The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Robust Training of Neural Networks against Bias Field Perturbations

Patrick Henriksen[1,2], **Alessio Lomuscio**[2]

[1] Imperial College London
[2] Safe Intelligence
patrick.henriksen18@imperial.ac.uk, alessio@safeintelligence.ai

Abstract

We introduce the problem of training neural networks such that they are robust against a class of smooth intensity perturbations modelled by bias fields. We first develop an approach towards this goal based on a state-of-the-art robust training method utilising Interval Bound Propagation (IBP). We analyse the resulting algorithm and observe that IBP often produces very loose bounds for bias field perturbations, which may be detrimental to training. We then propose an alternative approach based on Symbolic Interval Propagation (SIP), which usually results in significantly tighter bounds than IBP. We present ROBNET, a tool implementing these approaches for bias field robust training. In experiments networks trained with the SIP-based approach achieved up to 31% higher certified robustness while also maintaining a better accuracy than networks trained with the IBP approach.

1 Introduction

Neural networks have achieved remarkable performance in a wide range of applications recently; however, they remain fragile to even small perturbations of the networks' inputs (Szegedy et al. 2014). Thus, research into robust training, *i.e.,* training approaches that improve the robustness of neural networks, is of high importance.

Adversarial augmentation-based methods (Zhang and Li 2019) are one class of robust training approaches which augment the training set with adversarial examples (Szegedy et al. 2014) to promote robustness. Adversarial augmentation often produces networks that are empirically more robust (Madry et al. 2017); however, the method does not capture all adversarial inputs for the perturbation under consideration. Therefore, networks trained with adversarial augmentation often exhibit a higher empirical robustness against the adversarial attacks used during training than against different ones.

A further approach to robust training utilises *abstraction-based* methods to over-approximate the reachable output sets for each perturbation region (Zhang et al. 2020a). The output sets are used in loss functions that promote correctness for all perturbations under consideration. In contrast to adversarial training, this approach encapsulates all inputs defined by the perturbation. Moreover, the abstractions used during training can provide formal guarantees on the robustness of the networks; for these reasons, these methods are often referred to as certifiably robust training methods.

While abstraction-based robust training methods have achieved state-of-the-art results, their supported input perturbation types are limited with most methods supporting white noise perturbations only. But additional input perturbations such as contrast, brightness, and other smooth intensity perturbations remain of high interest in practice. This paper focuses on robust training against *bias field* perturbations (Tincher et al. 1993; Vovk, Pernus, and Likar 2007).

Bias field perturbations model a wide range of smooth, spatially varying intensity changes via multiplicative and additive polynomials. For instance, brightness and contrast perturbations are instantiations of 0-order bias field perturbations. While bias field perturbations have extensively been used to model noise in MRI images from imperfections in the magnetic imaging equipment and electrodynamic interactions (Styner et al. 2000), more recent work has employed them to model smooth intensity changes in natural images due to changes in lighting conditions (Henriksen et al. 2021) (see Figure 1). This makes them a useful perturbation to explore in the context of robust training of neural networks. In this paper we make the following contributions towards bias field robust training of neural networks.

- We define bias field robust training and propose a novel encoding such that white noise robust training methods become applicable to bias fields perturbations.
- We provide an analytical evaluation of the leading robust training method, RSIP-IBP, when instantiated on bias field robust training and identify some limitations.
- Based on our analytical findings, we propose a new method, called RSIP-SSIP, which addresses some of the limitations of RSIP-IBP when applied to bias field robust training.
- We implement RSIP-IBP and RSIP-SSIP in a toolkit ROBNET. In experiments networks trained with RSIP-SSIP achieved up to 31% higher certified bias field robustness than networks trained with RSIP-IBP.

Related Work. A significant body of work has considered methods for robust training of neural networks based on augmenting the training set with adversarial examples (Kurakin, Goodfellow, and Bengio 2017; Madry et al. 2017;

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: Bias field perturbed CIFAR10 images. Original image (left), correctly classified (mid), misclassified (right).

Shafahi et al. 2019; Zhang et al. 2020b; Wong, Rice, and Kolter 2020; Andriushchenko and Flammarion 2020) and various robustness-promoting loss functions (Wang et al. 2018a; Zheng, Chen, and Ren 2019; Xiao et al. 2019). While these methods often produce networks with good empirical robustness, they are limited in that they (i) do not consider all adversarial examples for the perturbation under consideration, (ii) do not produce any formal guarantees for the robustness of the network, and (iii) mostly target white noise perturbations of the inputs. In contrast, we here propose a method that (i) uses abstractions which encapsulate all adversarial examples in the perturbation region, (ii) can be used to give formal guarantees on the robustness of the network, and (iii) considers bias field perturbed inputs.

Closely related to our work are the abstraction-based robust training methods (Wong and Kolter 2018; Wong et al. 2018; Dvijotham et al. 2018; Mirman, Gehr, and Vechev 2018; Raghunathan, Steinhardt, and Liang 2018; Balunovic and Vechev 2020; Xu et al. 2020; Zhang et al. 2020a; Palma et al. 2022). These methods use abstractions to calculate overestimating reachable sets for the networks' output which, in turn, are used in loss functions to promote robustness. The abstractions encapsulate all adversarial examples in the perturbation region and produce lower bounds on the robustness. However, most abstraction-based methods are limited to white noise perturbations and no previous work has considered robustness to bias field perturbations.

The state-of-the-art abstraction-based method RSIP-IBP (Zhang et al. 2020a) is particularly relevant to our work. RSIP-IBP uses interval bound propagation (IBP) and reversed symbolic interval propagation (RSIP) to calculate reachable output sets for the networks. In our work, we propose an encoding of bias field perturbations such that RSIP-IBP can directly be applied to bias field robust training. Moreover, we analyse RSIP-IBP in the context of bias field robust training and, based on our findings, propose a novel method RSIP-SSIP. In experiments, we show that RSIP-SSIP significantly outperforms RSIP-IBP for bias field robust training.

Also relevant to this work is the formal verification method for bias field robustness in (Henriksen et al. 2021). While this paper builds on the formalisation of bias field robustness from (Henriksen et al. 2021), this paper deals with robust training, not verification.

2 Preliminaries

In this section we provide definitions for neural network robustness and summarise symbolic interval propagation algorithms and their connection to robust training approaches.

In the following we use $N : \mathbb{R}^n \to \mathbb{R}^m$ to denote a neural network, $\boldsymbol{x} \in R^n$ without superscripts to denote the input to a neural network, and $\boldsymbol{x}_i^j$ to denote the pre-activation value for node i in layer j. For notational convenience, we consider neural networks with 1-dimensional inputs only when the extension to more dimensions is straightforward. For more information about neural networks, we refer to (Goodfellow, Bengio, and Courville 2016).

We here consider the following definition of robustness (Botoeva et al. 2020; Kouvaros and Lomuscio 2021).

Definition 1 (NN-Robustness)**.** *A network $N : \mathbb{R}^n \to \mathbb{R}^m$ is robust wrt some input constraints $\psi_{\boldsymbol{x}}$ and output constraints $\psi_{\boldsymbol{y}}$ if $N(\boldsymbol{x})$ satisfies $\psi_{\boldsymbol{y}}$ for all $\boldsymbol{x}$ satisfying $\psi_{\boldsymbol{x}}$.*

The literature on formal verification and robust training usually considers $\psi_{\boldsymbol{x}}$ to be an ℓ_2 or ℓ_∞ bounded set centred around a concrete input $\boldsymbol{x}$, which we here refer to as white noise robustness. The output constraints $\psi_{\boldsymbol{y}}$ are usually considered to be linear constraints on the network's output. The certified robustness of a network is the number of proven robust inputs in a test set.

Interval Bound Propagation (IBP) (Gowal et al. 2018) is a method for calculating over approximating reachable output sets for neural networks given input sets on the form $X = \{\boldsymbol{x}_i | \boldsymbol{x}_{i,l} \leq \boldsymbol{x}_i \leq \boldsymbol{x}_{i,u}\}$ for $\boldsymbol{x}_{i,l}, \boldsymbol{x}_{i,u} \in \mathbb{R}$. The input bounds are propagated through the network on a layer-by-layer basis. For a fully connected layer i with lower and upper input bounds $\boldsymbol{x}_{:,l}^i, \boldsymbol{x}_{:,u}^i$, weight matrix W^i, and bias $\boldsymbol{b}^i$, the layer's output bounds are computed as $\boldsymbol{x}_{:,l}^{i+1} = W_+^i \boldsymbol{x}_{:,l}^i + W_-^i \boldsymbol{x}_{:,u}^i + \boldsymbol{b}^i$ and $\boldsymbol{x}_{:,u}^{i+1} = W_+^i \boldsymbol{x}_{:,u}^i + W_-^i \boldsymbol{x}_{:,l}^i + \boldsymbol{b}^i$. Here W_+^i and W_-^i are the weight matrices where negative and positive elements have been set to zero, respectively. Monotonously increasing activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ (e.g., ReLU, Sigmoid, Tanh) are handled by applying the activation function element-wise to the bounds.

Standard and Reversed Symbolic Interval Propagation (SSIP and RSIP) are extensions of IBP in which the bounds are represented as linear equations of the network's input variables. In SSIP the bounds are initialised at the network's input layer and propagated to the output layer. In contrast, RSIP initialises bounds at the layer of interest and back-substitutes the bounds' variables to the input layer. Both SIP versions use linear relaxations of activation functions (Wang et al. 2018b; Singh et al. 2018, 2019) that require calculating concrete bounds. In SSIP these bounds are calculated in the forward pass without significant additional computational costs. In RSIP, however, the bound propagation has to be repeated for each activation layer.

The symbolic bounds in SIP implicitly track node dependencies, and do thus usually produce tighter bounds than IBP (Wang et al. 2018c). For more information on SSIP, we refer to (Wang et al. 2018b); for RSIP we refer to (Henriksen and Lomuscio 2021; Singh et al. 2019; Zhang et al. 2018).

Note that RSIP is referred to as "CROWN" in (Zhang et al. 2018) and "DeepPoly" in (Singh et al. 2019).

Certifiable White Noise Robust Training. The field of certifiable robust training considers how to train robust networks as defined in Definition 1. In particular the problem is usually instantiated with $\psi_{\boldsymbol{x}}$ as a ℓ_∞-bounded neighbourhood around the inputs $\boldsymbol{x}$. The robust training problem can then be formalised as follows (Zhang et al. 2020a).

Definition 2 (White Noise Robust Training)**.** *Let $N : \mathbb{R}^n \to \mathbb{R}^m$ be a neural network with parameters θ, X be a training distribution, L be a loss function, and let $\lambda \in \mathbb{R}$. The white noise robust training problem is defined as follows.*

$$\min_{\theta} \underset{(\boldsymbol{x},\boldsymbol{y})\in X}{E} \left[\max_{\boldsymbol{x}' :\, ||\boldsymbol{x}'-\boldsymbol{x}||_\infty \leq \lambda} L(N(\boldsymbol{x}'); \boldsymbol{y}) \right]. \tag{1}$$

Directly evaluating the inner maximisation in the definition above is usually infeasible, thus certified robust training methods use an upper-bounding relaxation instead.

RSIP-IBP (Zhang et al. 2020a) is a state-of-the-art method for certifiable robust training in which the upper bounding relaxation of the inner maximisation in Definition 2 is calculated via IBP and RSIP. The combination of IBP and RSIP does not only reduce the computational cost for RSIP, but recent work has shown that the bounds produced by IBP are beneficial for robust training against ℓ_∞-perturbations (Jovanović et al. 2021). Note that the algorithm was originally named "CROWN-IBP"; however, in the following we will use RSIP-IBP for notational consistency.

Bias Fields. We here consider bias field transformations to be additive and multiplicative spatially varying polynomials. Formally, we define bias fields and bias field transformations as in (Henriksen et al. 2021).

Definition 3 (Bias Field)**.** *A k-th order bias field $B^k(\boldsymbol{a}) \in \mathbb{R}^n$ parametrised by $\boldsymbol{a}$ is defined as $B(\boldsymbol{a}) = \sum_{t=0}^{k} \boldsymbol{a}_t \boldsymbol{b}^t$ for $\boldsymbol{a}_t \in \mathbb{R}$ and $\boldsymbol{b}^t \in \mathbb{R}^n$ where $\boldsymbol{b}^t_i = (i/n)^t$.*

Definition 4 (Bias Field Transformation)**.** *A k-th order bias field transformation $T^k_B : \mathbb{R}^n \to \mathbb{R}^n$ is defined as $T^k_B(\boldsymbol{x}; \boldsymbol{a}^m, \boldsymbol{a}^a) = \boldsymbol{x} \odot B^m(\boldsymbol{a}^m) + B^a(\boldsymbol{a}^a)$ where B^m and B^a are k-th order multiplicative and additive bias fields. Here "$\odot$" denotes the Hadamard product.*

Definition 4 assumes inputs in $\mathbb{R}^n$ for for ease of presentation. However, the extension to $\mathbb{R}^{n\times m}$ is straightforward by considering bias fields on the form $B = \sum_{t=0}^{k}\sum_{g=0}^{k} \boldsymbol{a}_{t,g}\boldsymbol{b}^{t,g}$ for $\boldsymbol{a}_{t,g} \in \mathbb{R}$ and $\boldsymbol{b}^{t,g} \in \mathbb{R}^{n\times m}$ where $\boldsymbol{b}^{t,g}_{i,j} = (i/n)^t(j/m)^g$.

In (Henriksen et al. 2021) robustness to bias field perturbations is defined with respect to ℓ_∞ perturbations of the bias field's coefficients $\boldsymbol{a}^a$, $\boldsymbol{a}^m$, formalised as follows.

Definition 5 (Bias Field Robustness)**.** *Let $N : \mathbb{R}^n \to \mathbb{R}^m$ be a neural network, $\boldsymbol{x} \in \mathbb{R}^n$ be a input, T_B be a bias field transform, and $A^m = \{\bar{\boldsymbol{a}}^m : ||\bar{\boldsymbol{a}}^m - \boldsymbol{a}^m||_\infty < \lambda\}$ and $A^a = \{\bar{\boldsymbol{a}}^a : ||\bar{\boldsymbol{a}}^a - \boldsymbol{a}^a||_\infty < \lambda\}$ with $\boldsymbol{a}^m, \boldsymbol{a}^a \in \mathbb{R}^k, \lambda \in \mathbb{R}$ be sets of parameters for the bias field transform. The network N is bias field robust for $\boldsymbol{x}$, A^m, and A^a if $N(T_B(\boldsymbol{a}^m, \boldsymbol{a}^a; \boldsymbol{x}))$ satisfies $\psi_{\boldsymbol{y}}$ for all $\boldsymbol{a}^a \in A^a$ and $\boldsymbol{a}^m \in A^m$.*

We note that white noise and bias field robustness and robust training can easily be generalised from ℓ_∞-based perturbation sets to box constrained sets on the form $X = \{\boldsymbol{x} \mid \boldsymbol{x}_{i,low} \leq \boldsymbol{x}_i \leq \boldsymbol{x}_{i,up}\}_{\forall i}$ for $\boldsymbol{x}_{i,low}, \boldsymbol{x}_{i,up} \in \mathbb{R}$; this is because box constraints are supported by SIP algorithms. However, we here use ℓ_∞ for ease presentation.

3 Bias Field Transformations for Robust Training

In this section we define the bias field robust training problem and propose a method for reducing the problem to a white noise robust training problem. We then conclude that algorithms for white noise robust training can be applied to the reduced form of the bias field robust training problem.

Building on Definition 5, we here propose the following novel definition of the bias field robust training problem as an optimisation problem with respect to ℓ_∞ perturbations of the $\boldsymbol{a}^m$ and $\boldsymbol{a}^a$ coefficients.

Definition 6 (Bias Field Robust Training)**.** *Let $N : \mathbb{R}^n \to \mathbb{R}^m$ be a neural network with parameters θ, X be a training distribution, L be a loss function, and T_B be a bias field transform. Moreover, let $A^m = \{\bar{\boldsymbol{a}}^m : ||\bar{\boldsymbol{a}}^m - \boldsymbol{a}^m||_\infty < \lambda\}$ and $A^a = \{\bar{\boldsymbol{a}}^a : ||\bar{\boldsymbol{a}}^a - \boldsymbol{a}^a||_\infty < \lambda\}$ with $\boldsymbol{a}^m, \boldsymbol{a}^a \in \mathbb{R}^k, \lambda \in \mathbb{R}$, be sets of parameters for the multiplicative and additive bias fields in the bias field transformation. The bias field robust training problem is defined as the following optimisation problem.*

$$\min_{\theta} \underset{(\boldsymbol{x},\boldsymbol{y})\in X}{E} \left[\max_{\boldsymbol{a}^m, \boldsymbol{a}^a \in A^m, A^a} L(N(T_B(\boldsymbol{a}^m, \boldsymbol{a}^a; \boldsymbol{x})); \boldsymbol{y}) \right]. \tag{2}$$

The optimisation problem in Definition 6 has an inner maximisation on the parameters of the bias field; thus, we aim to minimise the maximal loss for any bias-field perturbation satisfying $\boldsymbol{a}^m \in A^m$ and $\boldsymbol{a}^a \in A^a$. Since these perturbations correspond to the perturbation in Definition 5, we expect a network with a small loss to be robust against bias field perturbations. Later, in Section 6, we will present empirical results supporting this assumption.

Observe that the bias field robust training problem in Definition 6 differs from the white noise robust training problem in Definition 2 in that the input variables to the network in the inner maximisation are perturbed by bias field transformations instead of ℓ_∞-bounded perturbations. Since RSIP-IBP only supports ℓ_∞-perturbations of the input variables, and not bias field transformations, RSIP-IBP cannot be applied directly to calculate the necessary relaxations.

To create a relaxation for the inner maximisation, we extend the method used in (Henriksen et al. 2021) for formal verification against bias field perturbations and apply the result to robust training of bias field networks. In particular, we utilise bias field transform networks.

Definition 7 (Bias Field Transform Network)**.** *Let $N : \mathbb{R}^n \to \mathbb{R}^m$ be a neural network and let T_B be the k-th order bias field transform defined by the bias fields $B^m(\boldsymbol{a}^m) = \sum_{t=0}^{k} \boldsymbol{a}^m_t \boldsymbol{b}^{m,t}$ and $B^a(\boldsymbol{a}) = \sum_{t=0}^{k} \boldsymbol{a}^a_t \boldsymbol{b}^{a,t}$. For a given input $\boldsymbol{x}$, the bias field transform network $N_{T_B,\boldsymbol{x}} : \mathbb{R}^{2k} \to \mathbb{R}^m$*

is given by $N_{T_B,\boldsymbol{x}}(\boldsymbol{a}^m, \boldsymbol{a}^a) = N(W_{T_B,\boldsymbol{x}}[\boldsymbol{a}^m, \boldsymbol{a}^a])$*, where* $W_{T_B,\boldsymbol{x}} = [\boldsymbol{x} \odot \boldsymbol{b}^{m,0}..., \boldsymbol{x} \odot \boldsymbol{b}^{m,k}, \boldsymbol{b}^{a,0}, ..., \boldsymbol{b}^{a,k}]$ *and* $[\boldsymbol{a}^m, \boldsymbol{a}^a]$ *is the concatenation of* $\boldsymbol{a}^m$ *and* $\boldsymbol{a}^a$*.*

The network $N_{T_B,\boldsymbol{x}}(\boldsymbol{a}^m, \boldsymbol{a}^a)$ is the neural network N, with a prepended fully connected layer with weight matrix $W_{T_B,\boldsymbol{x}}$ that encodes the bias field transform T_B for a given $\boldsymbol{x}$ and takes the bias field transform coefficients $\boldsymbol{a}^m, \boldsymbol{a}^a$ as input. Since $N_{T_B,\boldsymbol{x}}$ is a neural network in itself and takes as input the parameters $\boldsymbol{a}^m, \boldsymbol{a}^a$, the bias field robust training problem in Definition 6 can be reduced to a white noise robust training problem, formalised as follows.

Definition 8 (Reduced Bias Field Robust Training)**.** *Let* $N : \mathbb{R}^n \rightarrow \mathbb{R}^m$ *be a neural network with parameters* θ*, X be a training distribution, and* T_B *be a k-th order bias field transform. Moreover, for each* $\boldsymbol{x}$ *in the training distribution let* $N_{T_B,\boldsymbol{x}}$ *be the bias field transform network corresponding to* $\boldsymbol{x}$*,* N *and* T_B*, and let* $A^m = \{\bar{\boldsymbol{a}}^m : ||\bar{\boldsymbol{a}}^m - \boldsymbol{a}^m||_\infty < \lambda\}$ *and* $A^a = \{\bar{\boldsymbol{a}}^a : ||\bar{\boldsymbol{a}}^a - \boldsymbol{a}^a||_\infty < \lambda\}$ *with* $\boldsymbol{a}^m, \boldsymbol{a}^a, \in \mathbb{R}^k, \lambda \in \mathbb{R}$*. The reduced bias field optimisation problem is defined as:*

$$\min_{\theta} \underset{(\boldsymbol{x},y)\in X}{E} \left[\max_{\boldsymbol{a}^m,\boldsymbol{a}^a \in A^m, A^a} L(N_{T_B,\boldsymbol{x}}(\boldsymbol{a}^m, \boldsymbol{a}^a); \boldsymbol{y}) \right]. \quad (3)$$

Theorem 1. *The bias field robust training problem in Definition 6 is equivalent to the reduced bias field training problem in Definition 8.*

Proof. The result follows directly from the fact that

$$\begin{aligned} N_{T_B,\boldsymbol{x}}(\boldsymbol{a}^m, \boldsymbol{a}^a) &= N(W_{T_B,\boldsymbol{x}}[\boldsymbol{a}^m, \boldsymbol{a}^a]) \\ &= N(T_B(\boldsymbol{a}^m, \boldsymbol{a}^a; \boldsymbol{x})). \end{aligned} \quad (4)$$

Here, the first equality is from Definition 7, and the second is from Definition 4.

□

Theorem 1 shows that instead of considering the bias field robust training problem in Definition 6 we can consider the equivalent reduced formulation in Definition 8. Moreover, this reduced formulation is on the same form as the white noise robust training problem defined in Definition 2. Thus, we can directly apply any algorithm that supports white noise robust training problems, such as RSIP-IBP, to the reduced formulation above in order to solve the bias field robust training problem defined in Definition 6.

In summary we here defined the bias field robust training problem and proposed a method for reducing the problem to a white noise robust training problem. We then concluded that state-of-the-art algorithms for white noise robust training, such as RSIP-IBP, can be applied to bias field robust training. In the following section, we will analyse particular properties of the bias field robust training problem and introduce a novel algorithm for robust training targeted specifically at bias field robust training.

4 RSIP-SSIP for Bias Field Robustification

In the previous section we proposed a method for reducing the bias field robust training problem to a white noise robust training problem supported by RSIP-IBP. In this section first we present analytical results showing that the IBP part of RSIP-IBP may not be optimal for bias field robust training due to the looseness of the resulting bounds. Next, we propose a novel algorithm, RSIP-SSIP, which addresses the shortcoming of RSIP-IBP. Finally, we argue that RSIP-SSIP is computationally efficient for low-dimensional input networks such as bias field networks and thus suitable for robust training. In Section 6, we provide empirical evidence substantiating the analytical evaluation in this section and the advantages of RSIP-SSIP over RSIP-IBP for bias field robust training.

The bias field robust training formulation in Definition 8 differs from the white noise robustness training problem in Definition 2 in that the former optimises over the network N_{T_B}, which is the network N with the fully connected layer $T_B(\boldsymbol{a}^m, \boldsymbol{a}^a; \boldsymbol{x})$ appended to the neural network. The network N_{T_B} thus takes $\boldsymbol{a}^m, \boldsymbol{a}^a$ as input which are typically low-dimensional compared to $\boldsymbol{x}$. For example, for a third-order two-dimensional bias field transform, $\boldsymbol{a}^m$ and $\boldsymbol{a}^a$ have 16 elements each, while $\boldsymbol{x}$ typically has at least thousands of elements for computer vision networks.

We might expect networks with low input-dimensionality to have stronger *node dependencies* than those with high input-dimensionality. By strong node dependency, we here mean that knowing the value at some nodes in the network significantly limits the reachable values of other nodes in the network for a given input set. Due to the low input-dimensionality of bias field networks, we might expect that bias field transform networks have a stronger node dependency than the underlying neural network.

The IBP algorithm, which is essential in RSIP-IBP, does not take node-dependencies into account when calculating bounds (Wang et al. 2018c). Thus, from the argument above, we might expect IBP to produce less tight bounds for the bias field network when compared to the underlying network for comparable input sets. Indeed, this is formalised below for brightness perturbations, which is an instantiation of a 0-order additive bias field transformation.

Theorem 2. *Let* $N : \mathbb{R}^n \rightarrow \mathbb{R}^m$ *be a neural network and* $N_{T_B,\boldsymbol{x}} : \mathbb{R} \rightarrow \mathbb{R}^m$ *be the 0-order bias field transform network* $N_{T_B,\boldsymbol{x}}(a) = N(\boldsymbol{x} + a\mathbf{1})$ *where* $a \in \mathbb{R}$ *and* $\mathbf{1} \in \mathbb{R}^n$*. For input sets defined by* $A = \{a : ||a||_\infty < \lambda\}$ *and* $X = \{\boldsymbol{x}' : ||\boldsymbol{x}' - \boldsymbol{x}||_\infty < \lambda\}$*, we have* $IBP(N, X) = IBP(N_{T_B,\boldsymbol{x}}, A)$*, where* $IBP(N, X)$ *and* $IBP(N_{T_B,\boldsymbol{x}}, A)$ *are the bounds produced by IBP at the network's output layer.*

Proof. Note that N and $N_{T_B,\boldsymbol{x}}$ only differ in that $N_{T_B,\boldsymbol{x}}$ has a prepended extra fully connected layer, $T_B(a; \boldsymbol{x}) = \boldsymbol{x} + a\mathbf{1}$. It is enough to show that $IBP(T_B(a; \boldsymbol{x})) = \{\boldsymbol{x}' : ||\boldsymbol{x}' - \boldsymbol{x}||_\infty < \lambda\}$. The lower bounds of $IBP(T_B(a; \boldsymbol{x}))$ are:

$$\begin{aligned} \boldsymbol{x}_{:,l}^{T_B} &= W_+ a_l + W_- a_u + \boldsymbol{x} \\ &= \mathbf{1}(-\lambda) + \mathbf{0} a_u + \boldsymbol{x} \\ &= \boldsymbol{x} - \mathbf{1}\lambda. \end{aligned} \quad (5)$$

Similar calculations for the upper bounds gives us $\boldsymbol{x}_{:,u}^{T_B} = \boldsymbol{x} + \mathbf{1}\lambda$, thus $IBP(T_B(a; \boldsymbol{x})) = \{\boldsymbol{x}' : ||\boldsymbol{x}' - \boldsymbol{x}||_\infty < \lambda\} = X$. □

Algorithm 1: RSIP-SSIP

Input: $N, A, L_{std}, L_{rob}, \alpha$, TrainingData, NumEpochs

for epoch $\leftarrow$ 1 to NumEpochs **do**
  for $\boldsymbol{x}, l$ in TrainingData **do**
    $N_{T_B,\boldsymbol{x}} \leftarrow BiasFieldNetwork(N, \boldsymbol{x})$
    $\psi_{ssip} \leftarrow SSIP(N_{T_B,\boldsymbol{x}}, A)$
    $\bar{L}_{rob} \leftarrow RSIP(N_{T_B,\boldsymbol{x}}, \psi_{ssip}, A, L_{rob}, l)$
    L $\leftarrow L_{std}(N(\boldsymbol{x}), l) + \alpha \bar{L}_{rob}$
    $Optimise$(L, N.parameters)

Theorem 2 shows that IBP produces the same bounds for N given the input set $X = \{\boldsymbol{x}' : ||\boldsymbol{x}' - \boldsymbol{x}||_\infty < \lambda\}$ as for $N_{T_B,\boldsymbol{x}}$ given $A = \{a : ||a||_\infty < \lambda\}$. However, the brightness transform wrt A, *i.e.,* $\{\boldsymbol{x}' \mid \boldsymbol{x}' = \boldsymbol{x} + a, a \in A\}$, is clearly a subset of X, thus we would expect the bounds for $N_{T_B,\boldsymbol{x}}$ given A to be tighter than the bounds for N given X. As discussed above, the loose bounds are a result of IBP not taking node dependencies into account.

Similar arguments may be made for bias fields of higher-order; thus we can expect IBP to produce relatively loose bounds for bias field networks. We note that recent studies have shown that tighter bounds do not necessarily correlate with better performance in robust training (Jovanović et al. 2021). However, very loose bounds may still be detrimental to robust training. Indeed in Section 6 we show that IBP-based training quickly deteriorates for larger perturbations.

Considering the discussion above, it is clear that bias field robust training may benefit from replacing IBP in RSIP-IBP with a relaxation that better accounts for node dependencies. We here introduce **RSIP-SSIP** exactly to that effect. RSIP-SSIP differs from RSIP-IBP in that it utilises SSIP, as described in Section 2, to calculate the bounds for the intermediate layers in the network as opposed to IBP. Compared to IBP, SSIP generally produces tighter bounds as it keeps track of some of the node dependencies (Wang et al. 2018b). The full RSIP-SSIP algorithm is provided in Algorithm 1.

Algorithm 1 takes as input the network N, the set of parameters for the bias field perturbation A, the standard loss function L_{std}, the robustness loss L_{rob}, a weighting parameter α, the training data, and the number of training epochs. L_{rob} may be the loss function L_{std} under the worst-case perturbation; however, in practice L_{rob} is often adapted to facilitate bound calculation.

The inner training loop first initialises the bias field transform network $N_{T_B,\boldsymbol{x}}$. Next, the SSIP bounds are calculated for all layers in the network (ψ_{ssip}), followed by the RSIP upper bound for the robustness loss ($\bar{L}_{rob}$). Note that RSIP utilises the bounds from SSIP, thus back-propagating the gradients through the RSIP upper bound also back-propagates the gradients through the SSIP bounds.

Finally, the standard training loss is calculated with respect to the output of a forward pass and total loss is a weighted sum of the robust loss and standard loss. The parameters of the network are optimised with respect to this loss via back-propagation. We will comment on the implementation details of the Algorithm, such as the specific loss functions used, in the next section.

Computational Complexity. We end this section by deriving the computational complexity of RSIP-SSIP. We do not consider the concretisation of symbolic bounds as the cost is usually dominated by bound propagation.

Theorem 3. *The computational complexity of a forward pass off RSIP-SSIP is $\mathcal{O}((n_b + 2n_i + 3)F)$ where F is the computational complexity of a standard forward pass in the network, n_i is the the network's input dimensionality and n_b are the number of bounds calculated by RSIP.*

Proof. The computational complexity of the RSIP phase is $\mathcal{O}(n_b F)$ (Zhang et al. 2020a). SSIP keeps track of separate lower and upper symbolic bounds and each bound has n_i coefficients and one constant value. Each coefficient and constant value of the SSIP bounds has the same number of operations as a standard forward pass; thus the complexity of SSIP is $\mathcal{O}(2(n_i + 1)F)$. Finally RSIP-SSIP also performs a standard forward pass to calculate the standard training loss, giving us the total complexity $\mathcal{O}(2(n_i + 1)F) + \mathcal{O}(n_b F) + \mathcal{O}(F) = \mathcal{O}((n_b + 2n_i + 3)F)$

□

We note that n_b is typically small; e.g., for classification networks we usually calculate only one bound for each of the network's outputs (see Section 5). Moreover, for bias field transform networks, the input dimension is also typically relatively small (e.g., $n_i = 16$ for a 3-order 2-dimensional multiplicative bias field transformation network). In experiments (Section 6) we recorded training times per epoch for RSIP-SSIP to be only 8–9 times longer than for standard training with a medium-sized CIFAR10 network. The increase in training time may be an acceptable trade-off for increased robustness in many applications, especially when the applications are safety-critical.

In summary we (i) analysed the tightness of bounds calculated by IBP for bias field transform networks and concluded that they may be too loose for efficient robust training, (ii) we proposed a novel algorithm RSIP-SSIP specifically targeted at bias field transform networks, and (iii) we showed that RSIP-SSIP is computationally efficient for low dimensional networks. In the following section we will cover some implementation details of the algorithm proposed here and in Section 6 we empirically show that RSIP-SSIP produces significantly more robust and accurate networks than RSIP-IBP for bias field transform networks.

5 Implementation

In this section we present ROBNET, a toolkit implementing RSIP-IBP and RSIP-SSIP as described in the previous sections. In the rest of this paper, we consider only image classification networks for ease of presentation, but the material can suitably be adapted to other problems.

ROBNET is implemented in Python and uses the PyTorch library for vectorised computations, neural network training, and GPU acceleration. As in Algorithm 1, ROBNET takes as input a neural network N, the bias field constraints A, the training distribution X, the number of training epochs E, the loss-weighting parameter α, and a learning rate.

ROBNET uses stochastic gradient descent for optimising the networks' parameters, thus each iteration in the training loop considers a batch of inputs. For each input in the batch, a new bias field transform network is created. We note, however, that all parameters in the networks are shared, except parameters in the networks' transform layers.

The bias field transform layers may result in pixel values being perturbed out of the $[0, 1]$ range. To avoid this effect, we augment the network with pixel-clipping layers directly after the transform layer. The clipping layers are on the form $-(ReLU(-ReLU(\boldsymbol{x}) + 1) - 1)$ and ensure that each element is clipped to the $[0, 1]$ range.

For each bias field network, ROBNET uses IBP (for RSIP-IBP) or SSIP (for RSIP-SSIP) to calculate the bounds for all non-linear nodes in the network. In the case of RSIP-SSIP, IBP is first used in a pre-processing step; this is followed by SSIP; the tightest bounds obtained from either IBP or SSIP are used to calculate the relaxations for the non-linear operations. This combination is guaranteed to produce tighter bounds than either method in isolation.

The bounds calculated with IBP and SSIP are, in turn, used to calculate relaxations for all non-linear operations in the network; the relaxations are utilised in RSIP to calculate an upper bound for the following robustness loss.

$$L_{Rob}(N_{T_B,\boldsymbol{x}}, A; l) = \sum_{i \neq l} \max_{\boldsymbol{a} \in A}(\max(0, N_{T_B,\boldsymbol{x}}(\boldsymbol{a})_i - N_{T_B,\boldsymbol{x}}(\boldsymbol{a})_l)). \quad (6)$$

Here $\boldsymbol{x}$ is the network's input, l is the corresponding label, and $N_{T_B,\boldsymbol{x}}(\boldsymbol{a})_i$ is the output for class i in the neural network. The hinge loss from the inner maximisation ensures that provably robust inputs do not contribute to the loss. Moreover, we do not calculate the robustness loss for inputs that are misclassified (*i.e.,* $N(\boldsymbol{x})_l \leq N(\boldsymbol{x})_i$ for some $i \neq l$). We found that both of these design choices improved the performance of the training procedure.

ROBNET optimises over a weighted sum of the robust and standard loss:

$$L(\boldsymbol{x}, N, A; l) = L_{std}(N(\boldsymbol{x}); l) + \alpha \bar{L}_{Rob}(N_{T_B,\boldsymbol{x}}, A; l). \quad (7)$$

Here $\bar{L}_{Rob}$ is the upper bound for L_{Rob} as calculated by RSIP, L_{std} is the standard cross-entropy loss, and $\alpha \in \mathbb{R}$ is a weighting factor. ROBNET uses the PyTorch implementation of the Adam optimiser (Kingma and Ba 2015) to optimise over the loss. The RSIP and SSIP implementations are based on the SIP implementation from the open-source VeriNet toolkit (Henriksen and Lomuscio 2020). However, we significantly modified the implementations to support calculating bounds for batches of inputs in parallel with GPU acceleration. We found batch computations and GPU acceleration to be essential for efficient training.

6 Experimental Evaluation

This section presents experimental results for bias field robust training with ROBNET. As discussed in the introduction, bias field perturbations can model changes in lighting conditions for natural images and inhomogeneities for MRI images; we here represent these two applications with the CIFAR10 dataset (Krizhevsky, Nair, and Hinton 2014) and an MRI brain tumour classification dataset (Parvar 2021), respectively. The CIFAR10 dataset contains ten classes of natural images and is extensively used in experimental evaluations of robust training methods. The MRI dataset consists of 7023 2-dimensional slices of brain scans where the task is to classify each scan as one of four tumour classes.

All experiments were run on a workstation with Fedora 35, Linux kernel 5.15, NVIDIA RTX 3090 GPU, 256 GB Ram, and an AMD Threadripper 3970X 32-Core Processor. The networks were first trained for 200 epochs without robust loss, of which 80 epochs used a learning rate of 10^{-3}, 80 used 10^{-4} and 40 used 10^{-5}. After the initial training, the networks were trained for 100 epochs with a learning rate of 5×10^{-4}, while the perturbation radius was evenly increased from 0 to the radii under consideration. Finally, we trained 50 epochs with a learning rate of 5×10^{-5}, 25 epochs with 5×10^{-5} and 25 epochs with 5×10^{-6}. The learning rates for robust training ($5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}$) are the same as in (Zhang et al. 2020a).

In line with most research in the field, we here consider both the test accuracy and certified robustness of the networks. Typically, there is a trade-off between these metrics, *i.e.,* one can be increased at the cost of reducing the other. Thus, both metrics should be taken into consideration when evaluating the performance of the robust training algorithm.

We here consider robustness to third-order multiplicative bias field perturbations. The coefficient of the bias fields' constant term is constrained to $[1 - \varepsilon, 1 + \varepsilon]$, and the remaining terms are constrained to $[-\varepsilon/15, \varepsilon/15]$ as in (Henriksen et al. 2021). The certified robustness was evaluated with RSIP-SSIP over all data points in the test set. We note that certified robustness evaluated by RSIP-SSIP is always better than RSIP-IBP as RSIP-SSIP bounds are guaranteed to be tighter than RSIP-IBP bounds (see Section 5).

Perturbation radii and loss-weights. Table 1 shows the ROBNET results for a CIFAR10 network with the DM-Medium architecture from (Zhang et al. 2020a) (7 layers, 62k ReLU nodes, and 2.5m parameters) with various perturbation radii and loss-weights. The Table contains the results for the standard (pre-robust training) networks, the RSIP-IBP networks and the RSIP-SSIP networks.

For all ε (perturbation radii) and α (weight of the robust loss), the test accuracy and certified robustness for the RSIP-SSIP networks are better than the corresponding RSIP-IBP networks. For small ε, the accuracy and robustness of networks trained with RSIP-IBP are still comparable to RSIP-SSIP; however, the RSIP-IBP results quickly deteriorate for larger ε . We also note that RSIP-SSIP achieves 31% higher certified robustness than RSIP-IBP with $\varepsilon = 40$ and their best α values ($\alpha = 4$ and $\alpha = 1$, respectively).

The average training time for the standard network was 30 seconds per epoch. For RSIP-IBP, the training time ranged from 97 to 137, and for RSIP-SSIP from 253 to 273 seconds, depending on ε and α. We note that training time generally decreases slightly for larger ε; the decrease is mainly

	Test Accuracy			Cert. Robustness		
ε	Std %	RI %	RS %	Std %	RI %	RS %
1	82.9	78.1	**81.6**	65.7	77.8	**81.3**
5	82.9	72.1	**81.0**	41.2	71.3	**78.5**
10	82.9	66.8	**79.6**	5.7	64.9	**75.1**
20	82.9	54.8	**78.7**	0.0	49.6	**70.2**
40	82.9	46.3	**76.6**	0.0	35.2	**62.9**
α	Std %	RI %	RS %	Std %	RI %	RS %
0.25	82.9	49.0	**76.7**	0.0	33.0	**58.6**
0.50	82.9	45.0	**77.2**	0.0	33.2	**61.0**
1.00	82.9	46.3	**76.6**	0.0	35.2	**62.9**
2.00	82.9	44.9	**76.7**	0.0	35.0	**64.3**
4.00	82.9	41.8	**76.6**	0.0	30.6	**66.1**

Table 1: Robust training for a medium-sized CIFAR10 classification model. "Std" is the standard (not robustly trained), "RI" the RSIP-IBP, and "RS" the RSIP-SSIP trained model. The top five rows use $\alpha = 1$, the bottom five use $\varepsilon = 40$.

	Test Accuracy			Cert. Robustness		
ε	Std %	RI %	RS %	Std %	RI %	RS %
SC	75.6	51.2	**72.5**	9.45	41.2	**56.4**
MC	82.9	46.3	**76.6**	0.0	35.2	**62.9**
LC	85.9	44.8	**79.2**	0.0	34.3	**65.9**

Table 2: Robust training with $\alpha = 1$ and $\varepsilon = 40$ for a small (SC), medium (MC), and large (LC) CIFAR10 network.

because larger perturbations result in a lower accuracy, and RobNet does not calculate the robust loss for misclassified points (see Section 5).

Network architectures. In Table 2 we present the RobNet results for the CIFAR-10 DM-Small and DM-large architectures from (Zhang et al. 2020a). DM-Small has 4 layers, 6k ReLU nodes and 215k parameters, while DM-large has 7 layers, 230k ReLU nodes, and 17m parameters. For all architectures, RSIP-SSIP produces networks with significantly better accuracy and certified robustness than RSIP-IBP, and the difference increases for larger networks.

The average training time per epoch for DM-Small was 29 seconds with standard training, 99 with RSIP-IBP and 138 with RSIP-SSIP. Corresponding numbers for DM-large were 31, 149, and 703 seconds.

MRI classification experiments. In Table 3 we report the robust training results for the MRI brain tumour classification network (9 layers, 311k ReLU nodes and 17m parameters). The network takes MRI slices with one channel and 128×128 pixels as input.

In these experiments RSIP-IBP slightly outperformed RSIP-SSIP for $\varepsilon = 1$; however, for larger perturbations, RSIP-SSIP significantly outperforms RSIP-IBP both on accuracy and certified robustness.

The average training time per epoch for standard training was around 5 seconds, 59–63 seconds for RSIP-IBP, and 136–145 seconds in RSIP-SSIP. We note that the relative difference in training time for RSIP-SSIP compared to RSIP-IBP was larger in these experiments. The difference in training time is mainly due to the larger input dimension, which increases the size of the transform and clipping layers used in robust training.

	Test Accuracy			Cert. Robustness		
ε	Std %	RI %	RS %	Std %	RI %	RS %
1	97.8	**98.2**	97.6	97.7	**98.1**	97.5
5	97.8	97.1	**97.9**	89.2	95.6	**96.6**
10	97.8	97.2	**97.9**	17.4	92.1	**96.1**
20	97.8	95.3	**98.0**	1.4	71.2	**94.0**
40	97.8	87.4	**97.4**	0.0	44.3	**96.6**

Table 3: Robust training for an MRI classification network.

Summary. The experiments show that RSIP-IBP generally outperformed RSIP-SSIP for a range of perturbation radii, network architectures and loss-weights on the CIFAR10 and brain tumour MRI datasets. For small perturbation radii, the results for RSIP-IBP and RSIP-SSIP were generally comparable; however, for larger radii, RSIP-SSIP-trained networks were significantly more robust and accurate than RSIP-IBP-trained networks. The performance difference is in line with our hypothesis from Section 4; IBP bounds may become too loose to provide meaningful information during training for larger networks.

The training times for RSIP-SSIP were only 2–4.5 times larger for RSIP-SSIP than for RSIP-IBP; this increase is expected given the complexity remarks offered in Section 4. We note that the training time further increased somewhat for larger networks; we hypothesise that this is because RSIP-SSIP reached maximum GPU utilisation for smaller networks than RSIP-IBP did, while RSIP-IBP may comparatively utilise GPUs more efficiently for larger networks.

7 Conclusions

Research into robust training of neural networks remains of high importance to facilitate widespread adoption of networks in safety-critical applications. While much progress has been made lately, current approaches for robust training are limited in that they mainly consider robustness against white noise perturbations. In contrast, we here considered robust training against a class of smooth spatially varying intensity perturbations modelled via bias fields.

We proposed a method for encoding bias field perturbations into the neural networks via transformation layers and showed that the resulting networks can be combined with white noise robust training methods to perform bias field robust training. We analysed the state-of-the-art method RSIP-IBP in the context of bias field robust training and identified some shortcomings of the method. We addressed these shortcomings by proposing a novel method, RSIP-SSIP, for robust training and implemented both RSIP-SSIP and RSIP-IBP in a toolkit RobNet. In experiments RobNet produced networks with up to 31% higher certified robustness against bias field perturbations when using RSIP-SSIP compared to RSIP-IBP.

Acknowledgements

This work was partly supported by the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (EP/S023356/1).

References

Andriushchenko, M.; and Flammarion, N. 2020. Understanding and Improving Fast Adversarial Training. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS20)*, 16048–16059. Curran Associates, Inc.

Balunovic, M.; and Vechev, M. 2020. Adversarial training and provable defenses: Bridging the gap. In *Proceedings of the 8th International Conference on Learning Representations (ICLR20)*. OpenReview.net.

Botoeva, E.; Kouvaros, P.; Kronqvist, J.; Lomuscio, A.; and Misener, R. 2020. Efficient Verification of Neural Networks via Dependency Analysis. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI20)*, 3291–3299. AAAI Press.

Dvijotham, K.; Gowal, S.; Stanforth, R.; Arandjelovic, R.; O'Donoghue, B.; Uesato, J.; and Kohli, P. 2018. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*.

Goodfellow, A.; Bengio, Y.; and Courville, A. 2016. *Deep learning*, volume 1. MIT press Cambridge.

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

Henriksen, P.; Hammernik, K.; Rueckert, D.; and Lomuscio, A. 2021. Bias Field Robustness Verification of Large Neural Image Classifiers. In *Proceedings of the 32nd British Machine Vision Conference (BMVC21)*. BMVA Press.

Henriksen, P.; and Lomuscio, A. 2020. Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI20)*, 2513–2520. IOS Press.

Henriksen, P.; and Lomuscio, A. 2021. DEEPSPLIT: an Efficient Splitting Method for Neural Network Verification via Indirect Effect Analysis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 2549–2555. ijcai.org.

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

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR15)*.

Kouvaros, P.; and Lomuscio, A. 2021. Towards Scalable Complete Verification of ReLU Neural Networks via Dependency-based Branching. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 2643–2650. ijcai.org.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2014. The CIFAR-10 dataset. http://www.cs.toronto.edu/kriz/cifar.html. Accessed: 2022-06-01.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. In *Proceedings of the 5th International Conference on Learning Representations (ICLR17)*. OpenReview.net.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mirman, M.; Gehr, T.; and Vechev, M. 2018. Differentiable abstract interpretation for provably robust neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML18)*, 3578–3586. Omnipress.

Palma, A. D.; Bunel, R.; Dvijotham, K.; Kumar, M. P.; and Stanforth, R. 2022. IBP Regularization for Verified Adversarial Robustness via Branch-and-Bound. *arXiv preprint arXiv:2206.14772*.

Parvar, M. 2021. Brain Tumor MRI Dataset. https://www.kaggle.com/dsv/2645886. Accessed: 2022-06-01.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified Defenses against Adversarial Examples. In *Proceedings of the 5th International Conference on Learning Representations (ICLR19)*. OpenReview.net.

Shafahi, A.; Najibi, M.; Ghiasi, M.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS19)*, volume 32. Curran Associates, Inc.

Singh, G.; Gehr, T.; Püschel, M.; and Vechev, M. 2019. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL): 41.

Singh, G.; Gehr, T.; Mirman, M.; Püschel, M.; and Vechev, M. 2018. Fast and Effective Robustness Certification. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS18)*, 10802–10813. Curran Associates, Inc.

Styner, M.; Brechbuhler, C.; Szckely, G.; and Gerig, G. 2000. Parametric estimate of intensity inhomogeneities applied to MRI. *IEEE Transactions on Medical Imaging*, 19(3): 153–165.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR14)*.

Tincher, M.; Meyer, C. R.; Gupta, R.; and Williams, D. M. 1993. Polynomial modeling and reduction of RF body coil spatial inhomogeneity in MRI. *IEEE Transactions on Medical Imaging*, 12(2): 361–365.

Vovk, U.; Pernus, F.; and Likar, B. 2007. A Review of Methods for Correction of Intensity Inhomogeneity in MRI. *IEEE Transactions on Medical Imaging*, 26(3): 405–421.

Wang, S.; Chen, Y.; Abdou, A.; and Jana, S. 2018a. Mixtrain: Scalable training of verifiably robust neural networks. *arXiv preprint arXiv:1811.02625*.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018b. Efficient Formal Safety Analysis of Neural Networks. In *Proceedings on Advances in Neural Information Processing Systems (NeurIPS18)*, 6367–6377. Curran Associates, Inc.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018c. Formal Security Analysis of Neural Networks using Symbolic Intervals. In *Proceedings of the 27th USENIX Security Symposium (USENIX18)*.

Wong, E.; and Kolter, Z. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning (ICML18)*, 5286–5295. PMLR.

Wong, E.; Rice, L.; and Kolter, J. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*.

Wong, E.; Schmidt, F.; Metzen, J.; and Kolter, J. 2018. Scaling provable adversarial defenses. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS18)*.

Xiao, K.; Tjeng, V.; Shafiullah, N.; and Madry, A. 2019. Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability. In *Proceedings of the 7th International Conference on Learning Representations (ICLR19)*, 1–20. OpenReview.net.

Xu, K.; Shi, Z.; Zhang, H.; Wang, Y.; Chang, K.-W.; Huang, M.; Kailkhura, B.; Lin, X.; and Hsieh, C.-J. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS20)*, 1129–1141. Curran Associates, Inc.

Zhang, H.; Chen, H.; Xiao, C.; Gowal, S.; Stanforth, R.; Li, B.; Boning, D.; and Hsieh, C. 2020a. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR20)*. OpenReview.net.

Zhang, H.; Weng, T.; Chen, P.; Hsieh, C.; and Daniel, L. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems 2018 (NeurIPS18)*, 4944–4953.

Zhang, J.; and Li, C. 2019. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7): 2578–2593.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020b. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *Proceedings of the 37th International Conference on Machine Learning (ICML20)*, 11278–11287. PMLR.

Zheng, T.; Chen, C.; and Ren, K. 2019. Distributionally adversarial attack. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI19)*, 2253–2260. AAAI Press.