3730–3739, 2017.

[73] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 467–483. Springer, 2018. doi: 10.1007/978-3-030-01225-0_28. URL https://doi.org/10.1007/978-3-030-01225-0_28.

[74] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2692–2701, 2018.

[75] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary C. Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6872–6881. PMLR, 2019. URL http://proceedings.mlr.press/v97/wu19f.html.

[76] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). . URL http://www.cs.toronto.edu/~kriz/cifar.html.

[77] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[78] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2021.

[79] Rafael Pinot, Laurent Meunier, Florian Yger, Cédric Gouy-Pailler, Yann Chevaleyre, and Jamal Atif. On the robustness of randomized classifiers to adversarial examples. *CoRR*, abs/2102.10875, 2021. URL https://arxiv.org/abs/2102.10875.

[80] Marc Fischer, Maximilian Baader, and Martin T. Vechev. Certified defense to image transformations via randomized smoothing. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[81] Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6808–6817. PMLR, 2019. URL http://proceedings.mlr.press/v97/wong19a.html.

[82] Alexander Levine and Soheil Feizi. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks, 2019.

[83] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4058–4065. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17155.

[84] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. In *6th International*

Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hk6kPgZA-.

[85] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 137–144. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/b1b0432ceafb0ce714426e9114852ac7-Abstract.html.

[86] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representations for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7523–7532. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhao19a.html.

[87] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. Understanding the limits of unsupervised domain adaptation via data poisoning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17347–17359, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/90cc440b1b8caa520c562ac4e4bbcb51-Abstract.html.

[88] Maurice Weber, Linyi Li, Boxin Wang, Zhikuan Zhao, Bo Li, and Ce Zhang. Certifying out-of-domain generalization for blackbox functions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23527–23548. PMLR, 2022. URL https://proceedings.mlr.press/v162/weber22a.html.

[89] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. ISSN 00063444. URL http://www.jstor.org/stable/2331986.

[90] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL https://arxiv.org/pdf/1807.01069.

[91] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

[92] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

[93] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

[94] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017. URL http://arxiv.org/abs/1712.01815.

[95] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi: 10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

[96] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17:39:1–39:40, 2016. URL http://jmlr.org/papers/v17/15-522.html.

[97] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016. URL http://arxiv.org/abs/1604.07316.

[98] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=BJm4T4Kgx.

[99] Björn Lütjens, Michael Everett, and Jonathan P. How. Certified adversarial robustness for deep reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1328–1337. PMLR, 2019. URL http://proceedings.mlr.press/v100/lutjens20a.html.

[100] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on observations. *CoRR*, abs/2003.08938, 2020. URL https://arxiv.org/abs/2003.08938.

[101] J. Neyman and E. S. Pearson. *On the Problem of the Most Efficient Tests of Statistical Hypotheses*, pages 73–108. Springer New York, New York, NY, 1992. ISBN 978-1-4612-0919-5. doi: 10.1007/978-1-4612-0919-5_6. URL https://doi.org/10.1007/978-1-4612-0919-5_6.

[102] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983. doi: 10.1109/TSMC.1983.6313077.

[103] Andrew W. Moore. Efficient memory-based learning for robot control. 1990.

[104] Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8127–8138. Curran Associates, Inc., 2020.

[105] Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre M. Bayen. Robust reinforcement learning using adversarial populations. *CoRR*, abs/2008.01825, 2020. URL https://arxiv.org/abs/2008.01825.

[106] Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=sCZbhBvqQaU.

[107] Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. *CoRR*, abs/2105.11417, 2021. URL https://arxiv.org/abs/2105.11417.

[108] Aounon Kumar and Tom Goldstein. Center smoothing for certifiably robust vector-valued functions. *CoRR*, abs/2102.09701, 2021. URL https://arxiv.org/abs/2102.09701.

[109] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[110] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

[111] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. https://github.com/DLR-RM/stable-baselines3, 2019.

[112] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[113] Antonin Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

[114] Anurag Arnab, Ondrej Miksik, and Philip H. S. Torr. On the robustness of semantic segmentation models to adversarial attacks. *CoRR*, abs/1711.09856, 2017. URL http://arxiv.org/abs/1711.09856.

[115] Xiang He, Sibei Yang, Guanbin Li, Haofeng Li, Huiyou Chang, and Yizhou Yu. Non-local context encoder: Robust biomedical image segmentation against adversarial attacks. *CoRR*, abs/1904.12181, 2019. URL http://arxiv.org/abs/1904.12181.

[116] Xu Kang, Bin Song, Xiaojiang Du, and Mohsen Guizani. Adversarial attacks for image segmentation on multiple lightweight models. *IEEE Access*, 8:31359–31370, 2020. doi: 10.1109/ACCESS.2020.2973069.

[117] Fatemeh Vakhshiteh, Raghavendra Ramachandra, and Ahmad Nickabadi. Threat of adversarial attacks on face recognition: A comprehensive survey. *arXiv preprint arXiv:2007.11709*, 2020.

[118] Qing Song, Yingqi Wu, and Lu Yang. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *CoRR*, abs/1811.12026, 2018. URL http://arxiv.org/abs/1811.12026.

[119] Morgan Frearson and Kien Nguyen. Adversarial attack on facial recognition using visible light. *arXiv preprint arXiv:2011.12680*, 2020.

[120] Vladimir Shenmaier. Complexity and approximation of the smallest k-enclosing ball problem. *European Journal of Combinatorics*, 48:81–87, 2015. ISSN 0195-6698. doi: https://doi.org/10.1016/j.ejc.2015.02.011. URL http://www.sciencedirect.com/science/article/pii/S0195669815000335.

[121] Jiaye Teng, Guang-He Lee, and Yang Yuan. ℓ_1 adversarial robustness certificates: a randomized smoothing approach, 2020. URL https://openreview.net/forum?id=H1lQIgrFDS.

[122] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness, 2019. URL https://openreview.net/forum?id=SyxaYsAqY7.

[123] Alexander Levine, Aounon Kumar, Thomas Goldstein, and Soheil Feizi. Tight second-order certificates for randomized smoothing, 2020.

[124] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi S. Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 4911–4922, 2019. URL http://papers.nips.cc/paper/8737-tight-certificates-of-adversarial-robustness-for-randomly-sm

[125] Ping yeh Chiang, Michael J. Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection by median smoothing, 2020.

[126] Adva Wolf. Making medical image reconstruction adversarially robust. 2019. URL http://cs229.stanford.edu/proj2019spr/report/97.pdf.

[127] Marc Fischer, Maximilian Baader, and Martin T. Vechev. Scalable certified segmentation via randomized smoothing. *CoRR*, abs/2107.00228, 2021. URL https://arxiv.org/abs/2107.00228.

[128] Mihai Bundefineddoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 250–257, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581134959. doi: 10.1145/509907.509947. URL https://doi.org/10.1145/509907.509947.

[129] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00068. URL http:

//openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

[130] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878, 2016. URL http://arxiv.org/abs/1604.02878.

[131] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[132] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

[133] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

[134] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). . URL http://www.cs.toronto.edu/~kriz/cifar.html.

[135] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1lWUoA9FQ.

[136] Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. *International Conference on Machine Learning (ICML)*, 2020.

[137] Alexander Levine and Soheil Feizi. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3938–3947. PMLR, 2020. URL http://proceedings.mlr.press/v108/levine20a.html.

[138] Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA,*

February 7-12, 2020, pages 4585–4593. AAAI Press, 2020. ISBN 978-1-57735-823-7. URL https://aaai.org/ojs/index.php/AAAI/article/view/5888.

[139] Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. *CoRR*, abs/2002.09169, 2020. URL https://arxiv.org/abs/2002.09169.

[140] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random smoothing might be unable to certify ℓ_∞ robustness for high-dimensional images, 2020.

[141] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses, 2020.

[142] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[143] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016.

[144] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, 10 2011. ISSN 1067-5027. doi: 10.1136/amiajnl-2011-000291. URL https://doi.org/10.1136/amiajnl-2011-000291.

[145] Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes, 2020.

[146] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Ann. Math. Statist.*, 27(3):642–669, 09 1956. doi: 10.1214/aoms/1177728174. URL https://doi.org/10.1214/aoms/1177728174.

[147] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 2141–2149, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098117. URL https://doi.org/10.1145/3097983.3098117.

[148] Christopher Krauss, Xuan Anh Do, and Nicolas Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on

the s&p 500. *European Journal of Operational Research*, 259(2):689–702, 2017. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2016.10.031. URL https://www.sciencedirect.com/science/article/pii/S0377221716308657.

[149] Jerzy Korczak and Marcin Hemes. Deep learning for financial time series forecasting in a-trader system. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 905–912, 2017. doi: 10.15439/2017F449.

[150] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2017.11.054. URL https://www.sciencedirect.com/science/article/pii/S0377221717310652.

[151] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020.

[152] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3995–4001. AAAI Press, 2015. ISBN 9781577357384.

[153] Francisco Javier Ordonez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016. ISSN 1424-8220. doi: 10.3390/s16010115. URL https://www.mdpi.com/1424-8220/16/1/115.

[154] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59:235–244, 2016. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.04.032. URL https://www.sciencedirect.com/science/article/pii/S0957417416302056.

[155] Alex Graves and JÃŒrgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL https://www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN 2005.

[156] Don Dennis, Durmus Alp Emre Acar, Vikram Mandikal, Vinu Sankar Sadasivan, Venkatesh Saligrama, Harsha Vardhan Simhadri, and Prateek Jain. Shallow rnn: Accurate time-series classification on resource constrained devices. In H. Wallach, H. Larochelle, A. Beygelzimer,

F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/76d7c0780ceb8fbf964c102ebc16d75f-Paper.pdf.

[157] Roger Hsiao, Dogan Can, Tim Ng, Ruchir Travadi, and Arnab Ghoshal. Online automatic speech recognition with listen, attend and spell model. *IEEE Signal Processing Letters*, 27:1889–1893, 2020.

[158] Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=83A-0x6Pfi_.

[159] Andjela Mladenovic, Joey Bose, Hugo berard, William L. Hamilton, Simon Lacoste-Julien, Pascal Vincent, and Gauthier Gidel. Online adversarial attacks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=bYGSzbCM_i.

[160] Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 63–80. ACM, 2020. doi: 10.1145/3375395.3387658. URL https://doi.org/10.1145/3375395.3387658.

[161] Omri Ben-Eliezer and Eylon Yogev. The adversarial robustness of sampling. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 49–62. ACM, 2020. doi: 10.1145/3375395.3387643. URL https://doi.org/10.1145/3375395.3387643.

[162] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.09.027. URL https://www.sciencedirect.com/science/article/pii/S1568494617305665.

[163] C. Stamate, G.D. Magoulas, S. Kueppers, E. Nomikou, I. Daskalopoulos, M.U. Luchini, T. Moussouri, and G. Roussos. Deep learning parkinson's from smartphone data. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 31–40, 2017. doi: 10.1109/PERCOM.2017.7917848.

[164] Garam Lee, Kwangsik Nho, Byungkon Kang, Kyung-Ah Sohn, and Dokyoon Kim. Predicting alzheimer's disease progression using multi-modal deep learning approach. *Scientific reports*, 9(1):1–12, 2019.

[165] Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of translational medicine*, 8(11), 2020.

[166] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3530–3538. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.376. URL https://doi.org/10.1109/CVPR.2017.376.

[167] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020.

[168] Vinu Sankar Sadasivan, Mahdi Soltanolkotabi, and Soheil Feizi. Cuda: Convolution-based unlearnable datasets, 2023.

[169] Yonathan Efroni, Chi Jin, Akshay Krishnamurthy, and Sobhan Miryoosefi. Provable reinforcement learning with a short-term memory. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5832–5850. PMLR, 2022. URL https://proceedings.mlr.press/v162/efroni22a.html.

[170] JL Reyes-Ortiz, D Anguita, A Ghio, L Oneto, and X Parra. Uci machine learning repository: Human activity recognition using smartphones data set, 2012.

[171] P. Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *ArXiv e-prints*, April 2018. URL https://arxiv.org/abs/1804.03209.

[172] Slobodan Mitrovic, Ilija Bogunovic, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. Streaming robust submodular maximization: A partitioned thresholding approach. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4557–4566, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3baa271bc35fe054c86928f7016e8ae6-Abstract.html.

[173] Moses Ganardi, Danny Hucke, and Markus Lohrey. Derandomization for sliding window algorithms with strict correctness. In René van Bevern and

Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 237–249. Springer, 2019. doi: 10.1007/978-3-030-19955-5_21. URL https://doi.org/10.1007/978-3-030-19955-5_21.

[174] Joan Feigenbaum, Sampath Kannan, and Jian Zhang. Computing diameter in the streaming and sliding-window models. *Algorithmica*, 41(1):25–41, 2005. doi: 10.1007/s00453-004-1105-2. URL https://doi.org/10.1007/s00453-004-1105-2.

[175] Mayur Datar and Rajeev Motwani. *The Sliding-Window Computation Model and Results*, pages 149–167. Springer US, Boston, MA, 2007. ISBN 978-0-387-47534-9. doi: 10.1007/978-0-387-47534-9_8. URL https://doi.org/10.1007/978-0-387-47534-9_8.

[176] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Certifying model accuracy under distribution shifts. *CoRR*, abs/2201.12440, 2022. URL https://arxiv.org/abs/2201.12440.

[177] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 421–425. IEEE, 2017.

[178] Dingzeyu Li, Timothy R Langlois, and Changxi Zheng. Scene-aware audio for 360 videos. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.