Globally-Robust Neural Networks

Klas Leino [1] **Zifan Wang** [1] **Matt Fredrikson** [1]

Abstract

The threat of adversarial examples has motivated work on training *certifiably robust* neural networks to facilitate efficient verification of *local robustness* at inference time. We formalize a notion of *global robustness*, which captures the operational properties of on-line local robustness certification while yielding a natural learning objective for robust training. We show that widely-used architectures can be easily adapted to this objective by incorporating efficient global Lipschitz bounds into the network, yielding certifiably-robust models *by construction* that achieve *state-of-the-art* verifiable accuracy. Notably, this approach requires significantly less time and memory than recent certifiable training methods, and leads to negligible costs when certifying points on-line; for example, our evaluation shows that it is possible to train a large robust Tiny-Imagenet model in a matter of hours. Our models effectively leverage inexpensive global Lipschitz bounds for real-time certification, despite prior suggestions that tighter local bounds are needed for good performance; we posit this is possible because our models are specifically trained to achieve tighter global bounds. Namely, we prove that the maximum achievable verifiable accuracy for a given dataset is not improved by using a local bound.

1. Introduction

We consider the problem of training neural networks that are robust to input perturbations with bounded ℓ_p norm. Precisely, given an input point, x, network, F, and norm bound, ϵ, this means that F makes the same prediction on all points within the ℓ_p-ball of radius ϵ centered at x.

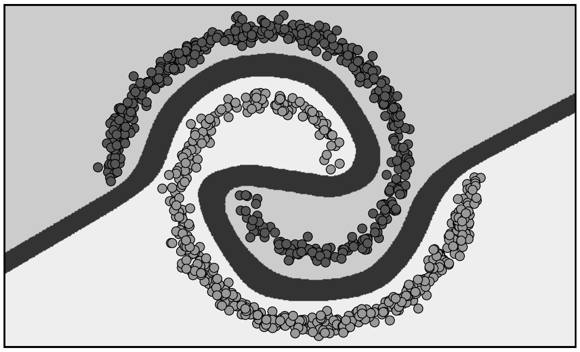

Figure 1: Illustration of global robustness. The model abstains from predicting on the margin between the classes (dark gray), which has width at least ϵ.

This problem is significant as deep neural networks have been shown to be vulnerable to *adversarial examples* (Papernot et al., 2016; Szegedy et al., 2014), wherein perturbations are chosen to deliberately cause misclassification. While numerous heuristic solutions have been proposed to address this problem, these solutions are often shown to be ineffective by subsequent adaptive attacks (Carlini & Wagner, 2017). Thus, this paper focuses on training methods that produce models whose robust predictions can be efficiently certified against adversarial perturbations (Lee et al., 2020; Tsuzuku et al., 2018; Wong et al., 2018).

We begin by introducing a notion of *global robustness* for classification models (Section 2.1), which requires that classifiers maintain a separation of width at least ϵ (in feature space) between any pair of regions that are assigned different prediction labels. This separation means that there are certain inputs on which a globally-robust classifier must refuse to give a prediction, instead signaling that a violation has occurred (see Figure 1). While requiring the model to abstain in some cases may at first appear to be a hindrance, we note that in operational terms this is no different than composing a model with a routine that only returns predictions when $\epsilon/2$-local robustness can be certified.

While it is straightforward to construct a globally-robust model in this way via composition with a certification procedure, doing so with most current certification methods leads to severe penalties on performance or utility. Most techniques for verifying local robustness are costly even on small models (Fischetti & Jo, 2018; Fromherz et al., 2021; Gehr et al., 2018; Jordan et al., 2019; Tjeng & Tedrake,

[1] Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Correspondence to: Klas Leino <kleino@cs.cmu.edu>, Matt Fredrikson <mfredrik@cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

2017), requiring several orders of magnitude more time than a typical forward pass of a network; on moderately-large CNNs, these techniques either time out after minutes or hours, or simply run out of memory.

One approach to certification that shows promise in this regard uses Lipschitz bounds to efficiently calculate the robustness region around a point (Weng et al., 2018a; Zhang et al., 2018). In particular, when *global* bounds are used with this approach, it is possible to implement the bound computation as a neural network of comparable size to the original (Section 4.2), making on-line certification nearly as efficient as inference. Unfortunately, current training methods do not produce models with sufficiently small global bounds for this to succeed (Weng et al., 2018a). Recent work (Lee et al., 2020) explored the possibility of training networks with sufficiently small *local* bounds, but the training cost in time and memory remains prohibitive in many cases.

Surprisingly, we find that using global Lipschitz bounds for certification may not be as limiting as previously thought (Huster et al., 2018; Yang et al., 2020). We show that for any set of points that can be robustly-classified using a local Lipschitz bound, there exists a model whose global bound implies the same robust classification (Theorem 3). This motivates a new approach to certifiable training that makes exclusive use of global bounds (Section 2.2). Namely, we construct a globally-robust model that incorporates a Lipschitz bound in its forward pass to define an additional "robustness violation" class, and use standard training methods to discourage violations while simultaneously encouraging accuracy.

Focusing on the case of deterministic guarantees against ℓ_2-bounded perturbations, we show that this approach yields state-of-the-art verified-robust accuracy (VRA), while imposing little overhead during training and *none* during certification. For example, we find that we can achieve 63% VRA with a large robustness radius of $\epsilon = 1.58$ on MNIST, surpassing all prior approaches by multiple percentage points. We also achieve state-of-the-art VRA on CIFAR-10, and scale to larger applications such as Tiny-Imagenet (see Section 5).

To summarize, we provide a method for training certifiably-robust neural networks that is simple, fast, capable of running with limited memory, and that yields state-of-the-art deterministic verified accuracy. We prove that the potential of our approach is not hindered by its simplicity; rather, its simplicity is an asset—our empirical results demonstrate the many benefits it enjoys over more complicated methods.

2. Constructing Globally-Robust Networks

In this section we present our method for constructing globally-robust networks, which we will refer to as *GloRo Nets*. We begin in Section 2.1 by formally introducing our notion of *global robustness*, after briefly covering the essential background and notation. We then show how to mathematically construct GloRo Nets in Section 2.2, and prove that our construction is globally robust.

2.1. Global Robustness

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a neural network that categorizes points into m different classes. Let F be the function representing the predictions of f, i.e., $F(x) = \text{argmax}_i\{f_i(x)\}$.

F is said to be ϵ-*locally-robust* at point x if it makes the same prediction on all points in the ϵ-ball centered at x (Definition 1).

Definition 1. *(Local Robustness) A model, F, is* ϵ-locally-robust *at point, x, with respect to norm, $||\cdot||$, if $\forall\, x'$,*

$$||x - x'|| \leq \epsilon \implies F(x) = F(x').$$

Most work on robustness verification has focused on this local robustness property; in this work, we present a natural notion of *global robustness*, which captures the operational properties of on-line local robustness certification.

Clearly, local robustness cannot be simultaneously satisfied at every point—unless the model is entirely degenerate, there will always exist points that are arbitrarily close to a decision boundary. Instead, we will introduce a global robustness definition that can be satisfied even on models with non-trivial behavior by using an additional class, $\bot$, that signals that a point cannot be certified as globally robust. At a high level, we can think of separating each of the classes with a margin of width at least ϵ in which the model always predicts $\bot$. In order to satisfy global robustness, we require that no two points at distance ϵ from one another are labeled with different non-$\bot$ classes.

More formally, let us define the following relation ($\stackrel{\bot}{=}$): we will say that $c_1 \stackrel{\bot}{=} c_2$ if $c_1 = \bot$ or $c_2 = \bot$ or $c_1 = c_2$. Using this relation, we provide our formal notion of global robustness in Definition 2.

Definition 2. *(Global Robustness) A model, F, is* ϵ-globally-robust, *with respect to norm, $||\cdot||$, if $\forall\, x_1, x_2$,*

$$||x_1 - x_2|| \leq \epsilon \implies F(x_1) \stackrel{\bot}{=} F(x_2).$$

An illustration of global robustness is shown in Figure 1. While global robustness can clearly be trivially satisfied by labeling all points as $\bot$, we note that the objective of robust training is typically to achieve high robustness *and* accuracy (i.e., VRA), thus ideally only points off the data manifold are labeled $\bot$, as illustrated in Figure 1.

2.2. Certified Globally-Robust Networks

Because of the threat posed by adversarial examples, and the elusiveness of such attacks against heuristic defenses (Carlini

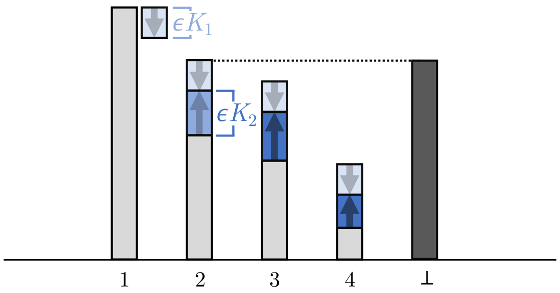

Figure 2: Illustration of calculating the $\perp$ logit. Note that ϵK_i provides a bound on changes to logit i within an ϵ-ball. The $\perp$ logit is chosen to account for the predicted class *decreasing* by the maximum amount and each other class *increasing* by the maximum amount. If the $\perp$ logit does not surpass that of the predicted class, then no class can overtake the predicted class within an ϵ-ball (Theorem 1).

& Wagner, 2017), there has been a volume of previous work seeking to verify local robustness on specific points of interest. In this work, we shift our focus to global robustness directly, resulting in a method for producing models that make predictions that are verifiably robust *by construction*.

Intuitively, we aim to instrument a model with an extra output, $\perp$, that labels a point as "not locally-robust," such that the instrumented model predicts a non-$\perp$ class *only if the point is locally-robust* (with respect to the original model). At a high level, we do this by ensuring that in order to avoid predicting $\perp$, the maximum output of f must surpass the other outputs by a sufficient margin. While this margin is measured in the output space, we can ensure it is sufficiently large to ensure local robustness by relating the output space to the input space via an upper bound on the model's Lipschitz constant.

Suppose that K_i is an upper bound on the Lipschitz constant for f_i. I.e., for all x_1, x_2, Equation 1 holds. Intuitively, K_i bounds the largest possible change in the logit output for class i per unit change in the model's input.

$$\frac{|f_i(x_1) - f_i(x_2)|}{||x_1 - x_2||} \leq K_i \tag{1}$$

Let $y = f(x)$, and let $j = F(x)$, i.e., the class predicted on point x. Let $y_\perp = \max_{i \neq j}\{y_i + (K_i + K_j)\epsilon\}$. Intuitively, $y_\perp$ captures the value that the class that is most competitive with the chosen class would take under the worst-case change to x within an ϵ-ball. Figure 2 provides an illustration of this intuition.

We then define the instrumented model, or GloRo Net, $\bar{f}^\epsilon$, as follows: $\bar{f}_i^\epsilon(x) ::= y_i \; \forall i \in [m]$ and $\bar{f}_\perp^\epsilon(x) ::= y_\perp$; that is, $\bar{f}^\epsilon$ concatenates $y_\perp$ with the output of f.

We show that the predictions of this GloRo Net, $\bar{F}^\epsilon$, can be used to certify the predictions of the instrumented model, F: whenever $\bar{F}^\epsilon$ predicts a class that is not $\perp$, the prediction coincides with the prediction of F, and F is guaranteed to be locally robust at that point (Theorem 1).

Theorem 1. *If $\bar{F}^\epsilon(x) \neq \perp$, then $\bar{F}^\epsilon(x) = F(x)$ and F is ϵ-locally-robust at x.*

The proof of Theorem 1 is given in Appendix A.1 in the supplementary material.

Note that in this formulation, we assume that the predicted class, j, will decrease by the maximum amount within the ϵ-ball, while all other classes increase by their respective maximum amounts. This is a conservative assumption that guarantees local robustness; however, in practice, we can dispose of this assumption by instead calculating the Lipschitz constant of the margin by which the logit of the predicted class surpasses the other logits, i.e., the Lipschitz constant of $y_j - y_i$ for $i \neq j$. The details of this tighter variant are presented in Appendix A.2 in the supplementary material, along with the corresponding correctness proof.

Notice that the GloRo Net, $\bar{F}^\epsilon$, will always predict $\perp$ on points that lie directly on the decision boundary of F. Moreover, any point that is within ϵ of the decision boundary will also be labeled as $\perp$ by $\bar{F}^\epsilon$. From this, it is perhaps clear that GloRo Nets achieve global robustness (Theorem 2).

Theorem 2. *$\bar{F}^{\epsilon/2}$ is ϵ-globally-robust.*

The proof of Theorem 2 is given in Appendix A.3 in the supplementary material.

3. Revisiting the Global Lipschitz Constant

The global Lipschitz constant gives a bound on the maximum rate of change in the network's output over the entire input space. For the purpose of certifying robustness, it suffices to bound the maximum rate of change in the network's output over any pair of points *within the ϵ-ball* centered at the point being certified, i.e., the *local* Lipschitz constant. Recent work has explored methods for obtaining upper bounds on the local Lipschitz constant (Weng et al., 2018a; Zhang et al., 2018; Lee et al., 2020); the construction of GloRo Nets given in Section 2 remains correct whether K represents a global or a local Lipschitz constant.

The advantage to using a local bound is, of course, that we may expect tighter bounds; after all, the local Lipschitz constant is no larger than the global Lipschitz constant. However, using a local bound also has its drawbacks. First, a local bound is typically more expensive to compute. In particular, a local bound always requires more memory, as each instance has its own bound, hence the required memory grows with the batch size. This in turn reduces the amount of parallelism that can be exploited when using a local bound, reducing the model's throughput.

Furthermore, because the local Lipschitz constant is different for every point, it must be computed every time the network sees a new point. By contrast, the global bound can be computed in advance, meaning that verification via the global bound is essentially free. This makes the global bound advantageous, assuming that it can be effectively leveraged for verification.

It may seem initially that a local bound would have greater prospects for successful certification. First, *local* Lipschitzness is sufficient for robustly classifying well-separated data (Yang et al., 2020); that is, global Lipschitzness is not necessary. Meanwhile, global bounds on typical networks have been found to be prohibitively large (Weng et al., 2018a), while local bounds on in-distribution points may tend to be smaller on the same networks. However, the potential disadvantages of a global bound become less clear if the model is specifically trained to have a small global Lipschitz constant.

For example, GloRo Nets that use a global Lipschitz constant will be penalized for incorrect predictions if the global Lipschitz constant is not sufficiently small to verify its predictions; therefore, the loss actively discourages any unnecessary steepness in the network function. In practice, this natural regularization of the global Lipschitz constant may serve to make the steepness of the network function more uniform, such that the global Lipschitz constant will be similar to the local Lipschitz constant.

We show that this is possible in theory, in that for any network for which local robustness can be verified on some set of points using the local Lipschitz constant, there exists a model on which the same points can be certified using the global Lipschitz constant (Theorem 3). This suggests that if training is successful, our approach has the same potential using a global bound as using a local bound.

Theorem 3. *Let f be a binary classifier that predicts $1 \iff f(x) > 0$. Let $K_L(x,\epsilon)$ be the local Lipschitz constant of f at point x with radius ϵ.*

Suppose that for some finite set of points, S, $\forall x \in S$, $|f(x)| > \epsilon K_L(x,\epsilon)$, i.e., all points in S can be verified via the local Lipschitz constant.

Then there exists a classifier, g, with global Lipschitz constant K_G, such that $\forall x \in S$, (1) g makes the same predictions as f on S, and (2) $|g(x)| > \epsilon K_G$, i.e., all points in S can be verified via the global Lipschitz constant.

Theorem 3 is stated for binary classifiers, though the result holds for categorical classifiers as well. Details on the categorical case and the proof of Theorem 3 can be found in Appendix A.4 in the supplementary material; however we provide the intuition behind the construction here. The proof relies on the following lemma, which states that among locally-robust points, points that are classified differently from one another are 2ϵ-separated. The proof of Lemma 1 can be found in Appendix A.5 in the supplementary material.

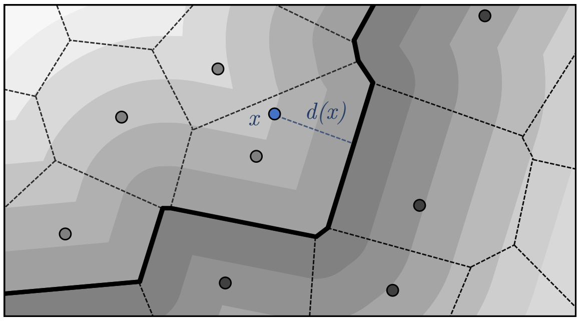

Figure 3: Illustration of a function, g, constructed to satisfy Theorem 3. The points in S are shown in light and dark gray, with different shades indicating different labels. The Voronoi tessellation is outlined in black, and the faces belonging to the decision boundary are highlighted in bold. The level curves of g are shown in various shades of gray and correspond to points, x, at some fixed distance, $d(x)$, from the decision boundary.

Lemma 1. *Suppose that for some classifier, F, and some set of points, S, $\forall x \in S$, F is ϵ-locally-robust at x. Then $\forall x_1, x_2 \in S$ such that $F(x_1) \neq F(x_2)$, $||x_1 - x_2|| > 2\epsilon$.*

In the proof of Theorem 3, we construct a function, g, whose output on point x increases linearly with x's minimum distance to any face in the Voronoi tessellation of S that separates points in S with different labels. An illustration with an example of g is shown in Figure 3. Notably, the local Lipschitz constant of g is everywhere the same as the global constant.

However, we note that while Theorem 3 suggests that networks exist *in principle* on which it is possible to use the global Lipschitz constant to certify 2ϵ-separated data, it may be that such networks are not easily obtainable via training. Furthermore, as raised by Huster et al. (2018), an additional potential difficulty in using the global bound for certification is the estimation of the global bound itself. Methods used for determining an upper bound on the global Lipschitz constant, such as the method presented later in Section 4.2, may provide a loose upper bound that is insufficient for verification even when the true bound would suffice. Nevertheless, our evaluation in Section 5 shows that in practice the global bound *can* be used effectively for certification (Section 5.1), and that the bounds obtained on the models trained with our approach are far tighter than those obtained on standard models (Section 5.3).

4. Implementation

In this section we describe how GloRo Nets can be trained and implemented. Section 4.1 covers training, as well as the loss functions used in our evaluation, and Section 4.2

provides detail on how we compute an upper bound on the global Lipschitz constant.

4.1. Training

Crucially, because certification is intrinsically captured by the GloRo Net's predictions—specifically, the $\bot$ class represents inputs that cannot be verified to be locally robust—a standard learning objective for a GloRo Net corresponds to a robust objective on the original model that was instrumented. That is, we can train a GloRo Net by simply appending a zero to the one-hot encodings of the original data labels (signifying that $\bot$ is never the correct label), and then optimizing a standard classification learning objective, e.g., with cross-entropy loss. Using this approach, $\bar{F}^\epsilon$ will be penalized for incorrectly predicting each point, x, unless x is both predicted correctly *and* F is ϵ-locally-robust at x.

While the above approach is sufficient for training models with competitive VRA, we find that the resulting VRA can be further improved using a loss inspired by TRADES (Zhang et al., 2019), which balances separately the goals of making *correct* predictions, and making *robust* predictions. Recent work (Yang et al., 2020) has shown that TRADES effectively controls the local Lipschitz continuity of networks. While TRADES is implemented using adversarial perturbations, which provide an under-approximation of the robust error, GloRo Nets naturally lend themselves to a variant that uses an over-approximation, as shown in Definition 3.

Definition 3. *(TRADES Loss for GloRo Nets) Given a network, f, cross-entropy loss L_{CE}, and parameter, λ, the GloRo-TRADES loss (L_T) of (x,y) is*

$$L_T(x,y) = L_{CE}\big(f(x),y\big) + \lambda D_{KL}\big(\bar{f}^\epsilon(x)||f(x)\big)$$

Intuitively, L_T combines the normal classification loss with the over-approximate robust loss, assuming that the class predicted by the underlying model is correct. Empirically we find that using the KL divergence, D_{KL}, in the second term produces the best results, although in many cases using L_{CE} in both terms works as well.

4.2. Bounding the Global Lipschitz Constant

There has been a great deal of work on calculating upper bounds on the Lipschitz constants of neural networks (see Section 6 for a discussion). Our implementation uses the fact that the product of the spectral norm of each of the individual layers of a feed-forward network provides an upper bound on the Lipschitz constant of the entire network (Szegedy et al., 2014). That is, if the output at class i of a neural network can be decomposed into a series of k transformations, i.e., $f_i = h^k \circ h^{k-1} \circ \cdots \circ h^1$, then Equation 2 holds (where $||\cdot||$ is the spectral norm).

$$K_i \le \prod_{j=1}^{k} ||h^j|| \qquad (2)$$

In the case of a CNN consisting of convolutional layers, dense layers, and ReLU activations, we use 1 for the spectral norm of each of the ReLU layers, and we use the power method (Farnia et al., 2019; Gouk et al., 2021) to compute the spectral norm of the convolutional and dense layers. Gouk et al. also give a procedure for bounding the spectral norm of skip connections and batch normalization layers, enabling this approach on ResNet architectures. For more complicated networks, there is a growing body of work on computing layer-wise Lipschitz bounds for various types of layers that are commonly used in neural networks (Zou et al., 2019; Fazlyab et al., 2019; Sedghi et al., 2019; Singla & Feizi, 2019; Miyato et al., 2018).

The power method may need several iterations to converge; however, we can reduce the number of iterations required at each training step by persisting the state of the power method iterates across steps. While this optimization may not guarantee an upper bound, this fact is inconsequential so long as we still obtain a model that can be certified with a true upper bound that is computed after training; this is actually not unreasonable to expect, presuming the underlying model parameters do not change too quickly. With a small number of iterations, the additional memory required to compute the Lipschitz constant via this method is approximately the same as to run the network on a single instance.

At test time, the power method must be run to convergence; however, after training, the global Lipschitz bound will remain unchanged and therefore it can be computed once in advance. This means that new points can be certified with *no additional non-trivial overhead*.

ℓ_∞ Bounds. While in this work, we focus on the ℓ_2 norm, the ideas presented in Section 2 can be applied to other norms, including the ℓ_∞ norm. However, we find that the analogue of the approximation of the global Lipschitz bound given by Equation 2 is loose in ℓ_∞ space. Meanwhile, a large volume of prior work applies ℓ_∞-specific certification strategies that proven effective for ℓ_∞ certification (Zhang et al., 2020; Balunovic & Vechev, 2020; Gowal et al., 2019).

5. Evaluation

In this section, we present an empirical evaluation of our method. We first compare GloRo Nets with several certified training methods from the recent literature in Section 5.1. We also report the training cost, in terms of per-epoch time and peak memory usage, required to train and certify the robustness of our method compared with other competitive approaches (Section 5.2). We end by demonstrating the relative tightness of the estimated Lipschitz bounds for GloRo Nets in Section 5.3. We compare against the KW (Wong et al., 2018) and BCP (Lee et al., 2020) certified training algorithms, which prior work (Lee et al., 2020; Croce et al., 2019) reported to achieve the best verified accuracy on MNIST (LeCun et al., 2010), CIFAR-10 (Krizhevsky, 2009) and Tiny-Imagenet (Le & Yang, 2015) relative to other previous certified training methods for ℓ_2 robustness.

We train GloRo nets to certify robustness against ℓ_2 perturbations within an ϵ-neighborhood of 0.3 and 1.58 for MNIST and $36/255$ for CIFAR-10 and Tiny-Imagenet (these are the ℓ_2 norm bounds that have been commonly used in the previous literature). For each model, we report the *clean accuracy*, i.e., the accuracy without verification on non-adversarial inputs, the *PGD accuracy*, i.e., the accuracy under adversarial perturbations found via the PGD attack (Madry et al., 2018), and the *verified-robust accuracy* (VRA), i.e., the fraction of points that are both correctly classified *and* certified as robust. For KW and BCP, we report the corresponding best VRAs from the original respective papers when possible, but measure training and certification costs on our hardware for an equal comparison. We run the PGD attacks using ART (Nicolae et al., 2019) on our models and on any of the models from the prior work for which PGD accuracy is not reported. When training BCP models for MNIST with $\epsilon = 0.3$, we found a different set of hyperparameters that outperforms those given by Lee et al.. For GloRo Nets, we found that MinMax activations (Anil et al., 2019) performed better than ReLU activations (see Appendix D in the supplementary material for more details); for all other models, ReLU activations were used.

Further details on the precise hyperparameters used for training and attacks, the process for obtaining these parameters, and the network architectures are provided in Appendix B. An implementation of our approach is available on GitHub[1].

5.1. Verified Accuracy

We first compare the VRA obtained by GloRo Nets to the VRA achieved by prior deterministic approaches. KW and BCP have been found to achieve the best VRA on the datasets commonly used in the previous literature (Lee et al., 2020; Croce et al., 2019). In Appendix C we provide a more comprehensive comparison to the VRAs that have been reported in prior work.

Figure 4a gives the best VRA achieved by standard training, GloRo Nets, KW, and BCP on several benchmark datasets and architectures. In accordance with prior work, we include the clean accuracy and the PGD accuracy as well. Whereas the VRA gives a lower bound on the number of correctly-classified points that are locally robust, the PGD accuracy serves as an upper bound on the same quantity. We also provide the (probabilistic) VRA achieved via Randomized Smoothing (RS) (Cohen et al., 2019a) on each of the datasets in our evaluation, as a comparison to stochastic certification.

We find that GloRo Nets consistently outperform the previous state-of-the-art deterministic VRA. On MNIST, GloRo Nets outperform all previous approaches with both ℓ_2 bounds commonly used in prior work ($\epsilon = 0.3$ and $\epsilon = 1.58$). When $\epsilon = 0.3$, the VRA begins to approach the clean accuracy of the standard-trained model; for this bound, GloRo Nets outperform the previous best VRA (achieved by KW) by nearly two percentage points, accounting for roughly 33% of the gap between the VRA of KW and the clean accuracy of the standard model. For $\epsilon = 1.58$, GloRo Nets improve upon the previous best VRA (achieved by BCP) by approximately 15 percentage points—in fact, the VRA achieved by GloRo Nets in this setting even slightly exceeds that of Randomized Smoothing, despite the fact that RS provides only a stochastic guarantee. On CIFAR-10, GloRo Nets exceed the best VRA (achieved by BCP) by approximately 7 percentage points. Finally, on Tiny-Imagenet, GloRo Nets outperform BCP by approximately 2 percentage points, improving the state-of-the-art VRA by roughly 10%. KW was unable to scale to Tiny-Imagenet due to memory pressure.

The results achieved by GloRo Nets in Figure 4a are achieved using MinMax activations (Anil et al., 2019) rather than ReLU activations, as we found MinMax activations provide a substantial performance boost to GloRo Nets. We note, however, that both KW and BCP tailor their analysis specifically to ReLU activations, meaning that they would require non-trivial modifications to support MinMax activations. Meanwhile, GloRo Nets easily support generic activation functions, provided the Lipschitz constant of the activation can be bounded (e.g., the Lipschitz constant of a MinMax activation is 1). Moreover, even with ReLU activations, GloRo Nets outperform or match the VRAs of KW and BCP; Appendix D in the supplementary material provides these results for comparison.

5.2. Training and Certification Cost

A key advantage to GloRo Nets over prior approaches is their ability to achieve state-of-the-art VRA (see Section 5.1) using a global Lipschitz bound. As discussed in Section 3, this confers performance benefits—both at train and test time—over using a local bound (e.g., BCP), or other expensive approaches (e.g., KW).

Figure 4a shows the cost of each approach both in time per epoch and in memory during training (results given for CIFAR-10). All timings were taken on a machine using a Geforce RTX 3080 accelerator, 64 GB memory, and Intel i9 10850K CPU, with the exception of those for the KW (Wong et al., 2018) method, which were taken on a Titan RTX card

[1] Code available at `https://github.com/klasleino/gloro`

method	Model	Clean (%)	PGD (%)	VRA(%)	Sec./epoch	# Epochs	Mem. (MB)
MNIST $(\epsilon = 0.3)$							
Standard	2C2F	99.2	96.9	0.0	0.3	100	0.6
GloRo	2C2F	99.0	97.8	**95.7**	0.9	500	0.7
KW	2C2F	98.9	97.8	94.0	66.9	100	20.2
BCP	2C2F	93.4	89.5	84.7	44.8	300	12.6
RS*	2C2F	98.8	-	97.4	-	-	-
MNIST $(\epsilon = 1.58)$							
Standard	4C3F	99.0	45.4	0.0	0.9	42†	2.2
GloRo	4C3F	97.0	81.9	**62.8**	3.7	300	2.7
KW	4C3F	88.1	67.9	44.5	138.1	60	84.0
BCP	4C3F	92.4	65.8	47.9	43.4	60	12.6
RS*	4C3F	99.0	-	59.1	-	-	-
CIFAR-10 $(\epsilon = {}^{36}/_{255})$							
Standard	6C2F	85.7	31.9	0.0	1.8	115†	2.5
GloRo	6C2F	77.0	69.2	**58.4**	6.9	800	3.6
KW	6C2F	60.1	56.2	50.9	516.8	60	100.9
BCP	6C2F	65.7	60.8	51.3	47.5	200	12.7
RS*	6C2F	74.1	-	64.2	-	-	-
Tiny-Imagenet $(\epsilon = {}^{36}/_{255})$							
Standard	8C2F	35.9	19.4	0.0	10.7	58†	6.7
GloRo	8C2F	35.5	32.3	**22.4**	40.3	800	10.4
KW	-	-	-	-	-	-	-
BCP	8C2F	28.8	26.6	20.1	798.8	102	715.2
RS*	8C2F	23.4	-	16.9	-	-	-

(a)

method	Model	Time (sec.)	Mem. (MB)
GloRo	6C2F	0.4	1.8
KW	6C2F	2,515.6	1,437.5
BCP	6C2F	5.8	19.1
RS*	6C2F	36,845.5	19.8

(b)

method	global UB	global LB	local LB
MNIST $(\epsilon = 1.58)$			
Standard	$5.4 \cdot 10^4$	$1.4 \cdot 10^2$	17.1
GloRo	2.3	1.9	0.8
CIFAR-10 $(\epsilon = {}^{36}/_{255})$			
Standard	$1.2 \cdot 10^7$	$1.1 \cdot 10^3$	96.2
GloRo	15.8	11.0	3.7
Tiny-Imagenet $(\epsilon = {}^{36}/_{255})$			
Standard	$2.2 \cdot 10^7$	$3.6 \cdot 10^2$	40.7
GloRo	12.5	5.9	0.8

(c)

Figure 4: **(a)** Certifiable training evaluation results on benchmark datasets. Best results are highlighted in bold. Randomized Smoothing (RS) is marked with a * superscript to indicate that it provides only a *stochastic* robustness guarantee. Training cost for RS is omitted as it essentially post-processes standard-trained models (see Appendix E for more details). A † superscript on the number of epochs denotes that an early-stop callback was used to determine convergence. **(b)** Certification timing and memory usage results on CIFAR-10 $(\epsilon = {}^{36}/_{255})$. **(c)** Upper and lower bounds on the global and average local Lipschitz constant. In (a) and (b), peak GPU Memory usage is calculated per-instance by dividing the total measurement by the training or certification batch size.

for toolkit compatibility reasons. Appendix E in the supplementary material provides further details on how memory usage was measured. Because different batch sizes were used to train and evaluate each model, we control for this by reporting the memory used *per instance in each batch*. The cost for standard training is included for comparison. The training cost of RS is omitted, as RS does not use a specialized training procedure, and is thus comparable to standard training. Appendix E provides more information on this point.

We see that KW is the most expensive approach to train, requiring tens to hundreds of seconds per epoch and roughly 35× more memory per batch instance than standard training. BCP is less expensive than KW, but still takes nearly one minute per epoch on MNIST and CIFAR and 15 minutes on Tiny-Imagenet, and uses anywhere between 5-106× more memory than standard training.

Meanwhile, the cost of GloRo Nets is more comparable to that of standard training than of KW or BCP, taking only a few seconds per epoch, and at most 50% more memory than standard training. Because of its memory scalability, we were able to use a larger batch size with GloRo Nets. As a result, more epochs were required during training however, this did not outweigh the significant reduction in time per epoch, as the total time for training was still only at most half of the total time for BCP.

Figure 4b shows the cost of each approach both in the time required to certify the entire test set and in the memory used to do so (results given for CIFAR-10). KW is the most expensive deterministic approach in terms of time and memory, followed by BCP. Here again, GloRo Nets are far superior in terms of cost, making certified predictions over 14× faster than BCP with less than a tenth of the memory, and over 6,000× faster than KW. We thus conclude that GloRo Nets are the most scalable state-of-the-art technique for robustness certification.

As reported in Figure 4a, Randomized Smoothing typically outperforms the VRA achieved by GloRo Nets, and is also inexpensive to train; though the VRA achieved by RS reflects a stochastic guarantee rather than a deterministic one. However, we see in Figure 4b that GloRo Nets are *several orders of magnitude* faster at certification than RS. GloRo Nets perform certification in a single forward pass, enabling certification of the entire CIFAR-10 test set in *under half a second*; on the other hand, RS requires tens of thousands of samples to provide confident guarantees, reducing throughput by orders of magnitude and requiring over *ten hours* to certify the same set of instances.

5.3. Lipschitz Tightness

Theorem 3 demonstrates that a global Lipschitz bound is theoretically sufficient for certifying 2ϵ-separated data. However, as discussed in Section 3, there may be several practical limitations making it difficult to realize a network satisfying Theorem 3; we now assess how these limitations are borne out in practice by examining the Lipschitz bounds that GloRo Nets use for certification.

Weng et al. (2018a) report that an upper bound on the global Lipschitz constant is not capable of certifying robustness for a non-trivial radius. While this is true of models produced via *standard training*, GloRo Nets impose a strong implicit regularization on the global Lipschitz constant. Indeed, Figure 4c shows that the global upper bound is several orders of magnitude smaller on GloRo Nets than on standard networks.

Another potential limitation of using an upper bound of the global Lipschitz constant is the bound itself (Huster et al., 2018). Figure 4c shows that a lower bound of the Global Lipschitz constant, obtained via optimization, reaches an impressive 83% of the upper bound on MNIST, meaning that the upper bound is fairly tight. On CIFAR-10 and Tiny-Imagenet the lower bound reaches approximately 70% and 47% of the upper bound, respectively. However, on a standard model, the lower bound is potentially orders of magnitude looser. These results show there is still room for improvement; for example, using the lower bound in place of the upper bound would lead to roughly a 10% increase in VRA on CIFAR-10, from 58% to 64%. However, the fact that the bound is tighter for GloRo Nets suggests the objective imposed by the GloRo Net helps by incentivizing parameters for which the upper bound estimate is sufficiently tight for verification.

Finally, we compare the global upper bound to an empirical lower bound of the local Lipschitz constant. The local lower bound given in Figure 4c reports the *mean* local Lipschitz constant found via optimization in the ϵ-balls centered at each of the test points. In the construction given for the proof of Theorem 3, the local Lipschitz constant is the same as the global bound at all points. While the results in Figure 4c show that this may not be entirely achieved in practice, the ratio of the local lower bound to the global upper bound is essentially zero in the standard models, compared to 6-35% in the GloRo Nets, establishing that the upper bound is again much tighter for GloRo Nets. Still, this suggests that a reasonably tight estimate of the local bound may yet help improve the VRA of a GloRo Net at runtime, although this is a challenge in its own right. Intriguingly, GloRo Nets outperform BCP, which utilizes a *local* Lipschitz bound for certification at train and test time, suggesting that GloRo Nets provide a better objective for certifiable robustness despite using a looser bound during training.

We provide further discussion of the upper and lower bounds, and details for how the lower bounds were obtained in Appendix F in the supplementary material.

6. Related Work

Utilizing the Lipschitz constant to certify robustness has been studied in several instances of prior work. On discovering the existence of adversarial examples, Szegedy et al. (2014) analyzed the sensitivity of neural networks using a global Lipschitz bound, explaining models' "blind spots" partially in terms of large bounds and suggesting Lipschitz regularization as a potential remedy. Huster et al. (2018) noted the potential limitations of using global bounds computed layer-wise according to Equation 2, and showed experimentally that direct regularization of the Lipschitz constant by penalizing the weight norms of a two-layer network yields subpar results on MNIST. While Theorem 3 does not negate their concern, as it may not always be feasible to compute a tight enough bound using Equation 2, our experimental results show to the contrary that global bounds can suffice to produce models with at least comparable utility to several more expensive and complicated techniques. More recently, Yang et al. (2020) showed that robustness and accuracy need not be at odds on common benchmarks when locally-Lipschitz functions are used, and call for further investigation of methods that impose this condition while promoting generalization. Our results show that globally-Lipschitz functions, which bring several practical benefits (Section 3), are a promising direction as well.

Lipschitz constants have been applied previously for fast post-hoc certification (Weng et al., 2018a; Hein & Andriushchenko, 2017; Weng et al., 2018b). While our work relies on similar techniques, our exclusive use of the global bound means that no additional work is needed at inference time. Additionally, we apply this certification only to networks that have been optimized for it.

There has also been prior work seeking to use Lipschitz bounds, or close analogues, during training to promote robustness (Tsuzuku et al., 2018; Raghunathan et al., 2018; Cisse et al., 2017; Cohen et al., 2019b; Anil et al., 2019; Pauli et al., 2021; Qin et al., 2019; Finlay & Oberman, 2019; Lee et al., 2020; Gouk et al., 2021; Singla & Feizi, 2019; Farnia et al., 2019). Cisse et al. (2017) introduced Perseval networks, which enforce contractive Lipschitz bounds on all layers by orthonormalizing their weights. Anil et al. (2019) proposed replacing ReLU activations with sorting activations to construct a class of *universal Lipschitz approximators*, that that can approximate any Lipschitz-bounded function over a given domain, and Cohen et al. (2019b) subsequently studied the application to robust training; these advances in architecture complement our work, as noted in Appendix D.

The closest work in spirit to ours is Lipschitz Margin Training (LMT) (Tsuzuku et al., 2018), which also uses global Lipschitz bounds to train models that are more certifiably robust. The approach works by constructing a loss that adds $\sqrt{2}\epsilon K_G$ to all logits other than that corresponding to the ground-truth class. Note that this is different from GloRo Nets, which add a *new logit* defined by the *predicted* class at x. In addition to providing different gradients than those of LMT's loss, our approach avoids penalizing logits corresponding to boundaries distant from x. In practical terms, Lee et al. (2020) showed that LMT yields lower verified accuracy than more recent methods that use local Lipschitz bounds (Lee et al., 2020) or dual networks (Wong et al., 2018), while Section 5.1 shows that our approach can provide greater verified accuracy than either. LMT's use of global bounds means its cost is comparable to our approach.

More recently, Lee et al. (2020) explored the possibility of training networks against local Lipschitz bounds, motivated by the fact that the global bound may vastly exceed a typical local bound on some networks. They showed that a localized refinement of the global spectral norm of the network offers a reasonable trade-off of precision for cost, and were able to achieve competitive, and in some cases superior, verified accuracy to prior work. Theorem 3 shows that in principle, the difference in magnitude between local and global bounds may not matter for robust classification. Moreover, while it is true that the bounds computed by Equation 2 may be loose on some models, our experimental results suggest that it is possible in many cases to mitigate this limitation by training against a global bound with the appropriate loss. The advantages of doing so are apparent in the cost of both training and certification, where the additional overhead involved with computing tighter local bounds is an impediment to scalability.

Finally, several other methods have been proposed for training ℓ_2-certifiable networks that are not based on Lipschitz constants. For example, Wong & Kolter (2018) use an LP-based approach that can be optimized relatively efficiently using a *dual network*, Croce et al. (2019) and Madry et al. (2018) propose training routines based on maximizing the size of the linear regions within a network, and Mirman et al. (2018) propose a method based on abstract interpretation.

Randomized Smoothing. The certification methods discussed thus far provide *deterministic* robustness guarantees. By contrast, another recent approach, Randomized Smoothing (Cohen et al., 2019a; Lecuyer et al., 2018), provides *stochastic* guarantees—that is, points are certified as *robust with high probability* (i.e., the probability can be bounded from below). Randomized Smoothing has been found to achieve better VRA performance than any deterministic certification method, including GloRo Nets. However, GloRo Nets compare favorably to Randomized Smoothing in a few key ways.

First, the fact that GloRo Nets provide a deterministic guarantee is an advantage in and of itself. In safety-critical applications, it may not be considered acceptable for a small fraction of adversarial examples to go undetected; meanwhile, Randomized Smoothing is typically evaluated with a false positive rate around 0.1% (Cohen et al., 2019a), meaning that instances of incorrectly-certified points are to be expected in validation sets with thousands of points.

Furthermore, as demonstrated in Section 5.2, GloRo Nets have far superior run-time cost. Because Randomized Smoothing does not explicitly represent the function behind its robust predictions, points must be evaluated and certified using as many as 100,000 samples (Cohen et al., 2019a), reducing throughput by several orders of magnitude. Meanwhile, GloRo Nets can certify a batch of points in *a single forward pass*.

7. Conclusion

In this work, we provide a method for training certifiably-robust neural networks that is simple, fast, memory-efficient, and that yields state-of-the-art deterministic verified accuracy. Our approach is particularly efficient because of its effective use of global Lipschitz bounds, and while we prove that the potential of our approach is in theory not limited by the global Lipschitz constant itself, it remains an open question as to whether our bounds on the Lipschitz constant can be tightened, or if additional training techniques can help unlock its remaining potential. Finally, we note that if instances arise where a global bound is not sufficient in practice, costlier post-hoc certification techniques may be complimentary, as a fall-back.

Acknowledgments.

The work described in this paper has been supported by the Software Engineering Institute under its FFRDC Contract No. FA8702-15-D-0002 with the U.S. Department of Defense, as well as DARPA and the Air Force Research Laboratory under agreement number FA8750-15-2-0277.

References

Anil, C., Lucas, J., and Gross, R. Sorting out Lipschitz function approximation. In *ICML*, 2019.

Balunovic, M. and Vechev, M. Adversarial training and provable defenses: Bridging the gap. In *ICLR*, 2020.

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

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019a.

Cohen, J. E., Huster, T., and Cohen, R. Universal lipschitz approximation in bounded depth neural networks. *arXiv preprint arXiv:1904.04861*, 2019b.

Croce, F., Andriushchenko, M., and Hein, M. Provable robustness of ReLU networks via maximization of linear regions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Farnia, F., Zhang, J., and Tse, D. Generalizable adversarial training via spectral normalization. In *ICLR*, 2019.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. J. Efficient and accurate estimation of lipschitz constants for deep neural networks. In *NIPS*, 2019.

Finlay, C. and Oberman, A. M. Scaleable input gradient regularization for adversarial robustness. *CoRR*, abs/1905.11468, 2019.

Fischetti, M. and Jo, J. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, Jul 2018.

Fromherz, A., Leino, K., Fredrikson, M., Parno, B., and Păsăreanu, C. Fast geometric projections for local robustness certification. In *ICLR*, 2021.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *Symposium on Security and Privacy (S&P)*, 2018.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, Feb 2021.

Gowal, S., Dvijotham, K. D., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NIPS*, 2017.

Huster, T., Chiang, C.-Y. J., and Chadha, R. Limitations of the lipschitz constant as a defense against adversarial examples. In *ECML PKDD*, 2018.

Jordan, M., Lewis, J., and Dimakis, A. G. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. In *NIPS*, 2019.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. 2015.

LeCun, Y., Cortes, C., and Burges, C. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist`.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *Symposium on Security and Privacy (S&P)*, 2018.

Lee, S., Lee, J., and Park, S. Lipschitz-certifiable training with a tight outer bound. In *NIPS*, 2020.

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

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

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., and Edwards, B. Adversarial robustness toolbox v1.0.0, 2019.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *European Symposium on Security and Privacy (EuroS&P)*, 2016.

Pauli, P., Koch, A., Berberich, J., Kohler, P., and Allgower, F. Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 2021.

Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. In *NIPS*, 2019.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. 2018.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. In *ICLR*, 2019.

Singla, S. and Feizi, S. Bounding singular values of convolution layers. *CoRR*, abs/1911.10258, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tjeng, V. and Tedrake, R. Verifying neural networks with mixed integer programming. *CoRR*, abs/1711.07356, 2017.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *NIPS*, 2018.

Weng, T.-W., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Boning, D., Dhillon, I. S., and Daniel, L. Towards fast computation of certified robustness for relu networks. In *ICML*, 2018a.

Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*, 2018b.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *NIPS*, 2018.

Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R. R., and Chaudhuri, K. A closer look at accuracy vs. robustness. In *NIPS*, 2020.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *NIPS*, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

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

Zou, D., Balan, R., and Singh, M. On lipschitz bounds of general convolutional neural networks. *IEEE Transactions on Information Theory*, 66(3):1738–1759, 2019.