

EFFECTIVE CERTIFICATION OF MONOTONE DEEP EQUILIBRIUM MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Monotone Operator Equilibrium Models (monDEQs) represent a class of models that combine the powerful deep equilibrium paradigm with convergence guarantees. As monDEQs are inherently robust to adversarial perturbations, investigating new methods to certify their robustness is a promising research direction. Unfortunately, existing certification approaches are either imprecise or severely limited in their scalability. In this work, we propose the first scalable *and* precise monDEQ verifier, based on two key ideas: (i) a novel convex relaxation which enables efficient inclusion checks, and (ii) non-trivial mathematical insights characterizing the fixpoint operations at the heart of monDEQs on sets rather than concrete inputs. An extensive evaluation of our verifier demonstrates that on the challenging ℓ_∞ perturbations it exceeds state-of-the-art performance in terms of speed (two orders of magnitude) and scalability (an order of magnitude) while yielding 25% higher certified accuracies on the same networks.

1 INTRODUCTION

Deep Equilibrium Models (DEQ) (Bai et al., 2019) and in particular Monotone Operator Equilibrium Models (monDEQs) (Winston & Kolter, 2020) are promising new architectures, based on implicit layers solving fixpoint problems at inference time. Their competitive accuracies *and* high inherent empirical robustness make an investigation of their certifiable robustness a promising research direction. Initial investigations, via bounds on their Lipschitz constant (Pabbaraju et al., 2021) or semidefinite programming (SDP) (Chen et al., 2021), indicate promising trends. However, in challenging settings such as ℓ_∞ -perturbations, Lipschitz constants are inherently loose and SDP encodings are prohibitively costly. Further, existing scalable verification approaches based on the propagation of convex sets, successful for feed forward networks (Singh et al., 2018; Xu et al., 2020), are unable to handle the implicit layers underlying (mon)DEQs and hence not applicable. A key challenge then is to design precise certification techniques able to handle fixpoint problems on convex sets.

This work: certification for monDEQs In this work, we address this challenge and introduce a new convex relaxation, called M-Zonotope, which, for the first time, allows an efficient and precise propagation of convex sets in the monDEQ setting, overcoming key limitations of prior approaches. Our new relaxation is based on the Zonotope approximation (Mirman et al., 2018; Wong & Kolter, 2018; Singh et al., 2018) which was successfully used to verify properties of (fixed-depth) classic neural networks. However, unlike Zonotopes, our relaxation allows for efficient inclusion checks, critical for handling implicit layers (which require unbounded fixed point iteration). Further, we ensure that our method is correct via rigorous mathematical analysis of iteration process.

Main Contributions Our key contributions are:

- We lift neural equilibrium models (with few conditions), such as monDEQs, from concrete points to sets, enabling their efficient and precise verification (Section 4).
- CRAFT, an verification algorithm, using this set formulation and a new convex relaxation, called M-Zonotope, enabling efficient fixpoint iteration and inclusion checks (Section 5).
- Our extensive evaluation shows that CRAFT achieves a new state-of-the-art for monDEQ verification, outperforming current approaches in precision (25%), speed (two orders of magnitude) and scalability (one order of magnitude) (Section 6).

2 RELATED WORK

We now briefly review related work in neural network verification.

Incomplete Neural Network Verification Incomplete verification approaches (such as ours), unlike complete methods (Katz et al., 2019; Bunel et al., 2020), are fast and efficient but can be imprecise, i.e., may fail to certify robustness for inputs even if they are robust. These methods can be divided into bound propagation approaches (Gehr et al., 2018; Zhang et al., 2018) and those that generate optimization problems (Singh et al., 2019a; Raghunathan et al., 2018) such as linear programming (LP) or semidefinite programming (SDP) formulations. However, existing approaches cannot be applied to (mon)DEQs without non-trivial extensions as they are unable to express the underlying fixpoint problems. Even when extended in the way we discuss in Section 4, precise bound propagation methods lack computationally feasible inclusion checks. Alternatively, stochastic defenses like randomized smoothing Lecuyer et al. (2018); Cohen et al. (2019) establish robustness with high probability but incur significant runtime costs at inference time, a problem further exacerbated by the relatively expensive fixpoint iterations needed in (mon)DEQ inference.

Certification of monDEQs Two main approaches have been proposed to certify the robustness of monDEQs: (i) Pabbaraju et al. (2021) use the special structure of monDEQs to bound the global Lipschitz constant of the network, and (ii) Chen et al. (2021) adapt an SDP approach by introducing a semi-algebraic representation of the ReLU-operator used in monDEQs. Depending on the encoding, the latter allows to bound the score difference between classes, the global Lipschitz constant, or yields an ellipsoidal relationship between inputs and outputs. All three approaches only scale to an implicit layer size of 87 neurons due to the limitations of the underlying SDP solver. Additionally, the most effective approach suffers from long runtimes (1350s/per sample) even for these small networks, making the certification of many inputs or larger networks infeasible. Orthogonally, Revay et al. (2020) show a way of bounding the Lipschitz constant of a monDEQ by construction. However, enforcing small Lipschitz constants this way reduces the resulting accuracy significantly, limiting the utility of the obtained networks.

3 MONOTONE OPERATOR EQUILIBRIUM MODELS ON POINTS

We now briefly discuss the background for (monotone) deep equilibrium models on concrete points before extending them to sets of points in Section 4 and discussing their verification in Section 5.

Deep-Equilibrium Models (DEQs) Implicit-Layer (Amos & Kolter, 2017; Ghaoui et al., 2019) and Deep-Equilibrium-Models (DEQ) (Bai et al., 2019) were recently introduced to enable more memory efficient model parameterizations. Unlike traditional deep neural networks, which propagate inputs through a finite number of different layers, DEQs, conceptually, apply the same layer(s) repeatedly until convergence to a fixpoint. This corresponds to an infinite depth model with parameter-sharing. A DEQ h obtains its final prediction y by applying a linear layer to this fixpoint:

$$y = h(x) := Vz^* + v, \quad z^* = f(x, z^*). \quad (1)$$

In practice, fixpoint solvers are used to compute the fixpoint z^* of layer f for some input x instead of repeatedly propagating the sample. Gradients can be backpropagated through the fixpoint directly using the implicit fixpoint theorem without requiring the solver iterations to be unrolled. As shown in Bai et al. (2019), this constitutes a powerful architecture, achieving almost state-of-the-art performance on text and vision tasks using significantly less memory.

Monotone Operator Equilibrium Models (monDEQs) A major drawback of general DEQs is that they do not guarantee the existence or uniqueness of fixpoints. Hence, solvers may not converge to the same fixpoints consistently, rendering their verification infeasible. In response to this, monDEQs were introduced in Winston & Kolter (2020) as a particular form of DEQs. Parametrizing

$$f(x, z) = \sigma(Wz + Ux + b) \quad (2)$$

with $\mathbf{x} \in \mathbb{R}^q, \mathbf{z} \in \mathbb{R}^p, \mathbf{U} \in \mathbb{R}^{p \times q}, \mathbf{W} = (1 - m)\mathbf{I} - \mathbf{P}^T \mathbf{P} + \mathbf{Q} - \mathbf{Q}^T$ where $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{p \times p}$ and monotonicity parameter $m > 0$, they show that the existence and uniqueness of the fixpoint $\mathbf{z}^*$ is guaranteed. These properties allow certification that is independent of how a fixpoint was obtained, yielding far stronger guarantees than possible in the general DEQ setting. Throughout this text we will use $\sigma := \text{ReLU}$ and discuss considerations for other choices in App. A.1.

Further, Winston & Kolter (2020) derive convergence guarantees for the following iterative solver strategies to this unique fixpoint. For a monDEQ h with iteration function $\mathbf{f}(\mathbf{x}, \mathbf{z}) = \text{ReLU}(\mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x} + \mathbf{b})$, we let $\mathbf{g}_\alpha$ denote an iteration of a solver using operator splitting:

- **Forward-Backward Splitting (FB)** with

$$\mathbf{z}_{n+1} = \mathbf{g}_\alpha^{\text{FB}}(\mathbf{x}, \mathbf{z}_n) := \text{ReLU}((1 - \alpha)\mathbf{z}_n + \alpha(\mathbf{W}\mathbf{z}_n + \mathbf{U}\mathbf{x} + \mathbf{b})), \quad (3)$$

which converges to $\mathbf{z}^*$ of $\mathbf{f}$, for any $0 < \alpha < \frac{2m}{\|\mathbf{I} - \mathbf{W}\|_2^2}$.

- **Peaceman-Rachford Splitting (PR)** with $[\mathbf{z}_{n+1}, \mathbf{u}_{n+1}] = \mathbf{g}_\alpha^{\text{PR}}(\mathbf{x}, \mathbf{z}_n, \mathbf{u}_n)$, where

$$\begin{aligned} \mathbf{u}_{n+1/2} &= 2\mathbf{z}_n - \mathbf{u}_n \\ \mathbf{z}_{n+1/2} &= (\mathbf{I} + \alpha(\mathbf{I} - \mathbf{W}))^{-1}(\mathbf{u}_{n+1/2} + \alpha(\mathbf{U}\mathbf{x} + \mathbf{b})) \\ \mathbf{u}_{n+1} &= 2\mathbf{z}_{n+1/2} - \mathbf{u}_{n+1/2} \\ \mathbf{z}_{n+1} &= \text{ReLU}(\mathbf{u}_{n+1}) \\ [\mathbf{z}_{n+1}, \mathbf{u}_{n+1}] &= \mathbf{g}_\alpha^{\text{PR}}(\mathbf{x}, \mathbf{z}_n, \mathbf{u}_n) \end{aligned} \quad (4)$$

which converges to $\mathbf{z}^*$ for any $\alpha > 0$ (Ryu & Boyd, 2016) by computing auxiliary $\mathbf{u}_n, \mathbf{u}_{n+1/2}, \mathbf{z}_{n+1/2}$, initialized by $\mathbf{u}_0 = \mathbf{0}$. Here $[\mathbf{z}_{n+1}, \mathbf{u}_{n+1}]$ denotes the concatenation of $\mathbf{z}_{n+1}$ and $\mathbf{u}_{n+1}$.

For both algorithms we initialize $\mathbf{z}_0 = \mathbf{0}$. In the following, we will write $\mathbf{g}_\alpha(\mathbf{x}, \mathbf{z}_n, \mathbf{u}_n)$ to refer to both PR and FB, for which we assume $\mathbf{u}_n$ to be zero dimensional.

4 MONOTONE OPERATOR EQUILIBRIUM MODELS ON SETS

In this section, we discuss the mathematics of replacing the concrete inputs $\mathbf{x}$ and intermediate states $\mathbf{z}$ with sets $\mathcal{X}$ and $\mathcal{Z}$ of concrete values and over-approximations $\hat{\mathcal{X}}$ and $\hat{\mathcal{Z}}$ thereof. We will then show how this produces a set containing all concrete fixpoints, used for verification in Section 5.

Fixpoint Iteration Both Forward-Backward (FB, see Eq. (3)) and Peaceman-Rachford splitting (PR, see Eq. (4)) are guaranteed to converge to the fixpoint of $\mathbf{f}(\mathbf{x}, \mathbf{z})$, as defined in Eq. (1), for which we write $\mathbf{z}^*(\mathbf{x})$ to highlight the dependence on $\mathbf{x}$. That is, when iterated until the quantity $\mathbf{z}_n - \mathbf{z}_{n-1}$ becomes smaller than a predetermined stopping criterion, both yield $\mathbf{z}_n \approx \mathbf{z}^*(\mathbf{x})$.

We lift this idea to sets of points: let $\mathcal{X} \subseteq \mathbb{R}^q$ denote a set of inputs, $\mathcal{Z}_n \subseteq \mathbb{R}^p$, $\mathcal{U}_n \subseteq \mathbb{R}^p$ the corresponding intermediate values at step n and $\mathcal{Z}^* := \{\mathbf{z}^*(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ the corresponding fixpoints. Similarly, we lift a function $\mathbf{y} = \mathbf{h}(\mathbf{x})$ operating on concrete points $\mathbf{x}, \mathbf{y}$ to sets $\mathcal{Y} = \mathbf{h}(\mathcal{X}) := \{\mathbf{h}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}$ and let $\mathbf{h}^\#$ denote over-approximation $\mathcal{Y} \subseteq \hat{\mathcal{Y}} = \mathbf{h}^\#(\mathcal{X})$.

Unlike the concrete case, here we want to iterate until $\mathcal{Z}_n$ contains the set of all fixpoints $\mathcal{Z}^*$. Fortunately, as we will show, it is sufficient to iterate the solver procedure on sets $\mathcal{Z}_n$ or their over-approximations $\hat{\mathcal{Z}}_n \supseteq \mathcal{Z}_n$ in order to find

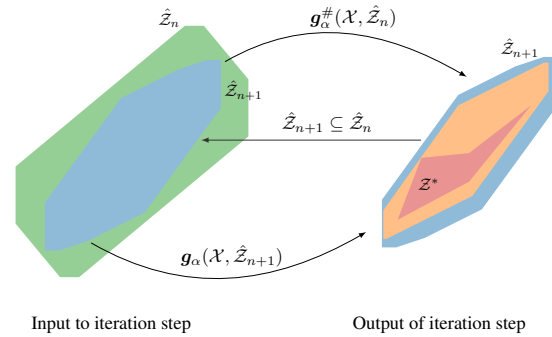


Figure 1: After an over-approximated solver iteration $\hat{\mathcal{Z}}_{n+1} = \mathbf{g}_\alpha^\#(\mathcal{X}, \hat{\mathcal{Z}}_n)$ (blue) is contained in the previous state $\hat{\mathcal{Z}}_n$ (green). This implies that any (exact) iteration of $\mathbf{g}_\alpha(\mathcal{X}, \hat{\mathcal{Z}}_{n+1})$ (orange) will not escape from $\hat{\mathcal{Z}}_{n+1}$. This implies containment of the true fixpoint set (red) $\mathcal{Z}^* \subseteq \hat{\mathcal{Z}}_{n+1}$.

$\hat{\mathcal{Z}}^* \supseteq \mathcal{Z}^*$. We formalize this in the following theorem, where we consider a monDEQ h with iteration function $f(x, z)$ and corresponding solver iteration $[z_{n+1}, u_{n+1}] = g_\alpha(x, z_n, u_n)$ and write $[\hat{\mathcal{Z}}_{n+1}, \hat{\mathcal{U}}_{n+1}]$ for the set containing vectors $[z_{n+1}, u_{n+1}]$.

Theorem 4.1 (Fixed-Point contraction). *Let $[\hat{\mathcal{Z}}_{n+1}, \hat{\mathcal{U}}_{n+1}] = g_\alpha^\#(\mathcal{X}, \hat{\mathcal{Z}}_n, \hat{\mathcal{U}}_n)$ be closed sets over-approximating z_{n+1} and u_{n+1} obtained by applying the solver iteration $n+1$ times for some z_0, u_0 and all inputs $x \in \mathcal{X}$. Then:*

$$\hat{\mathcal{Z}}_{n+1} \subseteq \hat{\mathcal{Z}}_n \wedge \hat{\mathcal{U}}_{n+1} \subseteq \hat{\mathcal{U}}_n \implies \mathcal{Z}_j \subseteq \hat{\mathcal{Z}}_{n+1} \forall j > n \implies \mathcal{Z}^* \subseteq \hat{\mathcal{Z}}_{n+1} \quad (5)$$

As illustrated in Fig. 1, once an iteration maps $\hat{\mathcal{Z}}_n$ to a subset of itself $\hat{\mathcal{Z}}_{n+1} \subseteq \hat{\mathcal{Z}}_n$, no further application of g_α will escape this set and hence it must contain the true fixpoint set $\mathcal{Z}^*$. Note that this does not necessarily hold for further applications of the over-approximated $g_\alpha^\#$.

Proof. Let $\hat{\mathcal{S}}_i := [\hat{\mathcal{Z}}_i, \hat{\mathcal{U}}_i]$ and $\mathcal{S}_i := [\mathcal{Z}_i, \mathcal{U}_i]$. Then

$$\mathcal{S}_{n+1} \subseteq \hat{\mathcal{S}}_{n+1} \subseteq \hat{\mathcal{S}}_n \quad (6)$$

where the first $\subseteq$ holds by definition and the second by the left hand side in Eq. (5). Now, we can over-approximate $\mathcal{S}_{n+1}$ with $\hat{\mathcal{S}}_n \supseteq \hat{\mathcal{S}}_{n+1} \supseteq \mathcal{S}_{n+1}$ and $\mathcal{S}_{n+2} \subseteq \hat{\mathcal{S}}_{n+1}$ follows immediately via Eq. (6). Then $\mathcal{S}_j \subseteq \hat{\mathcal{S}}_{n+1}$ for $j > n$ follows by induction and thereby $\mathcal{Z}_j \subseteq \hat{\mathcal{Z}}_{n+1} \forall j > n$.

To prove the second implication, we note that by the convergence guarantee of the concrete iteration: for any $\epsilon \in \mathbb{R}^{>0}$ there exists a $j \in \mathbb{N}$ with $j \geq n+1$ such that we have $\|z_j - z^*\| \leq \epsilon$. By the definition of $\mathcal{Z}_j$ we also have $z_j \in \mathcal{Z}_j$. For $\epsilon \rightarrow 0$ and hence $\|z_j - z^*\| \rightarrow 0$ it follows that $z^* \in \mathcal{Z}_j \cup \partial \mathcal{Z}_j = \mathcal{Z}_j$ with the last equality following from the closedness of $\mathcal{Z}_j$. Thus for each $x \in \mathcal{X}$, there exists a j_x such that $z^*(x) \in \mathcal{Z}_{j_x} \subseteq \hat{\mathcal{Z}}_{n+1}$, where we get the inclusion relation from the first implication. Finally, $\mathcal{Z}^* \subseteq \bigcup_{x \in \mathcal{X}} \mathcal{Z}_{j_x} \subseteq \hat{\mathcal{Z}}_{n+1}$. $\square$

Fig. 1 tells us that if we consistently apply $g_\alpha^\#$ until containment we know that we capture $\mathcal{Z}^*$. However, while practically unlikely, it does not guarantee that when applying a step of a different iteration function $g_\alpha^\#$ this fixpoint set is preserved. To formally ensure this, we use the specifics of the Forward-Backward solver:

Theorem 4.2 (Fixpoint set preservation for Forward-Backward splitting). *Let $\hat{\mathcal{Z}}_n$ be a sound over-approximation of $\mathcal{Z}^*$, i.e., $\mathcal{Z}^* \subseteq \hat{\mathcal{Z}}_n$. Then we have for $0 < \alpha < \frac{2m}{\|I - W\|_2^2}$:*

$$\mathcal{Z}^* \subseteq \hat{\mathcal{Z}}_{n+1} = g_\alpha^{FB\#}(\mathcal{X}, \hat{\mathcal{Z}}_n) \quad (7)$$

Intuitively, Forward-Backward splitting maps all concrete fixpoints onto themselves and hence any over-approximation of the fixpoint set will map to another over-approximation of the fixpoint set. In contrast to Theorem 4.1, Theorem 4.2 does not assume that the same iterative solver (including hyperparameters) is applied at each step. Instead, it makes a statement about one application of Forward-Backward splitting using any parameters. We will rely on this in Section 5.2 to apply Forward-Backward splitting after applying Peaceman-Rachford splitting and optimize hyperparameters in the course of an iteration.

We discuss a similar result to Theorem 4.2 for Peaceman-Rachford splitting in App. A.

Proof. For any concrete fixpoint $z^* = f(x, z^*) = \text{ReLU}(Wz^* + Ux + b)$, we consider an iteration of Forward-Backward splitting as per Eq. (3) with $z_n = z^*$:

$$z_{n+1} = \text{ReLU}((1 - \alpha)z^* + \underbrace{\alpha(Wz^* + Ux + b)}_{z'}) = z^*$$

We show this by considering the expression element-wise. Suppose $z' \leq 0$, then due to $z^* = f(x, z^*) = \text{ReLU}(z')$, we know $z^* = 0$ and else $z^* = z'$. Then

$$\text{ReLU}((1 - \alpha)z^* + \alpha z') = \begin{cases} \text{ReLU}((1 - \alpha)0 + \alpha z') & \text{if } z' \leq 0 \\ \text{ReLU}((1 - \alpha)z^* + \alpha z^*) & \text{else} \end{cases} = z^*.$$

In the first case we know $(1 - \alpha)0 + \alpha z' \leq 0$ as $z' \leq 0$. It follows that one step of Forward-Backward splitting will always map a fixpoint upon itself in the concrete $z_{n+1} = z_n = z^*$. Since $\hat{\mathcal{Z}}_n$ includes all fixpoints for $\mathcal{X}$, any sound $\hat{\mathcal{Z}}_{n+1} = g_\alpha^\#(\mathcal{X}, \hat{\mathcal{Z}}_n)$ includes all fixpoints for $\mathcal{X}$. $\square$

5 FIXPOINT SET ITERATION WITH M-ZONOTOPE

To utilize Theorem 4.1 for the verification of monDEQs, we need to be able to represent the over-approximations $(\hat{\mathcal{Z}}, \hat{\mathcal{U}})$ in a way that permits efficient containment checks. Of the commonly used convex relaxations in neural network verification, only Boxes meet this requirement. However, as we will show experimentally in Section 6.2, they are too imprecise for practical verification. For more precise relaxations, such as Zonotopes, the inclusion check is computationally prohibitively expensive (Kulmburg & Althoff, 2021). To overcome these issues, we introduce a new variant of the Zonotope relaxation called *Mixed-Zonotope* (M-Zonotope), discussed in Section 5.1. In Section 5.2 we introduce CRAFT, our novel verifier for monDEQs, enabled by the M-Zonotope relaxation.

5.1 M-ZONOTOPE

We denote a M-Zonotope $\hat{\mathcal{Z}} \subseteq \mathbb{R}^p$ over-approximating a volume $\mathcal{Z} \subseteq \hat{\mathcal{Z}}$:

$$\hat{\mathcal{Z}} = \mathbf{A}\boldsymbol{\nu} + \text{diag}(\mathbf{b})\boldsymbol{\eta} + \mathbf{a} \quad (8)$$

where $\mathbf{A} \in \mathbb{R}^{p \times k}$ is the so-called error coefficient matrix, $\mathbf{b} \in (\mathbb{R}^{\geq 0})^p$ the Box error vector, $\mathbf{a} \in \mathbb{R}^p$ the centre vector, and $\boldsymbol{\nu} \in [-1, 1]^k$ and $\boldsymbol{\eta} \in [-1, 1]^p$ the error terms. If $k = p$ and $\mathbf{A}$ is a basis of $\mathbb{R}^p$, we call $\hat{\mathcal{Z}}$ a proper M-Zonotope and else an improper one.

What enables efficient inclusion checks (discussed shortly) is that an M-Zonotope is proper. While a standard (proper) Zonotope with at most n error terms ($\mathbf{b} = \mathbf{0}$) and any Box approximation ($\mathbf{A} = \mathbf{0}$) can apply a similar inclusion check, an M-Zonotope yields much tighter abstraction than either, as it can effectively employ twice as many error terms. We visualize this in Fig. 2.

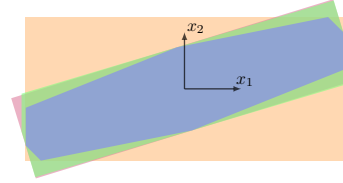


Figure 2: Over-approximations of an improper M-Zonotope (blue) by a proper one with (green) and without (red) Box component and Box (orange).

Transformations We handle affine transformations of an M-Zonotope as introduced in Singh et al. (2018), by casting the Box errors as Zonotope errors and setting $\mathbf{A}' = [\mathbf{A}, \text{diag}(\mathbf{b})]$ before applying the transformation. This turns a proper M-Zonotope with non-zero Box component into an improper one with a zero Box component. To encode the ReLU function for M-Zonotope, we also follow Singh et al. (2018), by adding new error terms as a Box component. Applying this transformer turns a proper M-Zonotope with zero Box component into a proper one with a non-zero Box component. Both of these operations are sound. However, while affine transformations can be encoded exactly, ReLUs result in over-approximations.

Consolidating the Error Terms We now discuss how an improper M-Zonotope $\hat{\mathcal{Z}}$ with a not necessarily full rank $\mathbf{A} \in \mathbb{R}^{p \times k}$ can be over-approximated with a proper M-Zonotope $\hat{\mathcal{Z}}'$ with a base $\mathbf{A}' \in \mathbb{R}^{p \times p}$. If $k > p$ we consolidate the k old error terms into p new ones, ensuring that the resulting $\mathbf{A}'$ has full rank and hence forms a basis of $\mathbb{R}^p$. If $k \leq p$, we pick a subset with full rank and complete it to a basis. In monDEQ certification $p = \dim(\mathbf{z})$ is the size of the latent dimension.

The theorem below shows how an improper M-Zonotope with an arbitrary number of error terms can be over-approximated as a proper M-Zonotope.

Theorem 5.1 (Consolidating errors). *Let $\hat{\mathcal{Z}} = \mathbf{A}\boldsymbol{\nu} + \text{diag}(\mathbf{b})\boldsymbol{\eta} + \mathbf{a}$ be an improper M-Zonotope with $\mathbf{A} \in \mathbb{R}^{p \times k}$ and $k > p$. Let further $\tilde{\mathbf{A}} \in \mathbb{R}^{p \times p}$ be a basis of $\mathbb{R}^p$. Then the proper M-Zonotope $\hat{\mathcal{Z}}' = \mathbf{A}'\mathbf{e}_1 + \text{diag}(\mathbf{b})\boldsymbol{\eta} + \mathbf{a}$ with*

$$\mathbf{A}' = \text{diag}(\mathbf{c})\tilde{\mathbf{A}} \quad \text{where } \mathbf{c} = (|\tilde{\mathbf{A}}^{-1}\mathbf{A}|\mathbf{1}) \quad (9)$$

is a sound over-approximation $\hat{\mathcal{Z}}' \supseteq \hat{\mathcal{Z}}$ of the improper one, where $\mathbf{1} = [1]^k$ denotes the k dimensional one vector and $|\cdot|$ the elementwise absolute. We call $\mathbf{c}$ the consolidation coefficients.

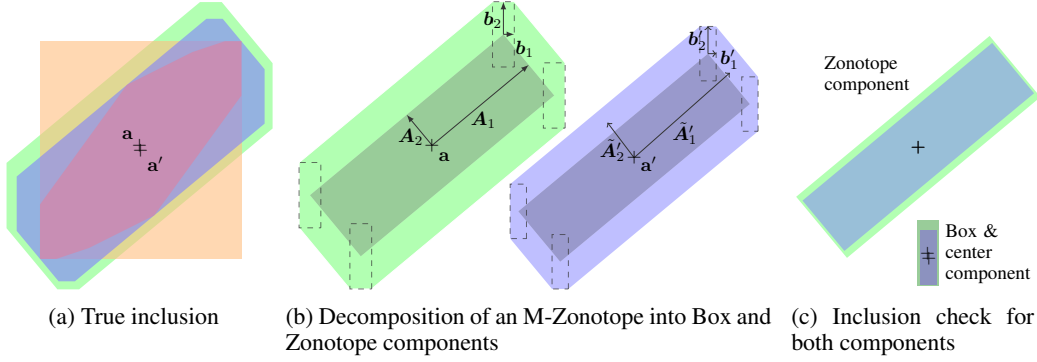


Figure 3: Illustration of checking the containment of an improper M-Zonotope (red) in a proper M-Zonotope (green), by consolidating errors (blue), or concretizing to Box (orange). In Fig. 3b we show how the proper M-Zonotope can be decomposed into their Box and Zonotope components. In Fig. 3c we illustrate the containment check of these components individually.

The intuition behind this is shown in Fig. 4. All error vectors (columns) in $\mathbf{A}$ (shown as red and blue solid arrows) are represented as a linear combination of error vectors in $\tilde{\mathbf{A}}$ (red and blue dashed arrows). Now we sum the absolute values (correcting for their orientation) of these contributions $\tilde{\mathbf{A}}^{-1}\mathbf{A}$ over all error vectors to obtain the consolidation coefficients c . Finally we multiply the obtained contributions with the error vectors of the new basis vectors $\tilde{\mathbf{A}}'_{:,i} = c_i \tilde{\mathbf{A}}_{:,i}$ (solid black arrows).

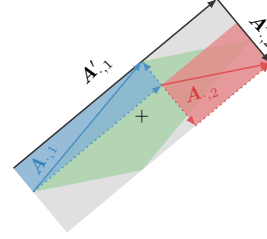


Figure 4: Illustration of Theorem 5.1. The solid red and blue vectors span the Zonotope (green) and can be decomposed into the dotted vectors, which are then consolidated into the solid black arrows. All vectors are scaled by factor 2.

Proof. Without loss of generality let $\mathbf{a} = \mathbf{0}$, $\mathbf{b} = \mathbf{0}$ and $\hat{\mathbf{Z}} = \mathbf{A}\nu = \sum_{j=1}^k \mathbf{A}_j \nu_j$ with k error terms, stored in the columns of $\mathbf{A}$. We can express the contribution of every error term as $\mathbf{A}_j \nu_j = \tilde{\mathbf{A}} \tilde{\nu}^{(j)}$ with $\tilde{\nu}^{(j)} = \tilde{\mathbf{A}}^{-1} \mathbf{A}_j \nu_j$ as $\tilde{\mathbf{A}}$ is a basis of $\mathbb{R}^p$ and hence invertible. From $\nu_j \in [-1, 1]$ it follows that $\tilde{\nu}^{(j)} \in \text{diag}(\tilde{\mathbf{A}}^{-1} \mathbf{A}_j) \tilde{\nu}^{(j)}$ with $\tilde{\nu}^{(j)} \in [-1, 1]^p$. This allows us to rewrite

$$\hat{\mathbf{Z}} = \left\{ \tilde{\mathbf{A}} \sum_{j=1}^k \tilde{\mathbf{A}}^{-1} \mathbf{A}_j \nu_j \right\} \subseteq \left\{ \tilde{\mathbf{A}} \sum_{j=1}^k \text{diag}(\tilde{\mathbf{A}}^{-1} \mathbf{A}_j) \tilde{\nu}^{(j)} \right\} = \left\{ \tilde{\mathbf{A}} \text{diag}(|\tilde{\mathbf{A}}^{-1} \mathbf{A}| \mathbf{1}) \tilde{\nu} \right\} = \hat{\mathbf{Z}}',$$

where the last equality follows from linearity and the choice $\tilde{\nu}_j = \pm \text{sign}(\tilde{\mathbf{A}}^{-1} \mathbf{A}_j)$. $\square$

Choosing the New Basis To minimize the imprecision incurred when consolidating error terms, a suitable basis $\tilde{\mathbf{A}}$ has to be chosen. Fortunately, in monDEQ verification we repeat the same operations in each solver iteration leading to M-Zonotope where many error terms are well-aligned. In particular, they will be stretched mainly by the matrix $\mathbf{W}$ with only an additive component of $\mathbf{U}\mathbf{x}$ and some rescaling in the ReLU transformer. To compute a new basis $\tilde{\mathbf{A}}$ for the error terms of $\hat{\mathbf{Z}}$, we compute the PCA-basis of $\mathbf{A}$. If $k < p$, we complete the basis with orthogonal vectors.

Containment of M-Zonotope Here, we outline the steps of an inclusion check between M-Zonotopes. We illustrate this in Fig. 3, with a proper M-Zonotope $\hat{\mathbf{Z}} = \mathbf{A}\nu + \text{diag}(\mathbf{b})\eta + \mathbf{a}$ (green in Fig. 3a) and the improper M-Zonotope $\hat{\mathbf{Z}}' = \mathbf{A}'\nu' + \text{diag}(\mathbf{b}')\eta' + \mathbf{a}'$ (red in Fig. 3a). To check containment $\hat{\mathbf{Z}}' \subseteq \hat{\mathbf{Z}}$, we decompose both M-Zonotopes into their Zonotope, Box and center components (shown in Fig. 3b) and show that the Box $\mathbf{b}$ and Zonotope $\mathbf{A}$ terms contain their respective counterparts (shown in Fig. 3c). If the centers are not aligned, $\mathbf{a} \neq \mathbf{a}'$, we check containment of the translated Boxes (also shown in Fig. 3c). In case the Box component $\mathbf{b}$ of $\hat{\mathbf{Z}}$ is not sufficient to contain the center difference as well as $\mathbf{b}'$, we allow the Zonotope terms $\mathbf{A}$ to compensate for the remaining center and Box components (not shown).

To determine containment of the Zonotope component, we consolidate the error matrix $\mathbf{A}'$ with basis $\mathbf{A}$. In Fig. 3a this is shown in blue. This leads to perfectly aligned error vectors, enabling us to directly compare the length of corresponding error terms and show containment if those of $\hat{\mathbf{Z}}'$ are all shorter than the equivalent ones in $\hat{\mathbf{Z}}$ (shown overlayed in Fig. 3c). More efficiently, we only compute the consolidation coefficients and check $|\mathbf{A}^{-1}\mathbf{A}'|\mathbf{1} < \mathbf{1}$.

To show containment of the non-negative Box components, we can simply check that $\mathbf{b} \leq \mathbf{b}'$. However, we observe that negative values in the difference vector $\mathbf{b}' - \mathbf{b}$ denote directions in which $\mathbf{b}$ is larger than $\mathbf{b}'$ and can hence compensate for differences in the center terms $\mathbf{a}' - \mathbf{a}$. Positive values in $\mathbf{b}' - \mathbf{b}$ denote directions in which $\mathbf{b}$ is too small to cover $\mathbf{b}'$. Combining these two, we can derive a residual Box component $\mathbf{d} = \max(0, |\mathbf{a}' - \mathbf{a}| + \mathbf{b}' - \mathbf{b})$, that additionally needs to be covered by the Zonotope component. To this end, we can cast $\mathbf{d}$ as additional error terms of $\mathbf{A}'$ and update the Zonotope inclusion check to $|\mathbf{A}^{-1}\mathbf{A}'|\mathbf{1} + |\mathbf{A}^{-1}\text{diag}(\mathbf{d})|\mathbf{1} < \mathbf{1}$. This compensation is not necessary in Fig. 3. We formally express this containment check as:

Theorem 5.2 (M-Zonotope Containment). *Let $\hat{\mathbf{Z}} = \mathbf{A}\boldsymbol{\nu} + \text{diag}(\mathbf{b})\boldsymbol{\eta} + \mathbf{a}$ be a proper M-Zonotope and $\hat{\mathbf{Z}}' = \mathbf{A}'\boldsymbol{\nu}' + \text{diag}(\mathbf{b}')\boldsymbol{\eta}' + \mathbf{a}'$ an improper one. $\hat{\mathbf{Z}}'$ is contained in $\hat{\mathbf{Z}}$ if*

$$|\mathbf{A}^{-1}\mathbf{A}'|\mathbf{1} + |\mathbf{A}^{-1}\text{diag}(\max(0, |\mathbf{a}' - \mathbf{a}| + \mathbf{b}' - \mathbf{b}))|\mathbf{1} \leq \mathbf{1}. \quad (10)$$

holds element-wise. $\mathbf{A}^{-1}$ must exist as $\hat{\mathbf{Z}}$ is proper and therefore $\mathbf{A} \in \mathbb{R}^{p \times p}$ is a basis of $\mathbb{R}^p$.

Proof. Containment is equivalent to showing that for all error terms $\boldsymbol{\nu}' \in [-1, 1]^k, \boldsymbol{\eta}' \in [-1, 1]^p$ describing points in $\hat{\mathbf{Z}}'$, there exist $\boldsymbol{\nu} \in [-1, 1]^p, \boldsymbol{\eta} \in [-1, 1]^p$ of $\hat{\mathbf{Z}}$ such that:

$$\mathbf{A}\boldsymbol{\nu} + \text{diag}(\mathbf{b})\boldsymbol{\eta} + \mathbf{a} = \mathbf{A}'\boldsymbol{\nu}' + \text{diag}(\mathbf{b}')\boldsymbol{\eta}' + \mathbf{a}'.$$

We subtract $\mathbf{a}$ from both sides and over-approximate the right hand side by increasing the Box size by the absolute center difference $|\mathbf{a}' - \mathbf{a}|$ yielding $\mathbf{b}'' := \mathbf{b}' + |\mathbf{a}' - \mathbf{a}|$. This leaves us to show that we can find $\boldsymbol{\nu}, \boldsymbol{\eta}$ such that $\mathbf{A}\boldsymbol{\nu} + \text{diag}(\mathbf{b})\boldsymbol{\eta} = \mathbf{A}'\boldsymbol{\nu}' + \text{diag}(\mathbf{b}'')\boldsymbol{\eta}''$ holds for all $\boldsymbol{\nu}', \boldsymbol{\eta}''$.

We choose $\boldsymbol{\eta} \in [-1, 1]^p$ such that $\text{diag}(\mathbf{b})\boldsymbol{\eta} = \text{sign}(\boldsymbol{\eta}'') \min(\mathbf{b}, \text{diag}(\mathbf{b}'')|\boldsymbol{\eta}''|)$, and obtain

$$\begin{aligned} \mathbf{A}\boldsymbol{\nu} &= \mathbf{A}'\boldsymbol{\nu}' + \max(0, \text{diag}(\mathbf{b}'')\boldsymbol{\eta}'' - \mathbf{b}) \\ \boldsymbol{\nu} &= \mathbf{A}^{-1}\mathbf{A}'\boldsymbol{\nu}' + \mathbf{A}^{-1}\max(0, \text{diag}(\mathbf{b}'')\boldsymbol{\eta}'' - \mathbf{b}) \\ &\stackrel{(*)}{\leq} |\mathbf{A}^{-1}\mathbf{A}'|\mathbf{1} + |\mathbf{A}^{-1}\text{diag}(\max(0, \mathbf{b}'' - \mathbf{b}))|\mathbf{1} \stackrel{(**)}{\leq} \mathbf{1} \end{aligned}$$

where in $(*)$ we use the relation shown in Theorem 5.1 for the sound representation of a decomposition and the fact that setting $\boldsymbol{\eta}''$ to a one vector maximizes $\text{diag}(\mathbf{b}'')\boldsymbol{\eta}''$ and $(**)$ follows from Eq. (10). Taking the absolute value we obtain $|\boldsymbol{\nu}| \leq \mathbf{1}$ and have shown that both $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ exist. $\square$

Theorem 5.2 postulates a sufficient but not necessary condition. This becomes apparent by the over-approximation in the proof, the fact that our Box component does not compensate the Zonotope component and the over-approximation due to error consolidation.

5.2 MONDEQ CERTIFICATION BY FIXPOINT SET ITERATION

Equipped with the building blocks discussed so far, we now introduce CRAFT (short for **C**onvex **R**elaxation **A**bstract **F**ixpoint **I**teration). On a high level, given an input $\mathbf{x}$, target label t , radius ϵ and

Algorithm 1: CRAFT

Input: $\mathbf{x}, \epsilon$, target t , monDEQ $\mathbf{h}$

Output: whether $\forall \mathbf{x}' \in \mathcal{X} \arg \max_i \mathbf{h}(\mathbf{x}')_i = t$

```

1  $\mathcal{X} \leftarrow \{\mathbf{x}' \mid \|\mathbf{x} - \mathbf{x}'\|_\infty \leq \epsilon\}$ 
2  $\hat{\mathbf{Z}}_0 \leftarrow \{\mathbf{z}^*(\mathbf{x})\}, \hat{\mathcal{U}}_0 \leftarrow \{\mathbf{z}^*(\mathbf{x})\}$ 
3 converged  $\leftarrow$  false
4 for  $n \leftarrow 0, 1, \dots, n_{max}$  do
5   if  $\neg$ converged
6      $[\hat{\mathbf{Z}}_n, \hat{\mathcal{U}}_n] \leftarrow \text{consolidate}([\hat{\mathbf{Z}}_n, \hat{\mathcal{U}}_n])$ 
7      $[\hat{\mathbf{Z}}_{n+1}, \hat{\mathcal{U}}_{n+1}] = \mathbf{g}_\alpha^{PR}(\mathcal{X}, \hat{\mathbf{Z}}_n, \hat{\mathcal{U}}_n)$ 
8     converged  $\leftarrow [\hat{\mathbf{Z}}_{n+1}, \hat{\mathcal{U}}_{n+1}] \subseteq [\hat{\mathbf{Z}}_n, \hat{\mathcal{U}}_n]$ 
9   else
10     $\hat{\mathbf{Z}}_{n+1} = \mathbf{g}_\alpha^{FB\#}(\mathcal{X}, \hat{\mathbf{Z}}_n)$ 
11     $\hat{\mathcal{Y}} \leftarrow \mathbf{V}\hat{\mathbf{Z}}_n + \mathbf{v}$ 
12    if  $\hat{\mathcal{Y}}_t - \hat{\mathcal{Y}}_i > 0 \forall i \neq t$ 
13      return true
14 return false
```

Table 1: Overview over the obtained natural accuracy (*Acc.*), adversarial accuracy (*Bound*), the number of samples for which the fixpoint set iteration converged (*Conv.*), the certified accuracy (*Cert.*), and the average time per sample for 100 samples.

<i>Dataset</i>	<i>Model</i>	<i>Latent Size</i>	<i># Acc.</i>	ϵ	<i># Bound</i>	<i># Conv.</i>	<i># Cert.</i>	<i>Time [s]</i>
MNIST	FCx40	40	99	0.05	70	100	36	17.2
	FCx87	87	99	0.05	75	100	30	15.8
	FCx100	100	96	0.05	73	100	23	11.3
	FCx200	200	99	0.05	83	100	26	14.0
	ConvSmall	648	97	0.05	80	100	68	22.4
CIFAR10	FCx200	200	63	2/255	36	100	22	16.8
	ConvSmall	800	55	2/255	32	98	29	41.1

monDEQ h , CRAFT propagates M-Zonotopes, utilizing Theorems 4.1 and 4.2 to determine an over-approximation $\hat{Z}^*$ of the true fixpoint set Z^* . Based on this, we compute an over-approximation of the last layer $\hat{Y} = V\hat{Z}^* + v$ and then check the logit differences $\hat{Y}_t - \hat{Y}_i > 0 \forall i \neq t$ (as described in Singh et al. (2018)) where $\hat{Y}_t - \hat{Y}_i$ denotes the minimal value of $y_t - y_i$ for $y \in \mathcal{Y}$.

This approach certifies that the true mathematical fixpoints $z^*(x')$, rather than the fixpoint found by any particular solver, for all x' with $\|x' - x\|_\infty \leq \epsilon$ lead to correct classification. However, as any correctly implemented solver is guaranteed to converge to these fixpoints with arbitrary precision, this yields a stronger certificate. Thus, we, in agreement with prior work (Chen et al., 2021; Pabbaraju et al., 2021; Revay et al., 2020), certify properties for the true fixpoints of a model.

Algorithm 1 shows a slightly simplified version of CRAFT. After initializing (line 2) $\hat{Z}_0 = \hat{U}_0 = \{z^*(x)\}$ to a concrete fixpoint we repeatedly consolidate the M-Zonotopes (Theorem 5.1), apply the set over-approximation of PR (line 7) and perform an inclusion check via Theorem 5.2 (lines 6 and 8). If the check succeeds, then by Theorem 4.1 we know that $\hat{Z}_{n+1}$ contains the true fixpoint set. After this we perform iterations of FB, which by Theorem 4.2 preserves this fixpoint set but can decrease the volume of the over-approximation, until we can verify correct classification. In App. B we discuss the engineering consideration leading to differences between the presented and implemented version.

6 EXPERIMENTAL EVALUATION

We now evaluate CRAFT on multiple monDEQs architectures for the CIFAR10 (Krizhevsky et al., 2009) and MNIST (LeCun et al., 1998) datasets. We first showcase new state-of-the-art results and then investigate the impact of different algorithmic components in an ablation study.

Experimental Setup We implemented CRAFT in PyTorch (Paszke et al., 2019) and evaluated it on a single Nvidia TITAN RTX using a 16 core Intel Xeon Gold 6242 CPU at 2.80GHz. As with prior work (Chen et al., 2021), we always evaluate the first 100 samples of the test set and report the average certification time for correctly classified samples (Time), the certified accuracy (Cert.) and the number of samples for which we found a fixpoint set over-approximation (Conv.). For implementation and experimental details as well as (hyper)parameter choices, see App. B and C.

6.1 MONDEQ ROBUSTNESS CERTIFICATION

In Table 1 we show results for a range of fully connected and convolutional monDEQs. *Bound* denotes the number of samples which were empirically robust to PGD attacks (Madry et al., 2018). We generally observe that while the smaller fully-connected networks have lower empirical robustness, it is easier to certify them, with the smallest network yielding the highest certified accuracy. Perhaps surprisingly, we find on both MNIST and CIFAR10 that convolutional networks, while empirically not quite as robust as large fully connected ones, are comparatively easier to verify, yielding notably higher certified accuracies and smaller gaps between the known upper bound and certification rate.

Comparison with SEMISDP We compare against the numbers reported for the “robustness model” (SEMISDP) approach introduced by Chen et al. (2021), the current state-of-the-art for verifying ℓ_∞ robustness properties, on the benchmarks they propose in Table 2: a fully connected MNIST network with latent space size 87, the largest their SDP solver is able to handle, and three perturbation sizes $\epsilon \in \{0.01, 0.05, 0.10\}$. For $\epsilon = 0.01$ both tools are able to certify all 99 empirically robust samples, with CRAFT on average taking only 1.4s compared to the 1350s of SEMISDP. For $\epsilon = 0.05$, CRAFT is 25% more precise (30 vs 24 samples) and with an average run time of 15.75s is two orders of magnitude faster. While for $\epsilon = 0.1$, there exist 8 empirically robust samples neither tool can verify any. Finally, as shown in Table 1, CRAFT scales to larger networks and more challenging datasets than SEMISDP. Chen et al. (2021) also propose two alternative certification modes, “lipschitz model” and “ellipsoid model”, however, neither can verify any property at $\epsilon = 0.05$, hence we omit a detailed comparison.

Table 2: Comparison of CRAFT with the approach from Chen et al. (2021) (SEMISDP) on the only network they evaluate on (FCx87).

ϵ	# Bound	SEMISDP		CRAFT (ours)	
		# Cert.	Time [s]	# Cert.	Time [s]
0.10	8	0	1350	0	9.75
0.05	75	24	1350	30	15.75
0.01	99	99	1350	99	1.40

6.2 ABLATION STUDY

Finally, in Table 3 we report the results of an ablation study on several key features of CRAFT.

M-Zonotope We analyse the effectiveness of M-Zonotope, by setting either $\mathbf{b} = 0$ (no Box) or $\mathbf{A} = 0$ (no zono). Disallowing the Zonotope component leaves a standard Box, which converges quickly, but fails to prove any property. Disallowing the Box component, which still leaves a M-Zonotope utilizing consolidation (rather than standard Zonotope), significantly reduces the range of solver parameters (in particular α) leading to convergence, up to the point where we were unable to find such parameters for some networks. The line indicates by $\dagger$ uses particularly tuned parameteres while the other our default hyperparameters, also used for other experiments.

Table 3: Overview over the obtained natural accuracy (*Acc.*), adversarial accuracy (*Bound*), the number of samples for which the fixpoint set iteration converged (*Conv.*) the certified accuracy (*Cert.*), and the average time per sample on FCx87.

Ablation	# Conv.	# Cert.	Time [s]
Reference	100	30	19.30
No zono component	100	0	0.38
No Box component	0	0	-
No Box component †	100	30	17.24
Maximum α for PR †	99	30	15.75
Only PR	100	5	9.61
Only FwdBwd	100	27	11.45
Only consider contained	100	0	7.14
No joint consolidation	0	0	-

† Significant parameter tuning needed for convergence

Iteration method Increasing the splitting parameter α for PR, which is employed until the fixpoint set iteration converges (line 7 in Algorithm 1), can reduce certification times by ~ 3.5 s at the cost of a less robust convergence. Only running PR, but not FB (Algorithm 1), retains the strong convergence but yields much looser bounds allowing us to certify only 5 samples. Running only FB forces us to choose an α for convergence that yields looser bounds, leading to only 27 samples being certified. Requiring containment to be shown for the same step in which we certify prevents all certification and highlights the importance of Theorems 4.2 and A.1.

Joint Consolidation Consolidating $\hat{\mathcal{U}}$ and $\hat{\mathcal{Z}}$ independently instead of jointly yields fewer but independent error terms, precluding any cancellations and preventing convergence.

7 CONCLUSION

We proposed a novel verification framework, called CRAFT, for fixpoint-based neural architectures such as monDEQs. The key challenge was to lift fixpoint iterations from concrete points to sets and to then approximate these sets with a novel convex relaxation, called M-Zonotope, enabling efficient inclusion checks. Our extensive experimental evaluation shows that CRAFT outperforms the current state-of-the-art in monDEQ certification by two orders of magnitude in terms of speed, one order of magnitude in terms of scalability, and about 25% in terms of certification rate.

8 REPRODUCIBILITY STATEMENT

During the review process we will make all code and trained models available as supplemental material, and upon publication release it as a public github repository. Further, App. B and C outline implementation details and parameter choices. Lastly, Section 6 contains information about the hardware used to perform all timing experiments.

9 ETHICS STATEMENT

We show robustness certification techniques for a particular class of machine learning model. While such models can be used can be used maliciously, our certification approach does not alter this. As we obtain mathematically certifiable results, advances building on this work can be used to ensure robustness of machine learning models in safety critical applications.

REFERENCES

- Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proc. of ICML*, volume 70, 2017.
- Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.
- Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Pushmeet Kohli, P Torr, and P Mudigonda. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21(2020), 2020.
- Tong Chen, Jean-Bernard Lasserre, Victor Magron, and Edouard Pauwels. Semialgebraic representation of monotone deep equilibrium models and applications to certification. *ArXiv preprint*, abs/2106.01453, 2021.
- Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proc. of ICML*, volume 97, 2019.
- Timon Gehr, Matthew Mirman, Dana Drachler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin T. Vechev. AI2: safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, 2018. doi: 10.1109/SP.2018.00058.
- Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, and Armin Askari. Implicit deep learning. *ArXiv preprint*, abs/1908.06315, 2019.
- Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiabl. *ArXiv preprint*, abs/1810.12715, 2018.
- Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy A. Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *ArXiv preprint*, abs/1910.09338, 2019.
- Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*. Springer, 2019.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Adrian Kulmburg and Matthias Althoff. On the co-np-completeness of the zonotope containment problem. *European Journal of Control*, 2021.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.
- Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *2019 IEEE Symposium on Security and Privacy (S&P)*, 2018.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of ICLR*, 2018.
- Matthew Mirman, Timon Gehr, and Martin T. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *Proc. of ICML*, volume 80, 2018.
- Chirag Pabbaraju, Ezra Winston, and J. Zico Kolter. Estimating lipschitz constants of monotone deep equilibrium models. In *Proc. of ICLR*, 2021.

- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.
- Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.
- Max Revay, Ruigang Wang, and Ian R. Manchester. Lipschitz bounded equilibrium networks. *ArXiv preprint*, abs/2010.01732, 2020.
- Ernest K Ryu and Stephen Boyd. Primer on monotone operator methods. *Appl. Comput. Math.*, 15(1), 2016.
- Sadra Sadraddini and Russ Tedrake. Linear encodings for polytope containment problems. In *58th IEEE Conference on Decision and Control, CDC 2019, Nice, France, December 11-13, 2019*, 2019. doi: 10.1109/CDC40024.2019.9029363.
- François Serre, Christoph Müller, Gagandeep Singh, Markus Püschel, and Martin Vechev. Scaling polyhedral neural network verification on GPUs. In *Proc. Machine Learning and Systems (MLSys)*, 2021.
- Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin T. Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.
- Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin T. Vechev. Beyond the single neuron convex barrier for neural network certification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019a.
- Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. An abstract domain for certifying neural networks. *PACMPL*, 3(POPL), 2019b. doi: 10.1145/3290354.
- Yusuke Tashiro, Yang Song, and Stefano Ermon. Output diversified initialization for adversarial attacks. *ArXiv preprint*, abs/2003.06878, 2020.
- Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane S. Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for relu networks. In *Proc. of ICML*, volume 80, 2018.
- Ezra Winston and J. Zico Kolter. Monotone operator equilibrium networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proc. of ICML*, volume 80, 2018.
- Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kaillkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.

A ADDITIONAL THEOREMS

Theorem A.1 (Fixed-Point set preservation for Peaceman-Rachford splitting). *Let $\mathbf{h}$ be a monDEQ with solver iteration $[z_{n+1}, u_{n+1}] = \mathbf{g}_\alpha^{PR}(\mathbf{x}, z_n, u_n)$ defined as per Eq. (4) (Peaceman-Rachford splitting). Let $\mathcal{X}$ denote the set of possible inputs and $\mathcal{Z}^*$ the set of reachable fixed-points of $\mathbf{h}$ given inputs in $\mathcal{X}$. Let $\hat{\mathcal{Z}}$ be a sound over-approximation of $\mathcal{Z}^*$, i.e., $\mathcal{Z}^* \subseteq \hat{\mathcal{Z}}$ and $\hat{\mathcal{U}}$ a sound over-approximation of the associated $\mathcal{U}^*$. Any application of the Peaceman-Rachford splitting yields an over-approximation of the fixpoint set $[\hat{\mathcal{Z}}', \hat{\mathcal{U}}'] = \mathbf{g}_\alpha^{PR\#}(\mathcal{X}, \hat{\mathcal{Z}}, \hat{\mathcal{U}})$ with $\mathcal{Z}^* \subseteq \hat{\mathcal{Z}}'$.*

Proof. To prove by contradiction, let $\tilde{z}$ be a point close to the fixpoint z^* s.t. $\|\tilde{z} - z^*\| \leq \epsilon$ with map $\tilde{z}' = \mathbf{g}_\alpha(\mathbf{x}, \tilde{z}, \mathbf{u})$ under the fixpoint iterator. Let us further assume that $\forall \mathbf{u} \in \mathcal{U}^*$ an application of $z' = \mathbf{g}_\alpha(\mathbf{x}, z^*, \mathbf{u})$ does not map back to z^* , i.e., $\|z^* - z'\| > d$.

- Recall that $\mathbf{g}_\alpha(\mathbf{x}, z, \mathbf{u})$ is locally Lipschitz with $L < \infty$ in $\mathbf{u}$, z and $\mathbf{x}$ as it is the composition of linear maps of finite width.
- It follows that $\|\mathbf{g}_\alpha(\mathbf{x}, \tilde{z}, \mathbf{u}) - \mathbf{g}_\alpha(\mathbf{x}, z^*, \mathbf{u})\| = \|\tilde{z}' - z'\| \leq L\epsilon$
- Hence by the inverse triangle inequality $\|z^* - \tilde{z}'\| \geq \|z^* - z'\| - \|\tilde{z}' - z'\| \geq d - L\epsilon$
- Choose $\epsilon < d/(L + 1) \implies d - L\epsilon > \epsilon \implies \|z^* - \tilde{z}'\| > \epsilon$
- It follows that $\|\mathbf{g}_\alpha(\mathbf{x}, \tilde{z}, \mathbf{u}) - z^*\| > \epsilon \quad \forall \tilde{z} \in \{z \mid \|z - z^*\| \leq \epsilon\}$ which contradicts the convergence guarantee.

It follows that $\exists \mathbf{u} \in \mathcal{U}^* : z^* = \mathbf{g}_\alpha(\mathbf{x}, z^*, \mathbf{u})$. $\square$

This result implies that we can apply any sound abstraction of Peaceman-Rachford splitting to an over-approximation of the iteration state $[z, u]$ and obtain a possibly tighter over-approximation of this state.

Theorem A.2 (s -step Fixed-Point contraction). *Let $[\hat{\mathcal{Z}}_{n+1}, \hat{\mathcal{U}}_{n+1}] = \mathbf{g}_\alpha^\#(\mathcal{X}, \hat{\mathcal{Z}}_n, \hat{\mathcal{U}}_n)$ be closed sets over-approximating z_{n+1} and u_{n+1} obtained by applying the solver iteration $n + 1$ times for some z_0, u_0 and all inputs $\mathbf{x} \in \mathcal{X}$. Further let $s \in \mathbb{N}^{\geq 1}$:*

$$\hat{\mathcal{Z}}_{n+s} \subseteq \hat{\mathcal{Z}}_n \wedge \hat{\mathcal{U}}_{n+s} \subseteq \hat{\mathcal{U}}_n \implies \mathcal{Z}^* \subseteq \hat{\mathcal{Z}}_{n+s} \quad (11)$$

Proof. Again we write $s = [z, u]$ and then define $s_{n+1} := \mathbf{g}'(s_n) = \mathbf{g}_\alpha(\mathbf{x}, s_n)$. Based on this, we then let $s_{n+s} = \mathbf{g}'_s(s_n) = \mathbf{g}'(\cdots \mathbf{g}'(s_n))$. Then Eq. (5) follows directly from applying Theorem 4.1 to $\mathbf{g}'_s$. $\square$

A.1 OTHER ACTIVATION FUNCTIONS

In order for CRAFT to be able to certify monDEQs utilizing an activation function σ anlogously to the discussed *ReLU*, we require:

- We need convergence and uniqueness guarantees for the original monDEQ in the concrete (via operator splitting); to this end Theorem 1 in Winston & Kolter (2020) requires σ to be a proximal operator of a CCP function, which most common Deep Learning activation functions are.
- In order to utilize FB after convergence we need a version of Theorem 4.2, which shows that Forward-Backward splitting preserves fixpoints. Our proof of Theorem 4.2 relies on the ReLU function. However, a more general proof can be constructed similar to that of Theorem A.1.
- Lastly, we need an M-Zonotope transformer for the activation function. For many choices the respective Zonotope transformers (such as those for Sigmoid and Tanh discussed in Singh et al. (2018)) can be simply adapted.

B IMPLEMENTATION DETAIL

Algorithm 1 is a slightly simplified version of the CRAFT algorithm that we actually implemented. Here we discuss the differences.

Consolidation and Inclusion check In practice we perform the consolidation (line 6) only every r^{th} iteration and only recompute the PCA basis for consolidation every 30 steps. Since we require a consolidated basis for the inclusion check (line 8) we always keep the up to 10 last consolidated $[\hat{Z}_k, \hat{U}_k]$ and check $[\hat{Z}_{n+1}, \hat{U}_{n+1}]$ against all of these. Note that this requires the use of Theorem A.2 rather than Theorem 4.1.

Consolidation after containment Furthermore, after convergence we still apply consolidation to $\hat{Z}_n$, every r' steps, but no inclusion check. We recompute the PCA basis every second check. As each iteration, the *ReLU* introduces potentially new error terms this consolidation allows us to keep the memory-footprint low.

PR after containment In Algorithm 1, after containment we switch from PR to FB. In practice however, we apply PR for another 20 iterations as it converges more stably. The correctness of this follows from both Theorem 4.1 and Theorem A.1.

Widening To show containment, not the absolute tightness of the over-approximation of the iteration state is key, but the increase in tightness. However, if we already have a very tight approximation, tightening it further can be very challenging. To overcome this issue, in Algorithm 1 we – perhaps counterintuitively – widen our over-approximation as part of the error consolidation by intentionally introducing looseness:

$$\mathbf{A}' = (w_{mul}|\tilde{\mathbf{A}}^{-1}\mathbf{A}|\mathbf{1} + w_{add}\mathbf{1}) \cdot \tilde{\mathbf{A}} \quad (12)$$

with the multiplicative and additive widening parameters w_{mul} and w_{add} , respectively. This increase in looseness between the current approximation and the exact fixpoint set. Therefore, it can be easier to tighten the approximation further and hence show containment. After containment has been shown, we set both parameters to 0.

Termination Heuristics In practice we abort the main loop early in cases where we likely won't be able to verify the input. Before convergence we abort if the volume of the M-Zonotope $[\hat{Z}_n, \hat{U}_n]$ reaches a width of 10^9 in any direction. After convergence we abort if in $3r'$ steps we did not observe any improvement in $\max_{i \neq t} \hat{\mathcal{Y}}_t - \hat{\mathcal{Y}}_i$ (and are about to compute a new PCA basis).

DeepZ Slope Additionally if we reach convergence, but are not able to verify robustness in the main loop and $\min_{i \neq t} \hat{\mathcal{Y}}_t - \hat{\mathcal{Y}}_i > -1$ we unroll 20 steps of FB into a standard-neural network and optimize the slopes of the ReLU transformer for 60 optimization steps (Wong & Kolter, 2018; Weng et al., 2018; Zhang et al., 2018). If $\min_{i \neq t} \hat{\mathcal{Y}}_t - \hat{\mathcal{Y}}_i > -0.15$ we unroll 60 steps and employ 200 optimization steps.

C PARAMETER CHOICES & EXPERIMENT DETAILS

C.1 MODEL TRAINING

All networks were trained with monotonicity parameter $m = 20.0$ using standard minibatch gradient descent and implicit differentiation as outlined in Winston & Kolter (2020).

C.2 CRAFT PARAMETERS

Generally we use the default values discussed in App. B unless stated otherwise. For all experiments we use $n_{max} = 500$ and summarize the main parameters in Table 4. By default we use $r = 3$ and increase it to 5 on larger MNIST models for better convergence.

Table 4: CRAFT verification parameters.

<i>Dataset</i>	<i>Model</i>	r	r'	α_{PR}	widening
MNIST	FCx40	3	50	0.1	const
	FCx87	3	50	0.1	const
	FCx100	5	50	0.06	const
	FCx200	5	50	0.05	const
	ConvSmall	5	50	0.05	-
CIFAR10	FCx200	3	30	0.06	exp
	ConvSmall	3	30	0.06	exp

Table 5: Comparison of M-Zonotope to other abstract domains, L denotes the number of network layers or solver iterations in our case, d is the latent space or input dimension, $d \leq n_e \leq Ld$ is the number of error terms for Zonotope.

	Propagation	Efficient Inclusion Check	Precise
Interval Bounds / Box	$\mathcal{O}(L)$ ✓	✓	✗
(Hybrid)Zonotope	$\mathcal{O}(L^2)$ ✗	✗	✓
Linear Backward-Mode	$\mathcal{O}(L^2)$ ✗	(✓)	?
Linear Forward-Mode	$\mathcal{O}(L)$ ✓	(✓)	✗
M-Zonotope	$\mathcal{O}(L)$ ✓	✓	✓

For widening “const” denotes $w_{mul} = 1 + 10^{-3}$ and $w_{add} = 1 + 10^{-2}$, “exp” denotes initialization with “const” and scaling by 1.1 and 1.2 respectively every second iteration, and “-” denotes no widening.

All parameters and in particular the values for α used in PR were found by coarse manual search. Overall we observe large stability with respect to most parameters and in particular the value of α does not have a large impact on PR, as we show in Section 6.2.

When switching from PR to FB, we apply a line search to determine an optimal α_{FB} (with regard to certification).

C.3 ADVERSARIAL ATTACK

In order to determine a bound on the certifiable accuracy of the models, we compute their empirical accuracy with respect to a strong attack. We apply a targeted version (towards all classes) of PGD (Madry et al., 2018) with 20 restarts, 50 steps utilizing margin loss (Gowal et al., 2019) and 5 step output diversification (Tashiro et al., 2020).

C.4 RELATION TO OTHER CONVEX RELAXATIONS

Here we briefly outline the differences between M-Zonotope and other abstract domains. In particular, we will highlight why only the M-Zonotope domain, developed for CRAFT, is suitable for monDEQ certification. In order to apply Algorithm 1, we require (i) an efficient propagation of abstract elements through many iterations L of operator splitting, (ii) the ability to perform efficient inclusion checks, and (iii) high enough precision to enable verification. We summarize this comparison in Table 5. We consider the dimension d of the latent space (see Table 1 for representative values) and d_i of the input, the number of iterations (or “layers”) L (at least 50 in our case) and the number of error terms n_e ($d_i \leq n_e \leq d_i + Ld$).

Interval Bounds / Box (Gehr et al., 2018; Mirman et al., 2018; Gowal et al., 2018) Interval Bound Propagation or the Box domain enable efficient propagation ($\mathcal{O}(Ld^2)$) and inclusion checks ($\mathcal{O}(d)$). However, we show in Table 3 (“No zono component”) that its precision is insufficient to verify any input. Propagation is dominated by a matrix-vector multiplication $\mathcal{O}(d^2)$ per layer L (which needs

to be executed twice) and inclusion checks can be performed by evaluating two inequalities per dimension d .

(Hybrid)Zonotope (Gehr et al., 2018; Mirman et al., 2018; Wong & Kolter, 2018; Singh et al., 2018) Zonotopes, HybridZonotopes and improper M-Zonotopes are equivalent in their precision, which we hence know to be sufficient for verification. Crucially however, there are two major differences: (i) without our error consolidation, each ReLU can potentially add an error term allowing their number to grow to $d_i + Ld$ making propagation very expensive and (ii) without requiring the error matrix of proper M-Zonotope to be basis of the $\mathbb{R}^d$, exact inclusion checks become co-NP-complete (Kulmburg & Althoff, 2021) and even approximate inclusion checks Sadraddini & Tedrake (2019) become infeasible in high (> 30) dimensions.

Linear Bounds (Zhang et al., 2018; Singh et al., 2019b) Methods based on the propagation of linear bounds (DeepPoly) are state-of-the-art for applications where runtime is critical (Serre et al., 2021). Yet, mathematically and empirically they are not necessarily more precise than Zonotope (Singh et al., 2019b). In a nutshell, for each neuron, these methods keep a lower and upper bound as a linear function in the previous layers neurons. Thus, linear bounds in terms of the input values can be computed by repeated substitutions, e.g. z_l , the activations in layer l are bounded $l_l \leq z_l \leq u_l$ where l_l, u_l are linear functions in the input values x . l_l and u_l can either be obtained by substituting forward (starting at the first layer) or backward (starting at the last layer).

In forward mode, the bounds are constructed layer-by-layer from the input, similar to Zonotope bounds, only incurring the cost of matrix-matrix multiplications ($\mathcal{O}(Ld^3)$). However, a key difference is that during the abstraction of non-linear functions, Zonotope transformers introduce only symbolic error terms which can cancel out later on while DeepPoly transformers introduce concrete error terms which can't cancel out later on. This turns out to be absolutely crucial during the iterative solve, to the extent that errors accumulate exponentially for DeepPoly, while we need widening for M-Zonotope. We showcase this in an example below.

In the backward construction, l_l is constructed by recursive back-substitution starting at the network output (or latent space in our case). While intermediate bounds can be computed more cheaply using a forward propagation method, every inclusion check (after every iteration) would require a full back-substitution through all preceding iterations. Thus the bounds for layer l have cost $\mathcal{O}(ld^3)$ (as bounding one layer is dominated by d -dimensional matrix-matrix multiplications). Thus the overall complexity becomes $\mathcal{O}(L^2d^3)$. However, while in principle limited in a similar way as forward propagation, back-substituting the linear expressions describing the outputs of affine layers allows for much more cancellations to happen before too many error terms are introduced, which leads to tighter bounds. We refer interested readers to Singh et al. (2019b) for more detail on these methods.

While their cost or lack of precision already makes both variants of DeepPoly unsuitable for mon-DEQ verification, the actual inclusion check would be conceptually and computationally simple, see Lemma 1 in Sadraddini & Tedrake (2019) for a linear programming approach with $4d^2 + d_i$ variables or Equation 9 for an approach based on $2d$ LPs with $d + d_i$ variables. Thus, depending on the encoding the overall complexity will be $\mathcal{O}((d^{4+m} + d_i^2)n)$ or $\mathcal{O}((d^{3+m} + d_i^2d)n)$ where n is the number of bits ($\mathcal{O}(dd_i)$) to encode the problem and $0 \geq m \geq 1$ (depending on the exact algorithm).

Imprecision of Forward-mode Linear Bounds To showcase the cancellation issue, consider the following simple example:

$$\begin{aligned} -0.5 &\geq x_0 \geq 0.5 \\ x_1 &= \text{ReLU}(x_0) \\ x_2 &= 2x_1 + 1 \\ x_3 &= x_2 - x_1. \end{aligned}$$

Using the Zonotope or M-Zonotope relaxation we can apply the transformers from Singh et al. (2018), to obtain the over-approximations $\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ of reachable values variables x_0, x_1, x_2, x_3 . If we want to track the reachable values with linear bounds (by following the DeepPoly transformers as described in Singh et al. (2019b)) we obtain a l_i, u_i as lower and upper bound for each x_i . l_i and u_i are linear functions in the concrete values of l_0, u_0 . Both approaches are shown below:

$\begin{aligned}\mathcal{X}_0 &= 0 + \frac{1}{2}\nu_0 & \nu_0 &\in [-1, 1] \\ \mathcal{X}_1 &= \frac{1}{8} + \frac{1}{4}\nu_0 + \frac{1}{8} + \nu_1 & \nu_1 &\in [-1, 1] \\ \mathcal{X}_2 &= \frac{10}{8} + \frac{2}{4}\nu_0 + \frac{2}{8} + \nu_1 \\ \mathcal{X}_3 &= \frac{9}{8} + \frac{1}{4}\nu_0 + \frac{1}{8} + \nu_1 \\ 0.75 &\leq x_3 \leq 1.5\end{aligned}$	$\begin{aligned}-0.5 &=: l_0 \leq x_0 \leq u_0 := 0.5 \\ l_0 &\leq x_1 \leq \frac{1}{2}u_0 + \frac{1}{4} \\ 2l_0 + 1 &\leq x_2 \leq u_0 + \frac{6}{4} \\ 2l_0 - \frac{1}{2}u_0 + \frac{3}{4} &\leq x_3 \leq -l_0 + u_0 + \frac{6}{4} \\ 0.0 &\leq x_3 \leq 1.5\end{aligned}$
--	--

We observe that the final bounds for the Zonotope (left) are much tighter than those for DeepPoly linear bounds. The reason of this is, that the linear bounds do not capture the relation between x_2 and x_1 , while the Zonotope does this via the error terms ν_0, ν_1 . This leads to an large accumulation of errors when subtracting (or adding with a negative factor) two related terms. In CRAFT this is a frequent operation, e.g., to determine $\mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x}$ in every iteration or the updates of $\mathbf{u}$ in PR, making rendering it unsuitable.

Note that we can make the DeepPoly bonds slightly tighter in this case by treating the ReLU similar to the Zonotope case, but this does not elevate the fundamental limitation.