Semialgebraic Representation of Monotone Deep Equilibrium Models and Applications to Certification

Tong Chen  
LAAS-CNRS  
Université de Toulouse  
31400 Toulouse, France  
tchen@laas.fr

Jean-Bernard Lasserre  
LAAS-CNRS & IMT  
Université de Toulouse  
31400 Toulouse, France  
lasserre@laas.fr

Victor Magron  
LAAS-CNRS  
Université de Toulouse  
31400 Toulouse, France  
vmagron@laas.fr

Edouard Pauwels  
IRIT & IMT  
Université de Toulouse  
31400 Toulouse, France  
edouard.pauwels@irit.fr

Abstract

Deep equilibrium models are based on implicitly defined functional relations and have shown competitive performance compared with the traditional deep networks. Monotone operator equilibrium networks (monDEQ) retain interesting performance with additional theoretical guaranties. Existing certification tools for classical deep networks cannot directly be applied to monDEQs for which much fewer tools exist. We introduce a semialgebraic representation for ReLU based monDEQs which allows to approximate the corresponding input output relation by semidefinite programming (SDP). We present several applications to network certification and obtain SDP models for the following problems: robustness certification, Lipschitz constant estimation, ellipsoidal uncertainty propagation. We use these models to certify robustness of monDEQs w.r.t. a general $L_q$ norm. Experimental results show that the proposed models outperform existing approaches for monDEQ certification. Furthermore, our investigations suggest that monDEQs are much more robust to $L_2$ perturbations than $L_{\infty}$ perturbations.

1 Introduction

With the increasing success of Deep Neural Networks (DNN) (e.g. computer vision, natural language processing), one witnesses a significant increase in size and complexity (topology and activation functions). This generates difficulties for theoretical analysis and a posteriori performance evaluation. This is problematic for applications where robustness issues are crucial, for example inverse problems (IP) in scientific computing. Indeed such IPs are notoriously ill-posed and as stressed in the March 2021 issue of SIAM News [1], “Yet DL has an Achilles’ heel. Current implementations can be highly unstable, meaning that a certain small perturbation to the input of a trained neural network can cause substantial change in its output. This phenomenon is both a nuisance and a major concern for the safety and robustness of DL-based systems in critical applications—like healthcare—where reliable computations are essential”. Indeed, the Instability Theorem [1] predicts unavoidable lower bound on Lipschitz contents, which may explain the lack of stability of some DNNs, over-performing on training sets. This underlines the need to evaluate precisely a posteriori critical indicators, such as Lipschitz constants of DNNs. However, obtaining an accurate upper bounds on the Lipschitz constant of a DNN is a hard problem, it reduces to certifying globally an inequality $\Psi(x) \geq 0$ for all $x$ in...
a domain”, i.e., to provide a certificate of positivity for a function $\Psi$, which has no simple explicit expression. Even for modest size DNNs this task is practically challenging, previous successful attempts [7, 22] were restricted in practice to no more than two hidden layers with less than a hundred nodes. More broadly existing attempts to DNN certification rely either on zonotope calculus, linear programming (LP) or hierarchies of SDP based on positivity certificates from algebraic geometry [26], which may suffer from the curse of dimensionality.

Recently, Deep Equilibrium Models (DEQ) [2] have emerged as a potential alternative to classical DNNs. With their much simpler layer structure, they provide competitive results on machine learning benchmarks [2, 3]. The training of DEQs involves solving fix-point equations for which algorithmic success requires conditions. Fortunately, Monotone operator equilibrium network (monDEQ) introduced in [36] satisfies such conditions. Moreover, the authors in [25] provide explicit bounds on global Lipschitz constant of monDEQs (w.r.t. the $L_2$-norm) which can be used for robustness certification.

From a certification point of view, DEQs have the definite advantage of being relatively small in size compared to DNNs and therefore potentially more amenable to sophisticated techniques (e.g. algebraic certificates of positivity) which rapidly face their limit even in modest size classical DNNs. Therefore monDEQs constitute a class of DNNs for which robustness certification, uncertainty propagation or Lipschicity could potentially be investigated in a more satisfactory way than classical networks. Contrary to DNNs, for which a variety of tools have been developed, certification of DEQ models is relatively open, the only available tool is the Lipschitz bound in [25].

Contribution

We present three general semialgebraic models of ReLU monDEQ for certification ($p \in \mathbb{Z}_+ \cup \{+\infty\}$):

- **Robustness Model** for network $L_p$ robustness certification.
- **Lipschitz Model** for network Lipschitz constant estimation with respect to any $L_p$ norm.
- **Ellipsoid Model** for ellipsoidal outer-approximation of the image by the network of a polyhedra or an ellipsoid.

All these models can be used for robustness certification, a common task which we consider experimentally. Both Lipschitz and Ellipsoid models can in addition be used for further a posteriori analyses. Interestingly, all three models are given by solutions of semidefinite programs (SDP), obtained by Shor relaxation of a common semialgebraic representation of ReLU monDEQs. Our models are all evaluated to simple ReLU monDEQs on MNIST dataset similar as [36, 25] on the task of robustness certification. We demonstrate that all three models outperform the approach of [25] and the Robustness Model being the most efficient. Our experiments also suggest that DEQs are much less robust to $L_\infty$ perturbations than $L_2$ perturbations, in contrast with classical DNNs [28].

Related works

Neural network certification is a challenging topic in machine learning, contributions include:

**Robustness certification of DNNs** Even with high test accuracy, DNNs are very sensitive to tiny input perturbations, see e.g. [33, 13]. Robustness to input perturbation has been investigated in many different works with various techniques, including SDP relaxation in SDP-cert [28], abstract interpretation with ERAN [12, 32], LP relaxation in Reluplex [19], analytical certification with Fast-lin [35] and CROWN [38], and their extension to convolutional neural networks with CNN-Cert [6]. All these methods are restricted to DNNs or CNNs and do not directly apply to DEQs.

**Lipschitz constant estimation of DNNs** Lipschitz constant of DNNs can be used for robustness certification [33, 15]. Existing contributions include naive layer-wise product bound [18, 34], the LP relaxations for CNNs [39] and DNNs [22], as well as SDP relaxations [27, 11, 7].

**Lipschitz constant estimation of DEQs** The authors in [25] provide an optimization-free upper bound of the Lipschitz constant of monDEQs depending on network weights. This bound is valid for $L^2$ norm and we present a general model for arbitrary $L^p$ norm.
2 Preliminary Background and Notations

We consider the \textit{monotone operator equilibrium network (monDEQ)} \cite{monDEQ}, with a single implicit hidden layer. The main difference between monDEQ and \textit{deep equilibrium network (DEQ)} \cite{DEQ} is that strong monotonicity is enforced on weight matrix and activation functions to guarantee the convergence of fixed point iterations. The authors in \cite{monDEQ} proposed various structures of implicit layer, we only consider fully-connected layers, investigation of more advanced convolutional layers is in our list of future works.

\textbf{Network description:} Denote by $F : \mathbb{R}^{p_0} \rightarrow \mathbb{R}^K$ a fully-connected monDEQ for classification, where $p_0$ is the input dimension and $K$ is the number of labels. Let $x_0 \in \mathbb{R}^{p_0}$ be the input variable and $z \in \mathbb{R}^p$ be the variable in the implicit layer. We consider the ReLU activation function, which is simply defined as $\text{ReLU}(x) = \max\{0, x\}$, and the output of the monDEQs can be written as

$$F(x_0) = Cz + c, z = \text{ReLU}(Wz + Ux_0 + u), \quad (\text{monDEQ})$$

where $W \in \mathbb{R}^{p \times p}, U \in \mathbb{R}^{p \times p_0}, u \in \mathbb{R}^p, C \in \mathbb{R}^{K \times p}, c \in \mathbb{R}^K$ are parameters of the network. The vector-valued function $F(x_0)$ provides a score for each label $i \in \{1, \ldots, K\}$ associated to the input $x_0$, the prediction corresponds to the highest score, i.e., $y_{x_0} = \max_{i=1,...,K} F(x_0)_i$. As in \cite{monDEQ}, the matrix $I_p - W$ is strongly monotone: there is a known $m > 0$ such that $I_p - W \succeq mI_p$, this constraint can be enforced by specific parametrization of the matrix $W$. With the monotonicity assumption, the solution to equation $z = \text{ReLU}(Wz + Ux_0 + u)$ is unique and can be evaluated using convergent algorithms, see \cite{monDEQ} for more details.

\textbf{Robustness of monDEQs:} Given an input $x_0 \in \mathbb{R}^{p_0}$, a norm $\| \cdot \|$, and a network $F : \mathbb{R}^{p_0} \rightarrow \mathbb{R}^K$, let $y_0$ be the label of input $x_0$, i.e., $y_0 = \max_{i=1,...,K} F(x_0)_i$. For $\varepsilon > 0$, denote by $E = \mathbb{B}(x_0, \varepsilon, \| \cdot \|)$ the ball centered at $x_0$ with radius $\varepsilon$ for norm $\| \cdot \|$. If for all inputs $x \in E$, the label of $x$ equals $y_0$, i.e., $y = \max_{i=1,...,K} F(x)_i = y_0$, then we say that the network $F$ is $\varepsilon$-robust at input $x_0$ for norm $\| \cdot \|$. An equivalent way to verify whether the network $F$ is $\varepsilon$-robust is to check that for all labels $i \neq y_0$, $F(x_i) - F(x_{y_0}) < 0$.

\textbf{Semialgebraicity of ReLU function:} The key reason why neural networks with ReLU activation function can be tackled using polynomial optimization techniques is \textit{semialgebraicity} of the ReLU function, i.e., it can be expressed with a system of polynomial (in)equalities. For $x, y \in \mathbb{R}$, we have $y = \text{ReLU}(x) = \max\{0, x\}$ if and only if $y(y-x) = 0, y \geq x, y \geq 0$. For $x, y \in \mathbb{R}^n$, we denote by $\text{ReLU}(x)$ the coordinate-wise evaluation of ReLU function, and by $xy$ the coordinate-wise product of $x$ and $y$. A subset of $\mathbb{R}^n$ defined by a finite conjunction of polynomial (in)equalities is called a \textit{basic closed semialgebraic set}. The graph of the ReLU function is a basic closed semialgebraic set.

Going back to equation \textbf{(monDEQ)}, we have the following equivalence:

$$z = \text{ReLU}(Wz + Ux_0 + u) \iff z - Wz - Ux_0 - u = 0, z \succeq Wz + Ux_0 + u, z \geq 0, \quad (1)$$

where the right hand side is a system of polynomial (in)equalities. For the rest of the paper, mention of the ReLU function will refer to the equivalent polynomial system in \textbf{(1)}.

\textbf{POP and Putinar’s positivity certificate:} In general, a \textit{polynomial optimization problem (POP)} has the form

$$\rho = \max_{x \in \mathbb{R}^n} \{ f(x) : g_i(x) \geq 0, i = 1, \ldots, p \}, \quad (\text{POP}_0)$$

where $f$ and $g_i$ are polynomials whose degree is denoted by $\deg$. The robustness certification model \textbf{(3.1)}, Lipchitz constant model \textbf{(3.2)} and ellipsoid model \textbf{(3.3)} are all POPs.

In most cases, the POPs are non-linear and non-convex problems, which makes them NP-hard. A typical approach to reduce the complexity of these problems is replacing the positivity constraints by Putinar’s positivity certificate \cite{Putinar}. The problem \textbf{(POP)_0} is equivalent to

$$\rho = \min_{\lambda \in \mathbb{R}} \{ \lambda : \lambda - f(x) \geq 0, g_i(x) \geq 0, i = 1, \ldots, p, \forall x \in \mathbb{R}^n \}. \quad (\text{POP})$$

In order to reduce the size of the feasible set of problem \textbf{(POP)}, we replace the positivity constraint $\lambda - f(x) \geq 0$ by a weighted sum-of-square (SOS) polynomial decomposition, involving the polynomials $g_i$. Let $d$ be a non-negative integer. Denote by $\sigma^d_0(x), \sigma^d_i(x)$ some SOS polynomials of degree at
most \( 2d \), for each \( i = 1, \ldots, p \). Note that if \( d = 0 \), such polynomials are non-negative real numbers. Then the positivity of \( \lambda - f(x) \) is implied by the following decomposition

\[
\lambda - f(x) = \sigma_0^d(x) + \sum_{i=1}^{p} \sigma_i^{d-\omega_i}(x)g_i(x), \quad \omega_i = \lceil \deg g_i/2 \rceil, \quad \forall x \in \mathbb{R}^n, \quad (2)
\]

for any \( d \geq \max_i \omega_i \). Equation (2) is called the order-\( d \) Putinar’s certificate. By replacing the positivity constraint \( f(x) - \lambda \geq 0 \) in problem (POP) by Putinar’s certificate (2), we have for \( d \geq \max_i \omega_i \),

\[
\rho_d = \min_{x \in \mathbb{R}} \{ \lambda : \lambda - f(x) = \sigma_0^d(x) + \sum_{i=1}^{p} \sigma_i^{d-\omega_i}(x)g_i(x), \quad \omega_i = \lceil \deg g_i/2 \rceil, \forall x \in \mathbb{R}^n \}. \quad \text{(POP-d)}
\]

It is obvious that \( \rho_d \geq \rho \) for all \( d \geq \max_i \omega_i \). Under certain conditions (slightly stronger than compactness of the set of constraints), it is shown that \( \lim_{d \to \infty} \rho_d = \rho \) \([21]\). The main advantage of relaxing problem (POP) to (POP-\( d \)) is that problem (POP-\( d \)) can be efficiently solved by semidefinite programming (SDP). Indeed a polynomial \( f \) of degree at most \( 2d \) is SOS if and only if there exists a positive semidefinite (PSD) matrix \( M \) (called a Gram matrix) such that \( f(x) = v(x)^T M v(x) \), for all \( x \in \mathbb{R}^p \), where \( v(x) \) is the vector of monomials of degree at most \( d \).

Problem (POP-\( d \)) is also called the order-\( d \) Lasserre’s relaxation. When the input polynomials are quadratic, the order-1 Lasserre’s relaxation is also known as Shor’s relaxation \([31]\). All our models are obtained using variations of Shor’s relaxation applied to different POPs, see Section 3.3 for more details.

## 3 Semialgebraic Models for Certifying Robustness of Neural Networks

In this section, we introduce several models for certification of monDEQs. All the models are based on semialgebraicity of ReLU and \( \partial \)ReLU (the subgradient of ReLU, see Section 3.2) to translate our targeted problems to POPs. Then we use Putinar’s certificates, defined in Section 2, to relax the non-convex problems to convex SDPs which can be solved efficiently using modern solvers. Each model can be eventually used to certify robustness but they also have their own independent interest.

**Notations:** Throughout this section, we consider a monDEQ for classification, denoted by \( F \), with fixed, given parameters, \( W \in \mathbb{R}^{p \times p}, \ U \in \mathbb{R}^{p \times p}, \ a \in \mathbb{R}^p, \ C \in \mathbb{R}^{K \times p}, \ c \in \mathbb{R}^K \), where \( p_0 \) is the number of input neurons, \( p \) is the number of hidden neurons, and \( K \) is the number of labels. For \( q \in \mathbb{Z}_+ \cup \{ +\infty \}, \| \cdot \|_q \) is the \( L_q \) norm defined by \( \| x \|_q := (\sum_{i=1}^{p_0} |x_i|^q)^{1/q} \) for all \( x \in \mathbb{R}^{p_0} \). Throughout this section \( \epsilon > 0 \) and \( x_0 \in \mathbb{R}^p \) are fixed, we denote by \( \mathcal{E} := \mathbb{B}(x_0, \epsilon, \| \cdot \|_q) := \{ x \in \mathbb{R}^{p_0} : \| x - x_0 \|_q \leq \epsilon \} \) the ball centered at \( x_0 \) with radius \( \epsilon \) for \( L_q \) norm, a perturbation region. If \( q < +\infty \), i.e., \( q \) is a positive integer, \( \| x - x_0 \|_q \leq \epsilon \) is equivalent to the polynomial inequality \( \| x - x_0 \|_q^q \leq \epsilon^q ; \) if \( q = +\infty \), \( \| x - x_0 \|_q \leq \epsilon \) is equivalent to \( \| x - x_0 \|_2 \leq \epsilon^2 \) (where \( \| \cdot \| \) denotes the vector of absolute values of coordinates of \( x \)) which is a system of \( p_0 \) polynomial inequalities. Hence the input set \( \mathcal{E} \) is a semialgebraic set for all considered \( L_q \) norms. For a matrix \( A \in \mathbb{R}^{m \times n} \), its operator norm induced by the norm \( \| \cdot \| \) is given by \( \| A \| := \inf \{ \lambda : \| Ax \| \leq \lambda \| x \|, \forall x \in \mathbb{R}^n \} \).

### 3.1 Robustness Model

Let \( y_0 \) be the label of \( x_0 \) and let \( z \in \mathbb{R}^p \) be the variables in the monDEQ implicit layer. The proposed model directly estimates upper bounds on the gap between the score of label \( y_0 \) and the score of labels different from \( y_0 \). Precisely, for \( i \in \{ 1, \ldots, K \} \) such that \( i \neq y_0 \), denote by \( \xi_i = (C_i - C_{y_0})^T \). For \( x \in \mathcal{E} \), the gap between its score of label \( i \) and label \( y_0 \) is \( F(x)_i - F(x)_{y_0} = \xi_i^T z \). The Robustness Model for monDEQ reads:

\[
\delta_i := \max_{x \in \mathbb{R}^{p_0}, z \in \mathbb{R}^n} \left\{ \xi_i^T z : z = \text{ReLU}(Wz + Ux + u), \ x \in \mathcal{E} \right\}. \quad \text{(CertMON-\( i \))}
\]

Using the semialgebraicity of both ReLU in \([1]\), and set \( \mathcal{E} \), problem (CertMON-\( i \)) is a POP for all \( i \). As discussed in Section 2 one is able to derive a sequence of SDPs (Lasserre’s relaxation) to obtain a converging serie of upper bounds of the optimal solution of (CertMON-\( i \)). For Robustness Model, we consider only the order-1 Lasserre’s relaxation (Shor’s relaxation), and denote by \( \delta_i \) the upper
bound of $\delta_i$ by Shor’s relaxation, i.e., $\delta_i \leq \tilde{\delta}_i$. Recall that if for all $i$ different from $y_0$, we have $F(x_i) < F(x_{y_0})$, then the label of $x$ is still $y_0$. This justifies the following claim:

**Certification criterion:** If $\tilde{\delta}_i < 0$ for all $i \neq y_0$, then the network $F$ is $\varepsilon$-robust at $x_0$.

Robustness Model for DNNs has already been investigated in [25], where the authors also use Shor’s relaxation as we do. Different from DNNs, we only have one implicit layer in monDEQ. Therefore, the number of variables in problem (CertMON) only depends on the number of input neurons $p_0$ and hidden neurons $p$.

### 3.2 Lipschitz Model

We bound the Lipschitz constant of monDEQ with respect to input perturbation. Recall that the Lipschitz constant of the vector-valued function $F$ (resp. $z$) w.r.t. the $L_q$ norm and input ball $S \supset \mathcal{E}$, denoted by $L^q_{F,S}$ (resp. $L^q_{z,S}$), is the smallest value of $L$ such that $\|F(x) - F(y)\|_q \leq L\|x - y\|_q$ (resp. $\|z(x) - z(y)\|_q \leq L\|x - y\|_q$) for all $x, y \in S$. For $x, x_0 \in S$, with $\|x - x_0\|_q \leq \varepsilon$, we can estimate the perturbation of the output as follows:

$$\|F(x) - F(x_0)\|_q \leq L^q_{F,S} \cdot \|x - x_0\|_q \leq \varepsilon L^q_{F,S},$$

(3)

$$\|F(x) - F(x_0)\|_q \leq \|C\|_q \cdot L^q_{z,S} \cdot \|x - x_0\|_q \leq \|C\|_q \cdot L^q_{z,S}.$$  (4)

The authors in [25] use inequality (4) with $q = 2$, as they provide an upper bound of $L^2_{z,S}$. In contrast, our model provides upper bounds on Lipschitz constants of $F$ or $z$ for arbitrary $L_q$ norms. We directly focus on estimating the value of $L^q_{F,S}$ instead of $L^q_{z,S}$.

Since the ReLU function is non-smooth, we define its subgradient, denoted by $\partial \operatorname{ReLU}$, as the set-valued map $\partial \operatorname{ReLU}(x) = 0$ for $x < 0$, $\partial \operatorname{ReLU}(x) = 1$ for $x > 0$, and $\partial \operatorname{ReLU}(x) = [0, 1]$ for $x = 0$. Similar to ReLU function, $\partial \operatorname{ReLU}$ is also semialgebraic.

**Semialgebraicity of $\partial \operatorname{ReLU}$:** If $x, y \in \mathbb{R}$, we have $y \in \partial \operatorname{ReLU}(x)$, if and only if $y(y-1) \leq 0, xy \geq 0, x(y-1) \geq 0$. If $x, y \in \mathbb{R}^n$, then $\partial \operatorname{ReLU}(x)$ denotes the coordinate-wise evaluation of $\partial \operatorname{ReLU}$.

Going back to monDEQ, let $s$ be any subgradient of the implicit variable: $s \in \partial \operatorname{ReLU}(\mathbb{W}z + \mathbb{U}x + \mathbb{u})$. We can write equivalently a system of polynomial inequalities:

$$s(s-1) \leq 0, \ s(\mathbb{W}z + \mathbb{U}x + \mathbb{u}) \geq 0, \ (s-1)(\mathbb{W}z + \mathbb{U}x + \mathbb{u}) \geq 0.$$  (5)

For the following discussion, $\partial \operatorname{ReLU}$ will refer to the equivalent polynomial systems (5).

With the semialgebraicity of $\operatorname{ReLU}$ in (1) and $\partial \operatorname{ReLU}$ in (5), one is able to compute upper bounds of the Lipschitz constant of $F$ via POPs. The proof of the next Lemma is postponed to Appendix A.1.

**Lemma 1 Define**

$$\hat{L}^q_{F,S} = \max\{t^T U^T y : t, x \in \mathbb{R}^{p_0}, s, z, y, r \in \mathbb{R}^p, v, w \in \mathbb{R}^k, x \in \mathcal{S}, ||t||_q \leq 1, w^T v \leq 1, ||w||_q \leq 1, r - W^T y = C^T v, \ y = \text{diag}(s) \cdot r, s \in \partial \operatorname{ReLU}(\mathbb{W}z + \mathbb{U}x + \mathbb{u}), z = \operatorname{ReLU}(\mathbb{W}z + \mathbb{U}x + \mathbb{u})\}.$$  (LipMON)

Then $\hat{L}^q_{F,S}$ is an upper bound of the Lipschitz constant of $F$ w.r.t. the $L_q$ norm, i.e., $L^q_{F,S} \leq \hat{L}^q_{F,S}$.

Since problem (LipMON) is a POP, we also consider Shor’s relaxation and denote by $\hat{L}^q_{F,S}$ the upper bound of $\hat{L}^q_{F,S}$ by Shor’s relaxation, i.e., $L^q_{F,S} \leq \hat{L}^q_{F,S}$. Define $\delta := \varepsilon \hat{L}^q_{F,S}$. By equations (1), (4), if $\mathcal{E} \subset \mathcal{S}$, using Lemma 1 and the fact that $\| \cdot \|_\infty \leq \| \cdot \|_q$, we have $\|F(x) - F(x_0)\|_\infty \leq \delta$, yielding the following criterion:

**Certification criterion:** Let $y_0$ be the label of $x_0$. Define $\tau := F(x_0)_{y_0} - \max_{k \neq y_0} F(x_0)_k$. If $2\delta < \tau$, then the network $F$ is $\varepsilon$-robust at $x_0$.

**Remark:** In order to avoid some possible numerical issues, we add some bound constraints and redundant constraints to problem (LipMON), see Appendix A.2 for details. Shor’s relaxation of Lipschitz Model for DNNs has already been extensively investigated in [11, 7]. If one want to certify robustness for several input test examples, then one may choose $\mathcal{S}$ to be a big ball containing all such examples with an additional margin of $\varepsilon$.  

5
3.3 Ellipsoid Model

As above, the input region $E$ is a neighborhood of input $x_0 \in \mathbb{R}^n$ with radius $\varepsilon$ for the $L_2$ norm, i.e., $E = \{ x \in \mathbb{R}^n : \| x - x_0 \|_2 \leq \varepsilon \}$. More generally, $E$ could be other general semialgebraic sets, such as polytopes or zonotopes. Denote by $F(E)$ the image of $E$ by $F$. In this section, we aim at finding a semialgebraic set $C$ such that $F(E) \subseteq C$. We choose $C$ to be an ellipsoid which can in turn be used for robustness certification. Our goal is to find such outer-approximation ellipsoid with minimum volume. Let $C := \{ \xi \in \mathbb{R}^K : \| Q \xi + b \|_2 \leq 1 \}$ be an ellipsoid in the output space $\mathbb{R}^K$ parametrized by $Q \in \mathbb{S}^K$ and $b \in \mathbb{R}^K$, where $\mathbb{S}^K$ is the set of PSD matrices of size $K \times K$. The problem of finding the minimum-volume ellipsoid containing the image $F(E)$ can be formulated as

$$\max_{Q \in \mathbb{S}^K, b \in \mathbb{R}^K} \{ \det(Q) : x \in E, \| Q(Cz + c) + b \|_2 \leq 1, z = \text{ReLU}(Wz + Ux + u) \}.$$  

(EllipMON-POP)

By semialgebraicity of both ReLU in (1) and $C$, problem (EllipMON-POP) can be cast as a POP (the determinant being a polynomial). We no longer apply Shor’s relaxation, we rather replace the non-negativity output constraint $1 - \| Q(Cz + c) + b \|_2 \geq 0$ by a stronger Putinar’s certificate related to both ReLU and input constraints. We can then relax the non-convex problem (EllipMON-POP) to a problem with SOS constraints, which can be reformulated by (convex) SDP constraints. This is due to the fact that a polynomial $f$ of degree at most $2d$ is SOS if and only if there exists a PSD matrix $M$ (called a Gram matrix) such that $f(x) = v(x)^T M v(x)$, for all $x \in \mathbb{R}^p$, with $v(x)$ being the vector containing all monomials of degree at most $d$. In summary, we relax problem (EllipMON-POP) to an SOS constrained problem keeping the determinant unchanged:

$$\max_{Q \in \mathbb{S}^K, b \in \mathbb{R}^K} \{ \det(Q) : 1 - \| Q(Cz + c) + b \|_2^2 = \sigma_0(x, z) + \sigma_1(x, z)^T g_q(x - x_0) + \tau(x, z)^T (z - Wz - Ux - u)) + \sigma_2(x, z)^T (z - Wz - Ux - u) + \sigma_3(x, z)^T z \}.$$  

(EllipMON-SOS-d)

where $g_q(x) = \varepsilon^q - \| x \|_2^q$ for $q < +\infty$ and $g_q(x) = \varepsilon^2 - \| x \|_2^2$ for $q = +\infty$, $\sigma_0$ is a vector of SOS polynomials of degree at most $2d$, $\sigma_1$ is a vector of SOS polynomials of degree at most $2d - \lceil q/2 \rceil$ for $q < +\infty$ and $2d - 2$ for $q = +\infty$, $\sigma_2, \sigma_3$ are vectors of SOS polynomials of degrees at most $2d - 2$, $\tau$ is a vector of polynomials of degree at most $2d - 2$. Problem (EllipMON-SOS-d) provides an ellipsoid feasible for (EllipMON-POP), that is an ellipsoid which contains $F(E)$. In practice, the determinant is replaced by a log-det objective because there exist efficient solver dedicated to optimize such objectives on SDP constraints. By increasing the relaxation order $d$ in problem (EllipMON-SOS-d), one is able to obtain a hierarchy of log-det objected SDP problems for which the outer-approximation ellipsoids have decreasing volumes.

In this paper, we only consider the case $p = 2, \infty$, and order-1 relaxation ($d = 1$). Therefore, $\sigma_i$ and $\tau$ are all (vectors of) real (non-negative) numbers for $i = 1, 2, 3$, except that $\sigma_0$ is an SOS polynomial of degree at most 2. In this case, problem (EllipMON-SOS-d) is equivalent to a problem with log-det objective and SDP constraints, as the following lemma states (proof postponed to Appendix A.3):

**Lemma 2** For $p = 2$ or $p = \infty$, problem (EllipMON-SOS-d) with $d = 1$ is equivalent to

$$\max_{Q \in \mathbb{S}^K, b \in \mathbb{R}^K, \sigma_1, \sigma_2, \sigma_3 \geq 0, \tau \in \mathbb{R}^p} \{ \log \det(Q) : -M \succeq 0 \}.$$  

(EllipMON-SDP)

where $M \in \mathbb{S}^{(p_0+1) \times (p_0+1)}$ is a symmetric matrix parametrized by the decision variables $(Q, b)$, the coefficients $(\sigma_1, \sigma_2, \sigma_3, \tau)$, and the parameters of the network $(W, U, u, C, c)$.

Since the outer-approximation ellipsoid $C = \{ \xi \in \mathbb{R}^K : \| Q \xi + b \|_2 \leq 1 \}$ contains the image $F(E)$, i.e., all possible outputs of the input region $E$, one is able to certify robustness by solving the following optimization problems.

**Certification criterion:** Let $y_0$ be the label of $x_0$. For $i \neq y_0$, define $\delta_i := \max_{\xi \in \mathbb{R}^K} \{ \xi_i - \xi_{y_0} : \| Q \xi + b \|_2 \leq 1 \}$. If $\delta_i < 0$ for all $i \neq y_0$, then the network $F$ is $\varepsilon$-robust at $x_0$.

The certification criterion for Ellipsoid Model has a geometric explanation: for $i \neq y_0$, denote by $P_i$ the projection map from output space $\mathbb{R}^K$ to its 2-dimensional subspace $\mathbb{R}^{\delta_i} \times \mathbb{R}_{i}$, i.e., $P_i(\xi) = [\xi_{y_0}, \xi_i]^T$ for all $\xi \in \mathbb{R}^K$. Let $L_i$ be the line in subspace $\mathbb{R}^{\delta_i} \times \mathbb{R}_{i}$ defined by $[\xi_{y_0}, \xi_i]^T \in \mathbb{R}^{\delta_i} \times \mathbb{R}_{i}$: $\xi_{y_0} = \xi_i$. Then the network $F$ is $\varepsilon$-robust if and only if the projection $P_i(C)$ lies strictly below the line $L_i$, for all $i \neq y_0$. We give an explicit example in Section A.3 to visually illustrate this.
3.4 Summary of the Models

We have already presented three models which can all be dedicated to certify robustness of neural networks. However, the size and complexity of each model are different. We summarize the number of variables in each model and the maximum size of PSD matrices in the resulting Shor’s relaxation, see Table 1. The complexity of our models only depends on the number of neurons in the input layer and implicit layer. The size of PSD matrices is a limiting factor for SDP solvers, our models are practically restricted to network for which such size can be handled by SDP solvers. For the popular dataset MNIST [37], whose input dimension is $28 \times 28 = 784$, we are able to apply our model on monDEQs with moderate size implicit layers (87) and report the corresponding computation time.

### Table 1: Summary of the number of variables the three models

|                     | Robustness Model | Lipschitz Model | Ellipsoid Model |
|---------------------|------------------|-----------------|----------------|
| Num. of variables   | $p_0 + p$        | $2p_0 + 4p + 2K$| $p_0 + 3p + K + K^2$ |
| Max. size of PSD matrices | $1 + p_0 + p$ | $1 + 2p_0 + 4p + 2K$ | $1 + p_0 + p$ |

4 Experiments

In this section, we present the experimental results of Robustness Model, Lipschitz Model and Ellipsoid Model described in Section 3 for a pretrained monDEQ on MNIST dataset. The network we use consists of a fully-connected implicit layer with 87 neurons and we set its monotonicity parameter $m$ to be 20. The training hyperparameters are set to be the same as in Table D1 of [36], where the training code (in Python) is available at [https://github.com/locuslab/monotone_op_net](https://github.com/locuslab/monotone_op_net). Training is based on the normalized MNIST database in [36], we use the same normalization setting on each test example with mean $\mu = 0.1307$ and standard deviation $\sigma = 0.3081$, which means that each input is an image of size $28 \times 28$ with entries varying from $-0.42$ to $2.82$. And for every perturbation $\varepsilon$, we also take the normalization into account, i.e., we use the normalized perturbation $\varepsilon/\sigma$ for each input.

Since all our three models can be applied to certify robustness of neural networks, we first compare the performance of each model in certification of the first 100 test MNIST examples. Then we compare the upper bounds of Lipschitz Model with the upper bounds proposed in [25]. Finally we show that Ellipsoid Model can also be applied for reachability analysis. For Certification model and Lipschitz model, we implement them in Julia [4] with JuMP [9] package; for Ellipsoid model, we implement it in Matlab [30] with CVX [14] package. For all the three models, we use Mosek [23] as a backend to solve the targeted POPs. All experiments are performed on a personal laptop with an Intel 8-Core i7-8665U CPU @ 1.90GHz Ubuntu 18.04.5 LTS, 32GB RAM. The code of all our models is available at [https://github.com/NeurIPS2021Paper4075/SemiMonDEQ](https://github.com/NeurIPS2021Paper4075/SemiMonDEQ).

4.1 Robustness certification

We consider $\varepsilon = 0.1$ for the $L_2$ norm and $\varepsilon = 0.1, 0.05, 0.01$ for the $L_\infty$ norm. For each model, we compute the ratio of certified test examples among the first 100 test inputs. Following [25], we also compute the projected gradient descent (PGD) attack accuracy using Foolbox library [29], which indicates the ratio of non-successful attacks among our 100 inputs. Note that the ratio of certified examples should always be less or equal than the ratio of non-successful attacks. The gaps between them shows how many test examples there are for which we are neither able to certify robustness nor find adversarial attacks.

**Remark:** For Lipschitz Model, we use inequality (3) to test robustness, i.e., we compute directly the upper bound of the Lipschitz constant of $F$ rather than $z$ (seen as a function of $x$) where $S$ is a big ball containing all test examples.

From Table 2 we see that the monDEQ is robust to all the 100 test examples for the $L_2$ norm and $\varepsilon = 0.1$ (the only example that we can not certify is because the label itself is wrong). However, it is not robust for the $L_\infty$ norm at the same level of perturbation (all our three models cannot certify any examples, and the PGD algorithm finds adversarial examples for 85% of the inputs. The network
becomes robust again for the $L_{\infty}$ norm when we reduce the perturbation $\varepsilon$ to 0.01. Overall, we see that Robustness Model is the best model as it provides the highest ratio, Ellipsoid Model is the second best model compared to Robustness Model, and Lipschitz Model provides the lowest ratio. As a trade-off, for each test example, Robustness Model requires to consider at most 9 optimization problems, each one being solved in around 150 seconds, while Ellipsoid Model requires to consider only one problem, which is solved in around 500 seconds. We only need to calculate one (global) Lipschitz constant, which takes around 1500 seconds, so that we are able to certify any number of inputs. Each model we propose provide better or equal certification accuracy compared to [25], and significant improvements for $L_{\infty}$ perturbations.

Table 2: Ratio of certified test examples and running time per example by different methods. We consider $L_2$ norm with $\varepsilon = 0.1$ and $L_{\infty}$ norm with $\varepsilon = 0.1, 0.05, 0.01$. The ratio is based on the first 100 MNIST test examples, and we count the average computation time (with unit second) for one example of each method. The ratio in parentheses of the column “Lipschitz Model” are computed by the Lipschitz constant given in [25] (see Section 4.2 for details). Exact binomial 95% confidence intervals are given in bracket.

| Norm   | $\varepsilon$ | Robustness Model (1500s / example) | Lipschitz Model (1500s in total) | Ellipsoid Model (500s / example) | PGD Attack
|--------|---------------|-----------------------------------|---------------------------------|---------------------------------|-----------
| $L_2$  | 0.1           | 99% [>94]                         | 91% (91%) [>83]                 | 99%[>94]                        | 99%[>94] |
| $L_{\infty}$ | 0.1           | 0% [<4]                           | 0% (0%) [<4]                    | 0%[<4]                          | 15% [8, 24] |
|        | 0.05          | 24% [16, 34]                      | 0% (0%) [<4]                    | 0%[<4]                          | 82% [73, 89] |
|        | 0.01          | 99% [>94]                         | 24% [16, 34] (0% [<4])          | 92%[>84]                        | 99% [>94] |

Figure 2 in Appendix A.4 shows the original image of the first test example (2a) and an adversarial attack (2b) for the $L_{\infty}$ norm with $\varepsilon = 0.1$ found by the PGD algorithm in [29].

4.2 Comparison with Lipschitz constants

In this section, we compare the upper bounds of Lipschitz constants computed by Lipschitz Model with the upper bounds proposed in [25]. Notice that the upper bounds in [25] only involve the function $z(\mathbf{x})$, hence we are only able to use inequality 4 to test robustness. In fact the quantity $|||C|||_q \cdot L_{q,S}^z$ can be regarded as an upper bound of $L_{F,S}^z$, the Lipschitz constant of $F$. We denote by $UB_2^z$ the upper bound of the Lipschitz constant of $z$ w.r.t. the $L_2$ norm, given by $UB_2^z = |||U|||_2/m$ according to [25], where $U$ is the parameter of the network and $m$ is the monotonicity factor. We can then compute the upper bound w.r.t. the $L_{\infty}$ norm by $UB_{\infty}^z = \sqrt{p_0} \cdot UB_2^z$ where $p_0$ is the input dimension. The upper bound of Lipschitz constant of $F$ is computed via the upper bound of $z$: $UB_F^z = |||C|||_q \cdot UB_2^z$. Denote similarly by SemiUB$^q_F$ the upper bounds of Lipschitz constants of $F$ provided by Lipschitz Model, w.r.t. the $L_q$ norm.

Table 3: Comparison of upper bounds of Lipschitz constant for $L_2$ and $L_{\infty}$ norm, and the corresponding computation time (with unit second).

| $q$  | $UB_F^q$ | SemiUB$^q_F$ |
|------|----------|--------------|
| $L_2$ | 4.80     | 4.67         |
| $L_{\infty}$ | 824.14   | 1756.58      |

From Table 3, we see that Lipschitz Model provides consistently tighter upper bounds than the ones in [25]. Especially for $L_{\infty}$ norm, the upper bound computed by $|||C|||_{\infty} \cdot UB_{\infty}^z$ is rather crude compared to the bound obtained directly by Lipschitz Model. Therefore, we are able to certify more examples using SemiUB$^q_F$ than $UB_F^q$, see Table 2.
4.3 Outer ellipsoid approximation

In this section, we provide a visible illustration of how Ellipsoid Model can be applied to certify robustness of neural networks. Recall that Ellipsoid Model computes a minimum-volume ellipsoid \( C = \{ \xi \in \mathbb{R}^K : \|Q\xi + b\|_2 \leq 1 \} \) in the output space \( \mathbb{R}^K \) that contains the image of the input region \( \mathcal{E} \) by \( F \), i.e., \( F(\mathcal{E}) \subseteq C \). The certification criterion for Ellipsoid Model claims that the network \( F \) is \( \varepsilon \)-robust at input \( x_0 \) if and only if the projection of \( C \) onto \( \mathbb{R}^{y_0} \times \mathbb{R}^i \) lies below the line \( L_i \) for all labels \( i \neq y_0 \).

Take the first MNIST test example (which is classified as 7) for illustration. For \( \varepsilon = 0.1 \), this example is certified to be robust for the \( L_2 \) norm but not for the \( L_\infty \) norm. We show the landscape of the projections onto \( \mathbb{R}^7 \times \mathbb{R}_3 \), i.e., the \( x \)-axis indicates label 7 and the \( y \)-axis indicates label 3. In Figure 1, the red points are projections of points in the image \( F(\mathcal{E}) \), for \( F \) an \( L_2 \) or \( L_\infty \) norm perturbation zone, the black circles are projections of some (successful and unsuccessful) adversarial examples found by the PGD algorithm. Notice that the adversarial examples also lie in the image \( F(\mathcal{E}) \). The blue curve is the boundary of the projection of the outer-approximation ellipsoid (which is an ellipse), and the blue dashed line plays the role of a certification threshold. Figure 1 shows the landscape for the \( L_2 \) norm, we see that the ellipse lies strictly below the threshold line, which means that for all points \( \xi \in C \), we have \( \xi_3 < \xi_7 \). Hence for all \( \xi \in F(\mathcal{E}) \), we also have \( \xi_3 < \xi_7 \). On the other hand, for the \( L_\infty \) norm, we see from Figure 1b that the threshold line crosses the ellipse, which means that we are not able to certify robustness of this example by Ellipsoid Model. Indeed, we can find adversarial examples with the PGD algorithm, as shown in Figure 1b by the black circles that lie above the threshold line. The visualization of one of the attack examples is shown in Figure 2 in Appendix A.4.

5 Conclusion and Future Works

In this paper, we introduce semialgebraic representations of monDEQ and propose several POP models that are useful for certifying robustness, estimating Lipschitz constants and computing outer-approximation ellipsoids. For each model, there are several hierarchies of relaxations that allow us to improve the results by increasing the relaxation order. Even though we simply consider the order-1 relaxation, we obtain tighter upper bounds of Lipschitz constants compared to the results in [25]. Consequently, we are able to certify robustness of more examples.

Our models are based on SDP relaxation, hence requires an efficient SDP solver. However, the state-of-the-art SDP solver Moisek (by interior-point method) can only handle PSD matrices of moderate size (smaller than 5000). This is the main limitation of our method if the dimension of the input gets larger. Moreover, we only consider the fully-connected monDEQ based on MNIST datasets for
One important and interesting future work is to generalize our model to single and multi convolutional monDEQ, and to other datasets such as CIFAR [20] and SVHN [24]. A convolutional layer can be regarded as a fully-connected layer with a larger sparse weight matrix. Hence one is able to build similar models via sparse polynomial optimization tools.

The authors in [25] showed that we can train DEQs with small Lipschitz constants for the \( L_2 \) norm, by controlling the monotonicity of the weight matrix. This guarantees the robustness of monDEQ w.r.t. the \( L_2 \) norm but not for the \( L_\infty \) norm. A natural investigation track is to adapt this training technique to the \( L_\infty \) norm with a better control of the associated Lipschitz constant.

**Acknowledgments and Disclosure of Funding**

This work has benefited from the Tremplin ERC Stg Grant ANR-18-ERC2-0004-01 (T-COPS project), the European Union’s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Actions, grant agreement 813211 (POEMA) as well as from the AI Interdisciplinary Institute ANITI funding, through the French “Investing for the Future PIA3” program under the Grant agreement n° ANR-19-PI3A-0004. The third author was supported by the FMJH Program PGMO (EPICS project) and EDF, Thales, Orange et Criteo. The fourth author acknowledge the support of Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant numbers FA9550-19-1-7026, FA9550-18-1-0226, and ANR MasDol.

**References**

[1] Vegard Antun, Nina M. Gottschling, Anders C. Hansen, and Ben Adcock. Deep learning in scientific computing: Understanding the instability mystery. *SIAM News March 2021*, 2021.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Multiscale deep equilibrium models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5238–5250. Curran Associates, Inc., 2020.

[4] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[5] Jérôme Bolte and Edouard Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, pages 1–33, 2020.

[6] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3240–3247, 2019.

[7] Tong Chen, Jean B Lasserre, Victor Magron, and Edouard Pauwels. Semialgebraic optimization for lipschitz constants of relu networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19189–19200. Curran Associates, Inc., 2020.

[8] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley New York, 1983.

[9] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM review*, 59(2):295–320, 2017.

[10] Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Probabilistic verification and reachability analysis of neural networks via semidefinite programming. In 2019 IEEE 58th Conference on Decision and Control (CDC), pages 2726–2731, 2019.

[11] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
[12] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[14] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

[15] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[16] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[17] Haimin Hu, Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Reach-sdp: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5929–5934. IEEE, 2020.

[18] Todd Huster, Cho-Yu Jason Chiang, and Ritu Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 16–29. Springer, 2018.

[19] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

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

[21] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[22] Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *International Conference on Learning Representations*, 2020.

[23] ApS Mosek. The mosek optimization toolbox for matlab manual, 2015.

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[25] Chirag Pabbaraju, Ezra Winston, and J. Zico Kolter. Estimating lipschitz constants of monotone deep equilibrium models. In *International Conference on Learning Representations*, 2021.

[26] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[27] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

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

[29] Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607, 2020.

[30] Gaurav Sharma and Jos Martin. Matlab®: a language for parallel computing. *International Journal of Parallel Programming*, 37(1):3–36, 2009.
A Appendix

A.1 Proof of Lemma 1

Definition 1 (Clarke’s generalized Jacobian) \[8\] Let \( f : \mathbb{R}^n \to \mathbb{R}^m \) be a locally Lipschitz vector-valued function, denote by \( \Omega_f \) any zero measure set such that \( f \) is differentiable outside \( \Omega_f \). For \( x \notin \Omega_f \), denote by \( \mathcal{J}_f(x) \) the Jacobian matrix of \( f \) evaluated at \( x \). For any \( x \in \mathbb{R}^n \), the generalized Jacobian, or Clarke Jacobian, of \( f \) evaluated at \( x \), denoted by \( \mathcal{J}^C_f(x) \), is defined as the convex hull of all \( m \times n \) matrices obtained as the limit of a sequence of the form \( \mathcal{J}_f(x_i) \) with \( x_i \to x \) and \( x_i \notin \Omega_f \). Symbolically, one has

\[
\mathcal{J}^C_f(x) := \text{conv}\{\lim \mathcal{J}_f(x_i) : x_i \to x, x_i \notin \Omega_f\}.
\]

In order to estimate the Lipschitz constant \( L_q^{F,S} \), we need the following lemma:

Lemma 3 Let \( F : \mathbb{R}^{p_0} \to \mathbb{R}^K \), \( x \mapsto \text{Cz}(x) \) be the fully-connected monDEQ. Its Lipschitz constant is upper bounded by the supremum of the operator norm of its generalized Jacobian, i.e., define

\[
L_q^{F,S} := \sup_{t \in \mathbb{R}^{p_0}, v, w \in \mathbb{R}^K, J \in \mathcal{J}_C^C(x)} \left\{ t^T J^T C^T v : \|t\|_q \leq 1, w^T v \leq 1, \|w\|_q \leq 1, x \in S \right\},
\]

then \( L_q^{F,S} \leq L_q^\mathcal{E}^{F,S} \).
Proof : Since \( z(x) = \text{ReLU}(Wz(x) + Ux + u) \) by definition of monDEQ, \( z(x) \) is Lipschitz according to [25] Theorem 1. Furthermore, \( z(x) \) is semialgebraic by the semialgebraicity of ReLU in (1). Therefore, the Clarke Jacobian of \( z \) is conservative. Indeed by [8] Proposition 2.6.2, the Clarke Jacobian is included in the product of subgradients of its coordinates which is a conservative field by [5] Lemma 3, Theorems 2 and 3. Since \( F = C \circ z \), the mapping \( CJ_{\bar{z}} : x \to CJ \), where \( J \in J_{\bar{z}}^{C} \), is conservative for \( F \) by [5] Lemma 5. So it satisfies an integration formula along segments. Let \( x_1, x_2 \in \mathcal{E} \), and let \( \gamma : [0, 1] \to \mathbb{R}^p \) be a parametrization of the segment defined by \( \gamma(t) = x_1 + t(x_2 - x_1) \) (which is absolutely continuous). For almost all \( t \in [0, 1] \), we have \( \frac{d}{dt} F(\gamma(t)) = CJ \gamma'(t) = CJ(x_2 - x_1) \) for all \( J \in J_{\bar{z}}^{C}(\gamma(t)) \).

Let \( M = \sup_{x \in S, t \in J^{\mathcal{C}}(x)} \| CJ \|_q \) be the suprema of the operator norm \( \| CJ \|_q \) for all \( J \in J_{\bar{z}}^{C}(x) \) and all \( x \in S \). We prove that \( M < +\infty \). Indeed, \( z(x) \) is Lipschitz, hence there exists \( N > 0 \) such that \( \| J \|_q < N \) for all \( J \in J_{\bar{z}}^{C}(x) \) and all \( x \in S \). The value \( M \) is thus upper bounded by \( \| C \|_q N \).

Therefore, for almost all \( t \in [0, 1] \), \( \| \frac{d}{dt} F(\gamma(t)) \|_q \leq M \| x_2 - x_1 \|_q \), and by integration,

\[
\| F(x_2) - F(x_1) \|_q = \left\| \int_0^1 \frac{d}{dt} F(\gamma(t))dt \right\|_q \leq \int_0^1 \left\| \frac{d}{dt} F(\gamma(t)) \right\|_q dt \leq M \| x_2 - x_1 \|_q , \tag{7}
\]

which proves that \( L^{q}_{F, S} \leq M \). Let us show that \( M = L^{q}_{F, S} \). Fix \( x \in \mathbb{R}^p \) and \( J \in J_{\bar{z}}^{C}(x) \). By the definition of operator norm,

\[
\| CJ \|_q = \| (CJ)^T \|_q^{*} = \max_{v \in \mathbb{R}^q} \{ t^T CJ^T v : \| t \|_q \leq 1, \| v \|_q^{*} \leq 1 \}
\]

\[
= \max_{t \in \mathbb{R}^p, v \in \mathbb{R}^q} \{ t^T CJ^T v : \| t \|_q \leq 1, w^T v \leq 1, \| w \|_q \leq 1 \} , \tag{8}
\]

where \( \| \cdot \|_q^{*} \) denotes the dual norm of \( \| \cdot \|_q \) defined by \( \| v \|_q^{*} := \sup_{w \in \mathbb{R}^p} \{ w^T v : \| w \|_q \leq 1 \} \)

for all \( v \in \mathbb{R}^p \), and the first equality is due to the fact that the operator norm of matrix \( CJ \) induced by norm \( \| \cdot \|_q \) is equal to the operator norm of its transpose \( (CJ)^T \) induced by the dual norm \( \| \cdot \|_q^{*} \).

Indeed, by definition of operator norm and dual norm, we have

\[
\| CJ \|_q = \sup_{x \in \mathbb{R}^p} \{ \| CJx \|_q : \| x \|_q \leq 1 \} = \sup_{x \in \mathbb{R}^p, y \in \mathbb{R}^p} \{ y^T CJx : \| x \|_q \leq 1, \| y \|_q^{*} \leq 1 \}
\]

\[
= \max_{t \in \mathbb{R}^p, v \in \mathbb{R}^q} \{ t^T CJ^T y : \| x \|_q \leq 1, \| y \|_q^{*} \leq 1 \} = \max_{t \in \mathbb{R}^p} \{ t^T CJ^T y : \| t \|_q \leq 1, \| y \|_q^{*} \leq 1 \} = \| (CJ)^T \|_q^{*} .
\]

The quantity \( L^{q}_{F, S} \) is just the maximization of Equation (8) for all \( x \in \mathbb{R}^p \) and all \( J \in J_{\bar{z}}^{C}(x) \) and therefore equals \( M \).

The function \( z \) is semialgebraic, and therefore, there exists a closed zero measure set \( \Omega_z \) such that \( z \) is continuously differentiable on the complement of \( \Omega_z \). For any \( x \notin \Omega_z \), since \( z \) is \( C^1 \) at \( x \), we have \( J_{\bar{z}}^{C}(x) = \{ J_{\bar{z}}(x) \} \) by definition of the Clarke Jacobian. Fix \( x \notin \Omega_z \) arbitrary. According to the Corollary of Theorem 2.6.6, on page 75 of [8], we have

\[
J_{\bar{z}}^{C}(x) \subseteq \text{conv} \{ J_{\text{ReLU}}^{C}(Wz(x) + Ux + u) \cdot J_{\bar{z}}^{C}(Wz(x) + Ux + u) \}
\]

\[
= \text{conv} \{ J_{\text{ReLU}}^{C}(Wz(x) + Ux + u) \cdot (W \cdot J_{\bar{z}}(x) + U) \}
\]

\[
= J_{\text{ReLU}}^{C}(Wz(x) + Ux + u) \cdot (W \cdot J_{\bar{z}}(x) + U) , \tag{9}
\]

where the first inclusion is from the cited Corollary, the first equality is because \( z \) is \( C^1 \) at \( x \) so that the chain rule applies, and the last one is because the Clarke Jacobian is convex.

Fix any \( \bar{x} \in \mathbb{R}^p \), then by definition \( J_{\bar{z}}^{C}(\bar{x}) = \text{conv} \{ \lim_{i \to \infty} J_{\bar{z}}(x_i) : x_i \to \bar{x}, i \to +\infty, x_i \notin \Omega_z \} \).

Let \( \{ x_i \}_{i \in \mathbb{N}} \) be a sequence not in \( \Omega_z \) converging to \( \bar{x} \), for each \( x_i \notin \Omega_z \), we have by (9) that \( J_{\bar{z}}(x_i) \in J_{\text{ReLU}}^{C}(Wz(x_i) + Ux_i + u) \cdot (W \cdot J_{\bar{z}}(x_i) + U) \), i.e., there exists \( Y_i \in J_{\text{ReLU}}^{C}(Wz(x_i) + Ux_i + u) \) such that \( J_{\bar{z}}(x_i) = Y_i (W \cdot J_{\bar{z}}(x_i) + U) \).

By [8] Proposition 2.6.2(b), \( J_{\text{ReLU}}^{C} \) has closed graph. Therefore, by continuity of \( z \), up to a subsequence, \( Y_i \to Y \in J_{\text{ReLU}}^{C}(Wz(x) + Ux + u) \) for \( i \to +\infty \), which means

\[
J_{\bar{z}}^{C}(\bar{x}) \subseteq \{ J : Y \in J_{\text{ReLU}}^{C}(Wz(x) + Ux + u), J = Y(WJ + U) \} , \tag{10}
\]
for all $x \in \mathbb{R}^{p_0}$. Let $Y \in \mathcal{J}_{\text{ReLU}}^C(Wz + Ux + u)$, since we have coordinate-wise applications of ReLU, we have that $Y = \text{diag}(s)$ with $s \in \partial \text{ReLU}(Wz + Ux + u)$. By equation (10), the right-hand side of equation (6) is upper bounded by

$$
\max_{t, x \in \mathbb{R}^{p_0}, s \in \mathbb{R}, y, \bar{s} \in \mathbb{R}^K, z \in \mathbb{R}^{p_0}} \{t^T J C^T v : \|t\|_q \leq 1, w^T v \leq 1, \|w\|_q \leq 1, x \in S, s \in \partial \text{ReLU}(Wz + Ux + u), z = \text{ReLU}(Wz + Ux + u), J = \text{diag}(s) \cdot (W \cdot J + U)\}.
$$

(LipMON-a)

Notice that in problem (LipMON-a), we have a matrix variable $J$ of size $p \times p_0$, i.e., containing $p \times p_0$ many variables, which is too large for any SDP solvers. To reduce the size, we use the vector-matrix product trick introduced in [36] to reduce the size of the unknown variables. From equation $J = \text{diag}(s) \cdot (W \cdot J + U)$, we have $J = (I_p - \text{diag}(s) \cdot W)^{-1} \cdot \text{diag}(s) \cdot U$. This inversion makes sense because of the strong monotonicity of $I_p - W$ and the fact that all entries of $s$ lie in $[0, 1]$. Hence

$$
v^T CJ = v^T C \cdot (I_p - \text{diag}(s) \cdot W)^{-1} \cdot \text{diag}(s) \cdot U = r^T \cdot \text{diag}(s) \cdot U,
$$

(11)

where $r^T = v^T C \cdot (I_p - \text{diag}(s) \cdot W)^{-1}$, which means $r - W^T \cdot \text{diag}(s) \cdot r = C^T v$. Set $y = \text{diag}(s) \cdot r$ and transpose both sides of equation (11), we have $J^T C^T v = U^T y$ with $r - W^T \cdot y = C^T v$. We can then rewrite the objective function of (LipMON-a) as $t^T U^T y$, leading to the following equivalent problem

$$
\max_{t, x \in \mathbb{R}^{p_0}, s \in \mathbb{R}, y, \bar{s} \in \mathbb{R}^K, z \in \mathbb{R}^{p_0}} \{t^T U^T y : \|t\|_q \leq 1, w^T v \leq 1, \|w\|_q \leq 1, x \in S, s \in \partial \text{ReLU}(Wz + Ux + u), z = \text{ReLU}(Wz + Ux + u), r - W^T y = C^T v, y = \text{diag}(s) \cdot r\}.
$$

(LipMON-b)

We have shown that (LipMON-b) is the right hand side of Equation (LipMON) in Lemma 1 and is an upper bound of the right hand side of Equation (6) in Lemma 3, i.e., $\tilde{L}_{P,S}^{T} \leq \tilde{L}_{P,S}^{T}$.

### A.2 Redundant Constraints of the Lipschitz Model

In order to avoid possible numerical issues of problem (LipMON), and to improve the bounds, we add some redundant constraints to it. For variables $r$ and $y$, Note that $r = (I_p - W^T \cdot \text{diag}(s))^{-1} \cdot C^T v$, hence $\|r\|_2 \leq \|I_p - W^T \cdot \text{diag}(s)\|^{-1}_2 \|\|C^T v\|_2 \cdot \|v\|_2$. The operator norm of a matrix induced by $L_2$ norm is its largest singular value. Hence the operator norm of $(I_p - W^T \cdot \text{diag}(s))^{-1}$ is the smallest singular value of matrix $I_p - W^T \cdot \text{diag}(s)$, which is smaller or equal than 1 from the recent work [36]. In summary, we have $\|r\|_2 \leq \|C\|_2 \cdot \|v\|_2$ and $\|y\|_2 \leq \|C\|_2 \cdot \|v\|_2$. For Lipschitz Model w.r.t. $L_2$ norm, we have $\|v\|_2 \leq 1$; for Lipschitz Model w.r.t. $L_\infty$ norm, we have $\|v\|_\infty \leq \|v\|_1 \leq 1$, thus $\|v\|_2 \leq \|v\|_1 \leq 1$. Therefore, for both $L_2$ and $L_\infty$ norm, we can bound the $L_2$ norm of variables $r$ and $y$ by $\|C\|_2$. Moreover, we multiply the equality constraint $r - W^T \cdot y = C^T v$ coordinate-wisely with variables $s, z, y, r$ to produce redundant constraints and improve the results. This strengthening technique is already included in the software Gloptipoly3 [16]. With all the discussion above, we now write the strengthened version of problem (LipMON-b) as follows:

$$
\max_{t, x \in \mathbb{R}^{p_0}, s \in \mathbb{R}, y, \bar{s} \in \mathbb{R}^K, z \in \mathbb{R}^{p_0}} \{t^T U^T y : \|t\|_q \leq 1, w^T v \leq 1, \|w\|_q \leq 1, x \in S, s \in \partial \text{ReLU}(Wz + Ux + u), z = \text{ReLU}(Wz + Ux + u), r - W^T y = C^T v, y = \text{diag}(s) \cdot r, \|y\|_2 \leq \|C\|_2 \cdot \|v\|_2, \|r\|_2 \leq \|C\|_2 \cdot \|v\|_2, s(r - W^T y) = s(C^T v), z(r - W^T y) = z(C^T v), y(r - W^T y) = y(C^T v), r(r - W^T y) = r(C^T v)\}.
$$

(LipMON-c)
A.3 Proof of Lemma 2

The SOS constraint in problem (EllipMON-SOS-\text{\text{-}}) can be written as

\[
\sigma_0(x, z) = -\left(\|Q(Cz + c) + b\|^2 - 1 \quad (=: f_1(x, z))
+ \sigma_1(x, z)^T g_q(x - x_0) \quad (=: f_2(x, z))
+ \tau(x, z)^T (z - Wz - Ux - u) \quad (=: f_3(x, z))
+ \sigma_2(x, z)^T (z - Wz - Ux - u) \quad (=: f_4(x, z))
+ \sigma_3(x, z)^T z \quad (=: f_5(x, z))\right)
= -(f_1(x, z) + f_2(x, z) + f_3(x, z) + f_4(x, z) + f_5(x, z)) = -f(x, z).
\]

For \(d = 1\), denote by \(M_i\) the Gram matrix of polynomial \(f_i(x, z)\) for \(i = 1, \ldots, 5\) and \(M\) the Gram matrix of polynomial \(f(x, z)\), with basis \([x^T, z^T, 1]\). We have explicitly \(M = \sum_{i=1}^5 M_i\), where \(M_i\) has the following form

\[
M_1 = \begin{bmatrix}
0_{p_0 \times p_0} & 0_{p_0 \times p} & 0_{p_0 \times 1} \\
p_{p_0 \times p_0} & C^T Q^2 C & C^T Q^2 c + C^T Q b \\
0 & c^T Q^2 C + b^T Q C & c^T Q^2 c + 2b^T Q c + b^T b - 1
\end{bmatrix},
\]

\[
M_2 = \begin{bmatrix}
-\text{diag}(\sigma_1) & 0_{p_0 \times p_0} & \text{diag}(\sigma_1) \cdot x_0 \\
0_{p_0 \times p_0} & 0_{p_0 \times p_0} & 0_{p_0 \times 1} \\
\text{diag}(\sigma_1) \cdot x_0^T & 0_{1 \times p_0} & \sigma_1^T (\varepsilon^2 - x_0^2)
\end{bmatrix}, \quad \text{for } L_\infty\text{-norm},
\]

\[
M_3 = \begin{bmatrix}
0_{p_0 \times p_0} & -\frac{1}{2} U^T \text{diag}(\tau) & 0_{p_0 \times 1} \\
-\frac{1}{2} \text{diag}(\tau) U & \text{diag}(\tau) (I_p - W) & -\frac{1}{2} \text{diag}(\tau) \cdot u \\
0_{1 \times p_0} & -\frac{1}{2} u^T \cdot \text{diag}(\tau) & 0
\end{bmatrix},
\]

\[
M_4 = \begin{bmatrix}
0_{p_0 \times p_0} & 0_{p_0 \times p_0} & -\frac{1}{2} U^T \sigma_3 \\
0_{p_0 \times p_0} & 0_{p_0 \times p_0} & \frac{1}{2} (I_p - W^T) \sigma_3 \\
-\frac{1}{2} \sigma_3 & \frac{1}{2} \sigma_3 & 0
\end{bmatrix},
\]

\[
M_5 = \begin{bmatrix}
0_{p_0 \times p_0} & 0_{p_0 \times p} & 0_{p_0 \times 1} \\
0_{p_0 \times p} & 0_{p_0 \times p} & \frac{1}{2} \sigma_2 \\
0_{1 \times p_0} & 0_{1 \times p} & 0
\end{bmatrix}.
\]

Moreover, in order to improve the quality of the ellipsoid, we can also use the slope restriction condition of ReLU function as proposed in [17]: \((z_j - z_i)(W_j; z + U_j; x + u_j - W_{ij}; z - U_{ij}; x - u_i) - (z_j - z_i)^2 \geq 0\) for \(i \neq j\). The Gram matrix of the SOS combination of these constraints with basis \([x^T, z^T, 1]\) has the form

\[
M_6 = \begin{bmatrix}
U & W & u \\
0_{p \times p} & I_p & 0_{p \times 1} \\
0_{1 \times p} & 0_{1 \times p} & 1
\end{bmatrix}^T \begin{bmatrix}
0_{p \times p} & T & 0_{p \times 1} \\
T & -2T & 0_{p \times 1} \\
0_{1 \times p} & 0_{1 \times p} & 0
\end{bmatrix} \begin{bmatrix}
U & W & u \\
0_{p \times p} & I_p & 0_{p \times 1} \\
0_{1 \times p} & 0_{1 \times p} & 1
\end{bmatrix},
\]

where \(T = \sum_{i=1}^{p-1} \sum_{j=i+1}^p \lambda_{ij} (e_i - e_j)(e_i - e_j)^T\) with \(\lambda_{ij} \geq 0\) for all \(i < j\), and \(\{e_i\}_{i=1}^p \subseteq \mathbb{R}^p\) is the canonical basis of \(\mathbb{R}^p\). Since \(\sigma_0(x, z)\) is an SOS polynomial of degree at most 2, we conclude that \(-M \succeq 0\). According to Lemma 5 in [10], the constraint \(-M \succeq 0\) is equivalent to an SDP constraint using Schur complements, which finishes the proof of Lemma 2.
A.4 An Adversarial Example

Figure 2: An adversarial example of the first test MNIST input found by PGD algorithm for $L_\infty$ norm with $\varepsilon = 0.1$.

(a) Original example, classified as 7  (b) Adversarial example, classified as 3

A.5 Licenses of Used Assets

| Software  | License                                      |
|-----------|----------------------------------------------|
| Julia     | MIT License                                  |
| JuMP      | Mozilla Public License                       |
| Matlab    | Proprietary Software                         |
| CVX       | CVX Standard License                         |
| Python    | Python Software Foundation License           |
| Pytorch   | Berkeley Software Distribution               |
| Mosek     | Proprietary Software                         |
| Our code  | CeCILL Free Software License                 |