# Exact Semidefinite Relaxations for Safety Verification of Neural Network

Godai Azuma (Aoyama Gakuin University)

Joint work with Sunyoung Kim (Ewha W. University)
Makoto Yamashita (Institute of Science Tokyo)

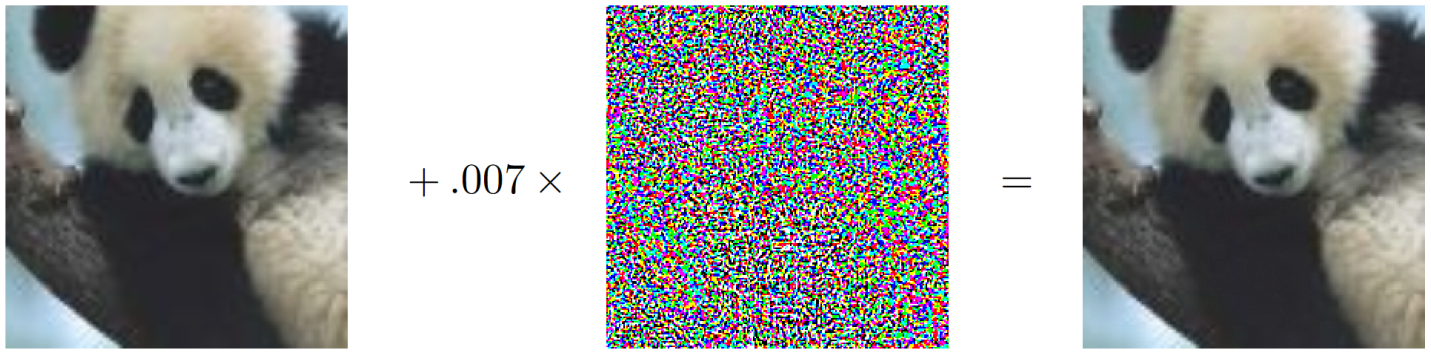
ICCOPT 2025 in Los Angeles (July 22, 2025)

This work was supported by JSPS KAKENHI JP24K20738, JP22KJ1307 and JP20H04145
NRF 2021-R1A2C1003810

# Vulnerability on Neural Networks (NNs)

Uncertainty and adversarial attacks

- Panda image + small noise [GSS14]




- Stop sign + optical attacks $\Longrightarrow$ Speed limit 60 sign

a barrier to applications where reliability is critical (e.g., self-driving cars)

[GSS14] Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015, 2014.

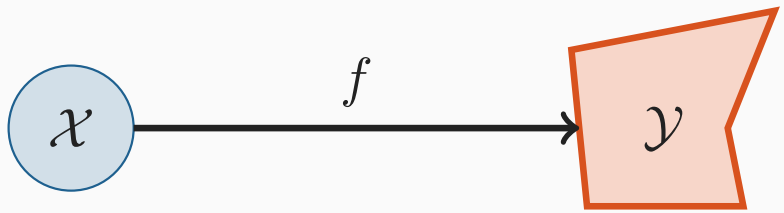
# Verifying Safety of NNs

Let $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ be a NN.

Input set $\mathcal{X} \subseteq \mathbb{R}^{n_0}$

we wish to evaluate

Output $\mathcal{Y} := \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}$




Let $S_y$: a set where no misclassification occurs = safety specification set

**Def: Safety Verification**

Check whether $\mathcal{Y} \subseteq S_y$ holds or not for a given $S_y$

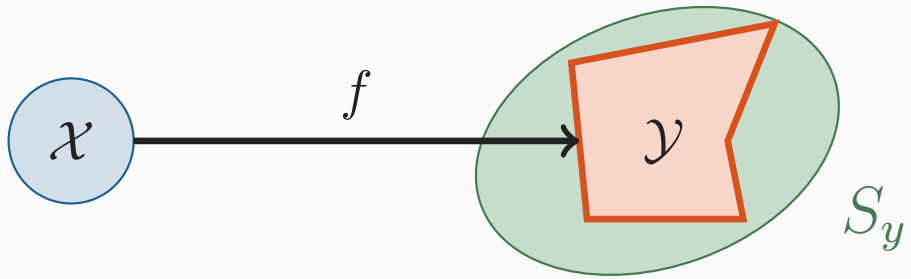
# Verifying Safety of NNs

Let $f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ be a NN.

Input set $\mathcal{X} \subseteq \mathbb{R}^{n_0}$

we wish to evaluate

Output $\mathcal{Y} := \{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}$




Let $S_y$: a set where no misclassification occurs = safety specification set

**Def: Safety Verification**

Check whether $\mathcal{Y} \subseteq S_y$ holds or not for a given $S_y$

# DeepSDP

Semidefinite programming-based method of safety verification.

- $M \succeq O \iff M$ is positive semidefinite.

## General Formulation of DeepSDP

$$\begin{aligned} \min_{P,Q,S} \quad & g(P,Q,S) \\ \text{s.t.} \quad & M_{\text{in}}(P) + M_{\text{mid}}(Q) + M_{\text{out}}(S) \succeq O, \\ & P \in \mathcal{P}_{\mathcal{X}},\ Q \in \mathcal{Q}_{\phi},\ S \in \mathcal{S}. \end{aligned}$$

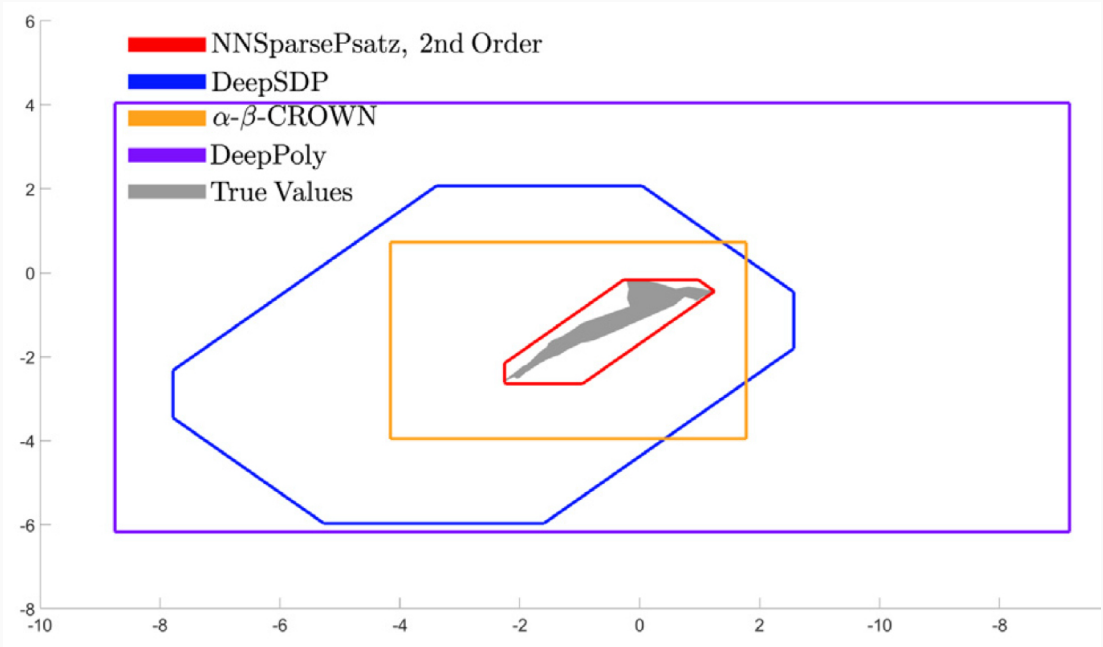
The above problem is less accurate than other methods [NP21].

---

[NP21] Newton and Papachristodoulou, Neural network verification using polynomial optimisation, IEEE CDC, 2021.

# Accuracy of DeepSDP with Other Method

$\hat{\mathcal{Y}}$: Estimated output set $\leftrightarrow$ $\mathcal{Y}$: True output set (gray) [NP21]




- All methods overestimate $\hat{\mathcal{Y}}$ in order to cover the whole output in $\mathcal{Y}$.
- Accurate verification favors a smaller $\hat{\mathcal{Y}}$.

**Motivation**

What conditions make DeepSDP highly accurate?

We address it by using an exact relaxation.

# Outline

- Introduction

- Quadratically constrained quadratic programming
- Exact semidefinite programming (SDP) relaxation

- Single-layer feed-forward neural network
- Formulation of DeepSDP for safety verification

- Exact SDP relaxation in safety verification
- Proof

- Summary

# QCQP: Quadratically Constrained Quadratic Programming

Consider a quadratic programming with quadratic constraints:

$$\begin{aligned} v^* \coloneqq \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} + 2\left(\boldsymbol{q}^p\right)^{\mathrm{T}} \boldsymbol{x} \leq b_p, \quad p \in [m] \coloneqq \{1, \ldots, m\}. \end{aligned} \tag{$\mathcal{P}$}$$

**Used in**

Binary programming, MAX-CUT, optimal flow problems,...

- Behind the safety verification
- Generally non-convex & NP-hard
- Approximately solvable via SDP relaxation

# Semidefinite Programming (SDP) Relaxation

Define two notation:

- $Q^p \bullet X := \sum_{i,j} Q^p_{ij} X_{ij}$: Frobenius inner product.
- $X \succeq \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \iff X - \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$ is positive semidefinite.

QCQP

$$v^* = \min \left\{ \boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} + 2(\boldsymbol{q}^p)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \right\} \qquad (\mathcal{P})$$

# Semidefinite Programming (SDP) Relaxation

Define two notation:

- $Q^p \bullet X := \sum_{i,j} Q^p_{ij} X_{ij}$: Frobenius inner product.
- $X \succeq \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \iff X - \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$ is positive semidefinite.

QCQP

$$v^* = \min\left\{ \boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} + 2(\boldsymbol{q}^p)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \right\} \tag{$\mathcal{P}$}$$

$$= \min\left\{ Q^0 \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \begin{array}{l} \boxed{X = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}} \\ Q^p \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \end{array} \right\}$$

# Semidefinite Programming (SDP) Relaxation

Define two notation:

- $Q^p \bullet X := \sum_{i,j} Q^p_{ij} X_{ij}$: Frobenius inner product.
- $X \succeq \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \iff X - \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$ is positive semidefinite.

QCQP

$$v^* = \min\left\{ \boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} + 2\left(\boldsymbol{q}^p\right)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \right\} \tag{$\mathcal{P}$}$$

$$= \min\left\{ Q^0 \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \begin{array}{l} \boxed{X = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}} \\ Q^p \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \end{array} \right\}$$

$$\ge \min\left\{ Q^0 \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \,\middle|\, \begin{array}{l} \boxed{X \succeq \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}} \\ Q^p \bullet X + 2\left(\boldsymbol{q}^0\right)^{\mathrm{T}} \boldsymbol{x} \le b_p,\ p \in [m] \end{array} \right\} \tag{$\mathcal{P}_{\mathrm{R}}$}$$

Semidefinite Programming (SDP) Relaxation

$$=: v^*_{\mathrm{SDP}}$$

# Exact SDP Relaxation

**Def: Exactness**

SDP relaxation ($\mathcal{P}_{\mathrm{R}}$) is exact (tight) if $v^* = v^*_{\mathrm{SDP}}$

i.e., $$\min\left\{\boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} \,\middle|\, \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} \le b_p,\ p \in \{1,\dots,m\}\right\} \qquad (\mathcal{P})$$

$$= \min\left\{ Q^0 \bullet X \,\middle|\, \begin{array}{l} X \succeq \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \\ Q^p \bullet X \le b_p,\ p \in \{1,\dots,m\} \end{array} \right\} \qquad (\mathcal{P}_{\mathrm{R}})$$

- Sufficient conditions for exactness have been studied
- Exact $\iff$ ($\mathcal{P}_{\mathrm{R}}$) has a rank-1 solution $X^*$
  - Decomposition $\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathrm{T}} = X^*$ exists
  - $\hat{\boldsymbol{x}}$ is a solution of QCQP ($\mathcal{P}$)

# Outline

- Introduction
- Quadratically constrained quadratic programming
- Exact semidefinite programming (SDP) relaxation
- Single-layer feed-forward neural network
- Formulation of DeepSDP for safety verification
- Exact SDP relaxation in safety verification
- Proof
- Summary

# Single-layer Neural Networks

$W^0, W^1$: weight matrices, $\boldsymbol{b}^0, \boldsymbol{b}^1$: bias vectors

**Neural network**

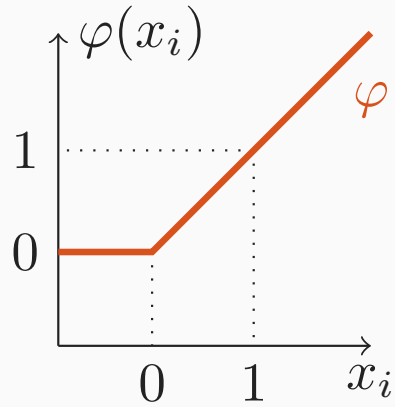
$$\boldsymbol{x}^1 := \phi(W^0 \boldsymbol{x}^0 + \boldsymbol{b}^0),$$
$$f(\boldsymbol{x}^0) := W^1 \boldsymbol{x}^1 + \boldsymbol{b}^1.$$

Note we consider the case that

- $W^1 = I$, and $\boldsymbol{b}^1 = \boldsymbol{0}$.
- $\phi$ is an element-wise ReLU function, i.e.,

$$\phi(\boldsymbol{x}) := \begin{bmatrix} \varphi(x_1) & \cdots & \varphi(x_n) \end{bmatrix}^{\mathrm{T}},$$

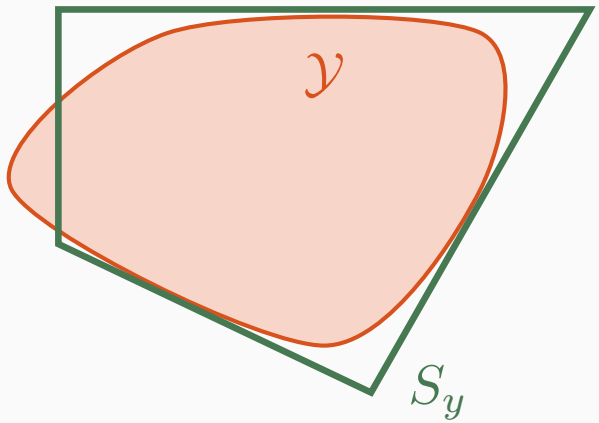
where $\varphi(x_i) := \max\{0, x_i\}$.



# Polytope Safety Specification Set

Consider polytope safety specification set $S_y$

- Let $S_y$ be a quadrilateral below.
- $\mathcal{Y} \subseteq S_y$ can be verified via four half-spaces.



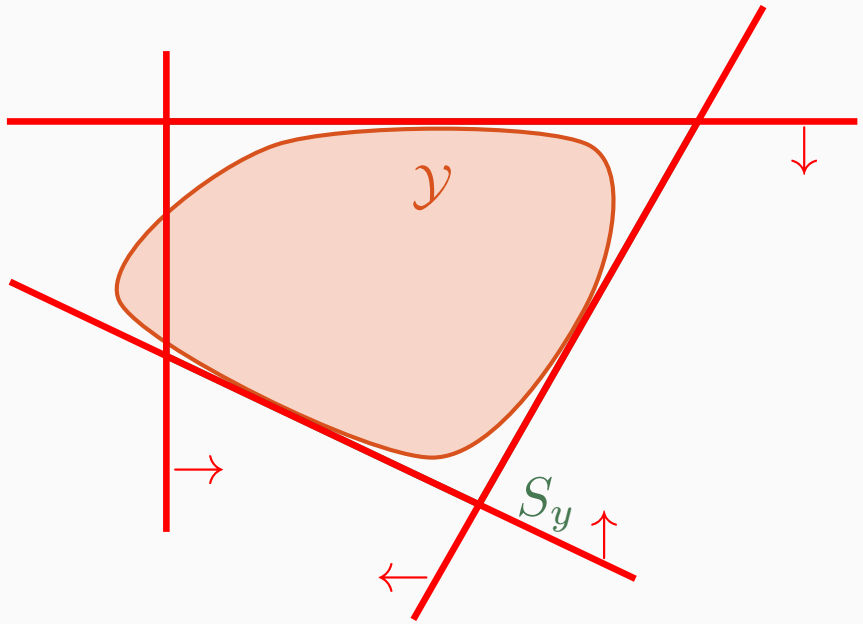

**Setting I**

Assume that safety specification set $S_y$ is a half-space.

# Polytope Safety Specification Set

Consider polytope safety specification set $S_y$

- Let $S_y$ be a quadrilateral below.
- $\mathcal{Y} \subseteq S_y$ can be verified via four half-spaces.




**Setting I**

Assume that safety specification set $S_y$ is a half-space.

# DeepSDP for Single-layer NN

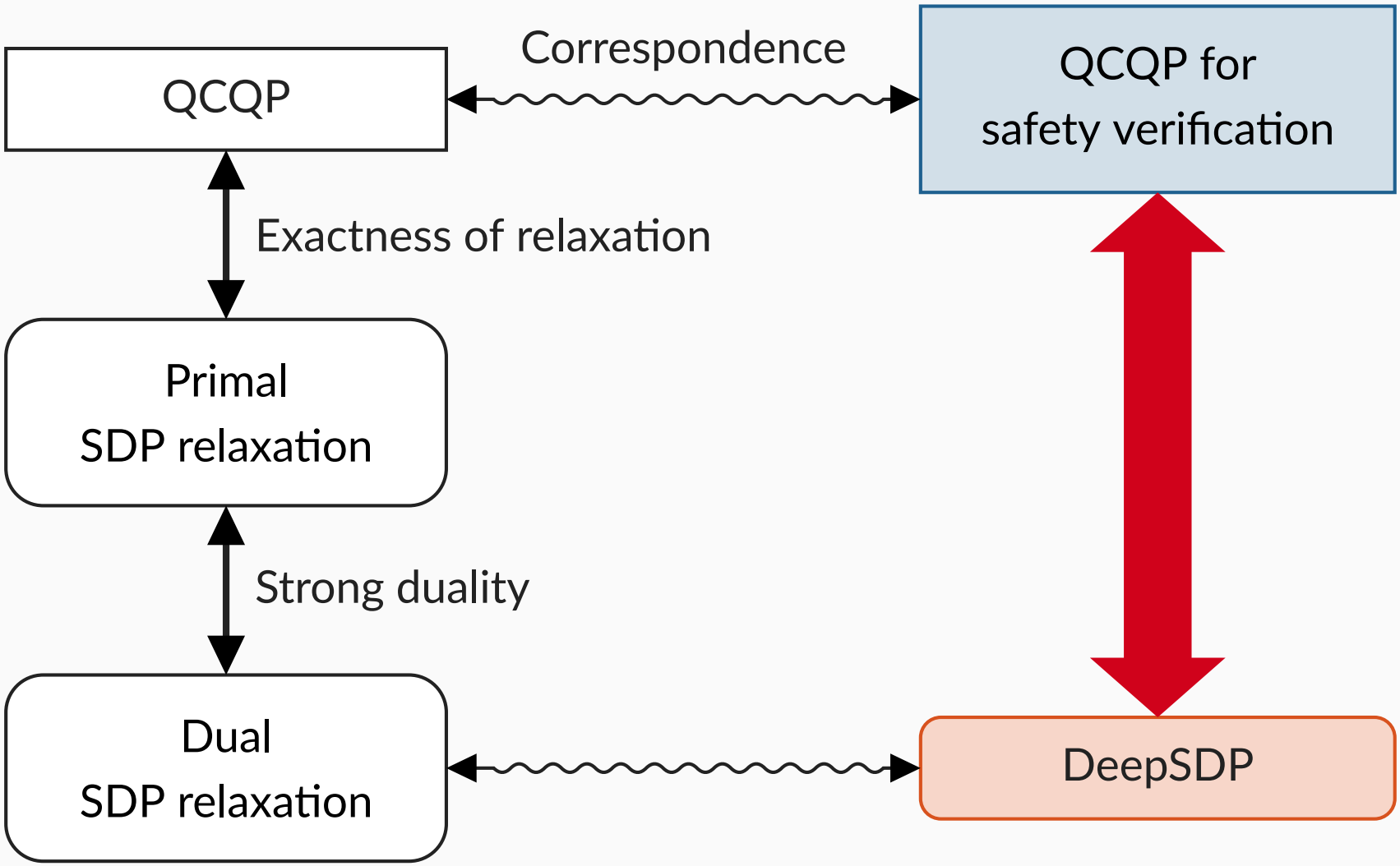
Consider a half-space $H := \{\boldsymbol{y} \in \mathbb{R}^{n_2} \mid \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - d \geq 0\}$.

## DeepSDP to verify $\mathcal{Y} \subseteq H$

$$
\begin{aligned}
\max_{\gamma,\boldsymbol{\lambda},\boldsymbol{\nu},\boldsymbol{\eta},d} \quad & 2d \\
\text{s.t.} \quad & \gamma \begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \boldsymbol{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \boldsymbol{0} & O & O \end{bmatrix} + \begin{bmatrix} -2d & \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{c}^{\mathrm{T}} \\ \boldsymbol{0} & O & O \\ \boldsymbol{c} & O & O \end{bmatrix} \\
& \quad + \begin{bmatrix} 0 & \boldsymbol{\nu}^{\mathrm{T}}W^0 & -\boldsymbol{\nu}^{\mathrm{T}} - \boldsymbol{\eta}^{\mathrm{T}} \\ (W^0)^{\mathrm{T}}\boldsymbol{\nu} & O & -(W^0)^{\mathrm{T}}\operatorname{diag}(\boldsymbol{\lambda}) \\ -\boldsymbol{\nu} - \boldsymbol{\eta} & -\operatorname{diag}(\boldsymbol{\lambda})W^0 & 2\operatorname{diag}(\boldsymbol{\lambda}) \end{bmatrix} \succeq O, \\
& \gamma \in \mathbb{R}_+, \quad \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\eta} \in \mathbb{R}^n_+, \quad d \in \mathbb{R}.
\end{aligned}
$$

- An SDP, solvable in polynomial-time
- Generated from a QCQP by relaxing and taking the dual

# Exactness Around Today's Problems




**Exactness and Accuracy**

Exact relaxation allows DeepSDP to solve the original QCQP.

# Primal SDP Relaxation (= Dual of DeepSDP)

$e^i$: a vector where $i$th element is $1$, the others are $0$.

$$
\begin{aligned}
\min_{\substack{\boldsymbol{x}^0,\boldsymbol{x}^1,\\ X^{00},X^{10},X^{11}}} \quad & 2\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^1 \\
\text{s.t.} \quad & \begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \mathbf{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \mathbf{0} & O & O \end{bmatrix} \bullet G \le 0,\ G := \begin{bmatrix} 1 & \left(\boldsymbol{x}^0\right)^{\mathrm{T}} & \left(\boldsymbol{x}^1\right)^{\mathrm{T}} \\ \boldsymbol{x}^0 & X^{00} & \left(X^{10}\right)^{\mathrm{T}} \\ \boldsymbol{x}^1 & X^{10} & X^{11} \end{bmatrix}, \\
& \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & -b_i^0\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \mathbf{0} & O & -\left(W^0\right)^{\mathrm{T}}\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ -b_i^0\boldsymbol{e}^i & -\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}}W^0 & 2\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \end{bmatrix} \bullet G \le 0, \\
& \begin{bmatrix} 2b_i^0 & \left(\boldsymbol{e}^i\right)^{\mathrm{T}}W^0 & -\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \left(W^0\right)^{\mathrm{T}}\boldsymbol{e}^i & O & O \\ -\boldsymbol{e}^i & O & O \end{bmatrix} \bullet G \le 0, \\
& \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & -\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \mathbf{0} & O & O \\ -\boldsymbol{e}^i & O & O \end{bmatrix} \bullet G \le 0, \qquad i = 1,\ldots,n_1.
\end{aligned}
$$

# Outline

- Introduction
- Quadratically constrained quadratic programming
- Exact semidefinite programming (SDP) relaxation
- Single-layer feed-forward neural network
- Formulation of DeepSDP for safety verification
- Exact SDP relaxation in safety verification
- Proof
- Summary

# Theoretical Result

**Thm: Exactness condition for a single-layer network**

The primal SDP relaxation is exact if

- $\mathcal{X} = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 \leq \rho\}$; or
- $\mathcal{X} = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_\infty \leq \rho\}$, and $W^0 = I$.

In addition, DeepSDP is also exact under strong duality.

- This talk focuses on the first sufficient condition (hyper-ellipsoid).
- We discuss the derivation in the remaining time, via
  - transformation using vector $e$,
  - decomposition to two problems.

# Gram Matrix Transformation

Fix $\boldsymbol{e} \in \mathbb{R}^{1+n_0+n_1}$ satisfying $\|\boldsymbol{e}\| = 1$. (arbitrary)

- Define new variables $\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n_0}$, and $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^{n_1} \in \mathbb{R}^{1+n_0+n_1}$.
- Substitute $\boldsymbol{x}^0, \boldsymbol{x}^1, X_{00}, X_{10}, X_{11}$ in the primal SDP by

$$\boldsymbol{x}^0 = \begin{bmatrix} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{u}^1 \\ \vdots \\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{u}^{n_0} \end{bmatrix} \in \mathbb{R}^{n_0}, \quad X^{10} = \begin{bmatrix} (\boldsymbol{v}^1)^{\mathrm{T}}\boldsymbol{u}^1 & \cdots\cdots & (\boldsymbol{v}^1)^{\mathrm{T}}\boldsymbol{u}^{n_0} \\ \vdots & \ddots\ddots & \vdots \\ (\boldsymbol{v}^{n_1})^{\mathrm{T}}\boldsymbol{u}^1 & \cdots\cdots & (\boldsymbol{v}^{n_1})^{\mathrm{T}}\boldsymbol{u}^{n_0} \end{bmatrix} \in \mathbb{R}^{n_1 \times n_0},$$

$$\boldsymbol{x}^1 = \begin{bmatrix} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^1 \\ \vdots \\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^{n_1} \end{bmatrix} \in \mathbb{R}^{n_1}, \quad X^{11} = \begin{bmatrix} (\boldsymbol{v}^1)^{\mathrm{T}}\boldsymbol{v}^1 & \cdots\cdots & (\boldsymbol{v}^1)^{\mathrm{T}}\boldsymbol{v}^{n_1} \\ \vdots & \ddots\ddots & \vdots \\ (\boldsymbol{v}^{n_1})^{\mathrm{T}}\boldsymbol{v}^1 & \cdots\cdots & (\boldsymbol{v}^{n_1})^{\mathrm{T}}\boldsymbol{v}^{n_1} \end{bmatrix} \in \mathbb{S}^{n_1}.$$

Example

$$0 \geq \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & -(\boldsymbol{e}^i)^{\mathrm{T}} \\ \mathbf{0} & O & O \\ -\boldsymbol{e}^i & O & O \end{bmatrix} \bullet G = -2\boldsymbol{e}^i \bullet \boldsymbol{x}^1 = -2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i \quad \text{for all } i.$$

# Equivalent Formulation of SDP Relaxation

The following problem is obtained from the primal SDP relaxation.

$$\min_{\boldsymbol{u}^j, \boldsymbol{v}^i} \quad 2\sum_{i=1}^{n_1} c_i \, \boldsymbol{e}^{\mathrm{T}} \boldsymbol{v}^i \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{e}^{\mathrm{T}} \boldsymbol{v}^i \geq 0, \qquad i = 1, \ldots, n_1, \tag{2}$$

$$\boldsymbol{e}^{\mathrm{T}} \boldsymbol{v}^i \geq \boldsymbol{e}^{\mathrm{T}} \left( \sum_{j=1}^{n_0} W_{ij} \boldsymbol{u}^j + b_i^0 \boldsymbol{e} \right), \qquad i = 1, \ldots, n_1, \tag{3}$$

$$\left\| \boldsymbol{v}^i \right\|_2^2 \leq \left( \sum_{j=1}^{n_0} W_{ij} \boldsymbol{u}^j + b_i^0 \boldsymbol{e} \right)^{\mathrm{T}} \boldsymbol{v}^i, \quad i = 1, \ldots, n_1, \tag{4}$$

$$\sum_{j=1}^{n_0} \left\| \boldsymbol{u}^j - \hat{x}_j \boldsymbol{e} \right\|_2^2 \leq \rho^2. \tag{5}$$

# Exactness Condition by Collinearity with $e$

**Prop: Exactness condition for collinearity in [Zhang '20]**

Suppose there exists an optimal solution $\{(\boldsymbol{u}^1)^*, \ldots, (\boldsymbol{u}^{n_0})^*, (\boldsymbol{v}^1)^*, \ldots, (\boldsymbol{v}^{n_1})^*\}$ which are collinear with $\boldsymbol{e}$. Then, the primal SDP relaxation is exact.

**Def: Collinearity**

Vectors $\{\boldsymbol{a}^1, \ldots, \boldsymbol{a}^m\} \subseteq \mathbb{R}^{1+n_0+n_1}$ are collinear with $\boldsymbol{e}$ if

$$\left|\boldsymbol{e}^{\mathrm{T}}\boldsymbol{a}^i\right| = \left\|\boldsymbol{a}^i\right\| \quad \text{for all } i \in \{1, \ldots, m\}.$$

It is suffice to show the collinearity of a solution of the transformed problem.

---

[Zhang '20] Zhang, On the tightness of semidefinite relaxations for certifying robustness to adversarial examples, NeurIPS, 2020.

# Decomposition according to $u^j$ and $v^i$

## Inner problem: constraints using $u^j$

$$\Psi(\boldsymbol{v}^1,\ldots,\boldsymbol{v}^{n_1}) :=$$

$$\left.\begin{aligned} \min_{\boldsymbol{u}^1,\ldots,\boldsymbol{u}^{n_0}} \quad & \textstyle\sum_{j=1}^{n_0} \left\|\boldsymbol{u}^j - \hat{x}_j \boldsymbol{e}\right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i \geq \boldsymbol{e}^{\mathrm{T}} \left(\textstyle\sum_{j=1}^{n_0} W_{ij}\boldsymbol{u}^j + b_i^0 \boldsymbol{e}\right), \quad i = 1,\ldots,n_1, \ (3) \\ & \left\|\boldsymbol{v}^i\right\|_2^2 \leq \left(\textstyle\sum_{j=1}^{n_0} W_{ij}\boldsymbol{u}^j + b_i^0 \boldsymbol{e}\right)^{\mathrm{T}} \boldsymbol{v}^i, \ i = 1,\ldots,n_1, \ (4) \end{aligned}\right\} \quad (S_2)$$

## Outer problem: the remains and $\Psi$

$$\left.\begin{aligned} \min_{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^{n_1}} \quad & 2\textstyle\sum_{i=1}^{n_1} c_i \, \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i & & (1) \\ \text{s.t.} \quad & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i \geq 0, & i = 1,\ldots,n_1, \quad & (2) \\ & \Psi(\boldsymbol{v}^1,\ldots,\boldsymbol{v}^{n_1}) \leq \rho^2. & & \end{aligned}\right\} \quad (S_1)$$

The case $\boldsymbol{e} = \boldsymbol{e}^1$ is only considered due to time limitation.

# Relationship Between Their Solutions

A part of KKT condition of $(S_2)$:

$$\begin{bmatrix} \boldsymbol{u}^1 \\ \vdots \\ \boldsymbol{u}^{n_0} \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \boldsymbol{e}^1 \\ \vdots \\ \hat{x}_{n_0} \boldsymbol{e}^1 \end{bmatrix} - \sum_{i=1}^{n_1} \frac{\nu_i}{2} \begin{bmatrix} W_{i1} \boldsymbol{e}^1 \\ \vdots \\ W_{in} \boldsymbol{e}^1 \end{bmatrix} + \sum_{i=1}^{n_1} \frac{\lambda_i}{2} \begin{bmatrix} W_{i1} \boldsymbol{v}^i \\ \vdots \\ W_{in} \boldsymbol{v}^i \end{bmatrix}$$

**Lemma: Linear Combination**

For any optimal solution $(\boldsymbol{u}^1)^*, \ldots, (\boldsymbol{u}^{n_0})^*$ of $(S_2)$,
there exist $\boldsymbol{m} \in \mathbb{R}^{n_0}$ and $M \in \mathbb{R}^{n_1 \times n_0}$ such that

$$(\boldsymbol{u}^j)^* = m_j \boldsymbol{e}^1 + \sum_{i=1}^{n_1} M_{ij} \boldsymbol{v}^i \quad \text{for each } j \in \{1, \ldots, n_0\}.$$

It suffices to show $(\boldsymbol{v}^i)^*$s are collinear to $\boldsymbol{e}^1$.

# Collinearity in $(S_1)$

$$
\left.
\begin{aligned}
\min_{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^{n_1}} \quad & 2\sum_{i=1}^{n_1} c_i\, \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i && (1)\\
\text{s.t.} \quad & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i \geq 0,\ i=1,\ldots,n_1, && (2)\\
& \sum_{j=1}^{n_0} \left\| (m_j - \hat{x}_j)\,\boldsymbol{e}^1 + \sum_{i=1}^{n_1} M_{ij}\boldsymbol{v}^i \right\|_2^2 \leq \rho^2.
\end{aligned}
\right\} \quad (S_1)
$$

For $(S_1)$, we can show the following lemma:

**Lemma: Collinearity of $(\boldsymbol{v}^i)^*$**

$(S_1)$ has an optimal solution $(\boldsymbol{v}^1)^*,\ldots,(\boldsymbol{v}^{n_1})^*$ which are collinear to $\boldsymbol{e}^1$.

Therefore, the SDP relaxation is exact due to the collinearity.

# Collinearity in $(S_1)$

$$\left.\begin{aligned} \min_{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^{n_1}} \quad & 2\sum_{i=1}^{n_1} c_i\, \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i && (1)\\ \text{s.t.} \quad & \boldsymbol{e}^{\mathrm{T}}\boldsymbol{v}^i \geq 0,\ i=1,\ldots,n_1, && (2)\\ & \sum_{j=1}^{n_0}\left\|(m_j-\hat{x}_j)\,\boldsymbol{e}^1+\sum_{i=1}^{n_1} M_{ij}\boldsymbol{v}^i\right\|_2^2 \leq \rho^2. \end{aligned}\right\} \quad (S_1)$$

Essence of Proof.

- Let $(\bar{\boldsymbol{v}}^1,\ldots,\bar{\boldsymbol{v}}^{n_1})$ be an optimal solution of $(S_1)$.
- Assume at least one of $\bar{\boldsymbol{v}}^1,\ldots,\bar{\boldsymbol{v}}^{n_1}$ is not collinear to $\boldsymbol{e}^1$.
- Define

$$\hat{\boldsymbol{v}}^i := \left[\bar{v}_1^i,\ 0,\ \cdots,\ 0\right]^{\mathrm{T}}.$$

- Then, $(\hat{\boldsymbol{v}}^1,\ldots,\hat{\boldsymbol{v}}^{n_1})$ is another optimal solution.
  - The objective value is the same.

# Conclusion

Summary

- Verification of the safety of Neural Networks
- Exactness conditions of DeepSDP in this context

Future works

- Analyze the non-polyhedral case of $S_y$
- Weaken the assumptions on the input set $\mathcal{X}$

Thank you for your attention!

For more details, see arXiv:2504.09934

*Tight Semidefinite Relaxations for Verifying Robustness of Neural Networks*

# QCQP: Quadratically Constrained Quadratic Programming

Consider a quadratic programming with quadratic constraints:

$$\begin{aligned} v^* \coloneqq \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{x}^{\mathrm{T}} Q^0 \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{x}^{\mathrm{T}} Q^p \boldsymbol{x} \leq b_p, \quad p \in [m] \coloneqq \{1, \ldots, m\}, \end{aligned} \tag{$\mathcal{P}$}$$

- Generally, non-convex & NP-hard
- Semidefinite programming (SDP) relaxation

**Applications**

Binary programming, MAX-CUT, optimal flow problems,...

# Exactness for SDP Relaxation

The following equality holds:

$$v^* = \min \left\{ Q^0 \bullet X \;\middle|\; \begin{array}{l} X \succeq O \\ Q^p \bullet X \leq b_p \quad \forall p \in [m] \end{array} \right\} = v^*_{\mathrm{SDP}}$$

$\Longleftrightarrow$ rank-1 solution $X^*$ exists

Exact SDP relaxation $\Longrightarrow$

- Original QCQP is exactly solvable (in theorically)
- The gap between a class of QCQPs and their relaxations is identified.

**Motivation**

What conditions of QCQPs guarantee the exactness?

# Constraints for Hidden Layers

$$\begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & O & -\boldsymbol{e}^i \left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \mathbf{0} & -\boldsymbol{e}^i \left(\boldsymbol{e}^i\right)^{\mathrm{T}} & 2\boldsymbol{e}^i \left(\boldsymbol{e}^i\right)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix} = 0, \quad i = 1, \ldots, N, \tag{6a}$$

$$\begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & \left(\boldsymbol{e}^i\right)^{\mathrm{T}} & -\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \boldsymbol{e}^i & O & O \\ -\boldsymbol{e}^i & O & O \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix} \leq 0, \quad i = 1, \ldots, N, \tag{6b}$$

$$\begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & -\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \mathbf{0} & O & O \\ -\boldsymbol{e}^i & O & O \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix} \leq 0, \quad i = 1, \ldots, N. \tag{6c}$$

# Valid Cuts

Let $\boldsymbol{w}^0 = W^0\boldsymbol{x}^0 + \boldsymbol{b}^0$.

## Valid Cuts for ReLU

The following inequation always holds

$$\left[\phi(w_j^0) - \phi(w_i^0)\right]\left[\phi(w_j^0) - \phi(w_i^0) - (w_j - w_i)\right] \leq 0 \quad \forall (i,j) \in \{1,\ldots,n\}^2$$

$$\begin{bmatrix} \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} O & -(\boldsymbol{e}^i - \boldsymbol{e}_j)(\boldsymbol{e}^i - \boldsymbol{e}_j)^{\mathrm{T}} \\ -(\boldsymbol{e}^i - \boldsymbol{e}_j)(\boldsymbol{e}^i - \boldsymbol{e}_j)^{\mathrm{T}} & 2(\boldsymbol{e}^i - \boldsymbol{e}_j)(\boldsymbol{e}^i - \boldsymbol{e}_j)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix} \leq 0, \quad (7)$$

# Input Constraint

Since $\boldsymbol{x}^0 \in \mathcal{X} = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 \leq \rho\}$.

In QCQP

$$\left\|\boldsymbol{x}^0 - \hat{\boldsymbol{x}}\right\|_2^2 \leq \rho^2$$

In SDP relaxation

$$\begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \mathbf{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \mathbf{0} & O & O \end{bmatrix} \bullet \underbrace{\begin{bmatrix} 1 & \left(\boldsymbol{x}^0\right)^{\mathrm{T}} & \left(\boldsymbol{x}^1\right)^{\mathrm{T}} \\ \boldsymbol{x}^0 & X_{00} & X_{10}{}^{\mathrm{T}} \\ \boldsymbol{x}^1 & X_{10} & X_{11} \end{bmatrix}}_{=: G} \leq 0$$

In DeepSDP By introducing a dual variable $\gamma$,

$$\gamma \begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \mathbf{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \mathbf{0} & O & O \end{bmatrix}$$

# DeepSDP for Single-layer NN

Consider a half-space $H := \{\boldsymbol{y} \in \mathbb{R}^{n_2} \mid \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - d \geq 0\}$.

## DeepSDP to verify $\mathcal{Y} \subseteq H$

$$
\begin{aligned}
\max_{\gamma,\boldsymbol{\lambda},\boldsymbol{\nu},\boldsymbol{\eta},d} \quad & 2d \\
\text{s.t.} \quad & \gamma \begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \boldsymbol{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \boldsymbol{0} & O & O \end{bmatrix} + \begin{bmatrix} -2d & \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{c}^{\mathrm{T}} \\ \boldsymbol{0} & O & O \\ \boldsymbol{c} & O & O \end{bmatrix} \\
& \quad + \begin{bmatrix} 0 & \boldsymbol{\nu}^{\mathrm{T}}W^0 & -\boldsymbol{\nu}^{\mathrm{T}} - \boldsymbol{\eta}^{\mathrm{T}} \\ (W^0)^{\mathrm{T}}\boldsymbol{\nu} & O & -(W^0)^{\mathrm{T}}\operatorname{diag}(\boldsymbol{\lambda}) \\ -\boldsymbol{\nu} - \boldsymbol{\eta} & -\operatorname{diag}(\boldsymbol{\lambda})W^0 & 2\operatorname{diag}(\boldsymbol{\lambda}) \end{bmatrix} \succeq O, \\
& \gamma \in \mathbb{R}_+, \quad \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\eta} \in \mathbb{R}^n_+, \quad d \in \mathbb{R}.
\end{aligned}
$$

- An SDP, solvable in polynomial-time
- Generated from a QCQP by relaxing and taking the dual

# Safety Specification Set

Consider a half-space $H := \left\{ \boldsymbol{y} \in \mathbb{R}^{n_2} \,\middle|\, \boldsymbol{c}^{\mathrm{T}} \boldsymbol{y} - d \geq 0 \right\}$.

- The slope $\boldsymbol{c}$ according to each half-space is given.
- The largest $d$ makes $H$ smaller.

In SDP relaxation

$$\boldsymbol{x}^1 \in H \iff \begin{bmatrix} -2d & \mathbf{0}^{\mathrm{T}} & \boldsymbol{c}^{\mathrm{T}} \\ \mathbf{0} & O & O \\ \boldsymbol{c} & O & O \end{bmatrix} \bullet \begin{bmatrix} 1 & \left(\boldsymbol{x}^0\right)^{\mathrm{T}} & \left(\boldsymbol{x}^1\right)^{\mathrm{T}} \\ \boldsymbol{x}^0 & X_{00} & {X_{10}}^{\mathrm{T}} \\ \boldsymbol{x}^1 & X_{10} & X_{11} \end{bmatrix} \leq 0$$

In DeepSDP Let $d$ behave as a dual variable.

# DeepSDP for Single-layer NN

Consider a half-space $H := \{\boldsymbol{y} \in \mathbb{R}^{n_2} \mid \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - d \geq 0\}$.

## DeepSDP to verify $\mathcal{Y} \subseteq H$

$$
\begin{aligned}
\max_{\gamma,\boldsymbol{\lambda},\boldsymbol{\nu},\boldsymbol{\eta},d} \quad & 2d \\
\text{s.t.} \quad & \gamma \begin{bmatrix} \hat{\boldsymbol{x}}^{\mathrm{T}}\hat{\boldsymbol{x}} - \rho^2 & -\hat{\boldsymbol{x}}^{\mathrm{T}} & \boldsymbol{0} \\ -\hat{\boldsymbol{x}} & I & O \\ \boldsymbol{0} & O & O \end{bmatrix} + \begin{bmatrix} -2d & \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{c}^{\mathrm{T}} \\ \boldsymbol{0} & O & O \\ \boldsymbol{c} & O & O \end{bmatrix} \\
& \quad + \begin{bmatrix} 0 & \boldsymbol{\nu}^{\mathrm{T}}W^0 & -\boldsymbol{\nu}^{\mathrm{T}} - \boldsymbol{\eta}^{\mathrm{T}} \\ (W^0)^{\mathrm{T}}\boldsymbol{\nu} & O & -(W^0)^{\mathrm{T}}\operatorname{diag}(\boldsymbol{\lambda}) \\ -\boldsymbol{\nu} - \boldsymbol{\eta} & -\operatorname{diag}(\boldsymbol{\lambda})W^0 & 2\operatorname{diag}(\boldsymbol{\lambda}) \end{bmatrix} \succeq O, \\
& \gamma \in \mathbb{R}_+, \quad \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\eta} \in \mathbb{R}^n_+, \quad d \in \mathbb{R}.
\end{aligned}
$$

- An SDP, solvable in polynomial-time
- Generated from a QCQP by relaxing and taking the dual

# Quadratic Formulation for ReLU Function

Review: $\phi$ applies element-wisely ReLU function $\varphi$

Let $\boldsymbol{w}^0 := W^0\boldsymbol{x}^0 + \boldsymbol{b}^0$. For any $i \in \{1, \ldots, n_1\}$,

$$\varphi(w_i^0) = \max\{0, w_i^0\} \quad \Longleftrightarrow \quad \begin{cases} \varphi(w_i^0)\left(\varphi(w_i^0) - w_i^0\right) \leq 0, \\ \varphi(w_i^0) \geq w_i^0, \quad \varphi(w_i^0) \geq 0. \end{cases}$$

In QCQP The first inequality is

$$\begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & O & -\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \\ \mathbf{0} & -\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}} & 2\boldsymbol{e}^i\left(\boldsymbol{e}^i\right)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{w}^0 \\ \phi(\boldsymbol{w}^0) \end{bmatrix} \leq 0, \quad i = 1, \ldots, n_1.$$

# Transformation of $[1, w^0, \phi(w^0)]^{\mathrm{T}}$

Equivalently, for $i \in \{1, \ldots, n_1\}$,

$$\begin{bmatrix} 1 \\ x^0 \\ x^1 \end{bmatrix}^{\mathrm{T}} \underbrace{\begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ b^0 & W^0 & O \\ \mathbf{0} & O & I \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & O & -e^i (e^i)^{\mathrm{T}} \\ \mathbf{0} & -e^i (e^i)^{\mathrm{T}} & 2e^i (e^i)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ b^0 & W^0 & O \\ \mathbf{0} & O & I \end{bmatrix}}_{=: L_i} \begin{bmatrix} 1 \\ x^0 \\ x^1 \end{bmatrix} \leq$$

In SDP relaxation $\quad L_i \bullet G \leq 0, \quad i \in \{1, \ldots, n_1\}$

In DeepSDP Introducing a dual variable $\boldsymbol{\lambda} \in \mathbb{R}^{n_1}_+$, $\sum_{i=1}^{n_1} \lambda_i L_i$

| Constraint | $\varphi(w_i^0)\left(\varphi(w_i^0) - w_i^0\right) \leq 0$ | $\varphi(w_i^0) \geq w_i^0$ | $\varphi(w_i^0) \geq 0$ |
|---|---|---|---|
| Dual variable | $\lambda_i$ | $\nu_i$ | $\eta_i$ |

# Input Set $\mathcal{X} \subseteq \boldsymbol{R}^{n_0}$

Set $\mathcal{X}$ contains the uncertainty and attacks.

- Each input $\boldsymbol{x}^0$ is chosen from $\mathcal{X}$.
- The safety of $\boldsymbol{x}^0$ is evaluated by $\boldsymbol{S}_y$.

**Note**

$\mathcal{X}$ is not the domain of NN $f$.

Various shapes are possible.

- hyper-ellipsoid $\quad \mathcal{X} = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 \leq \rho\}$.
- hyper-rectangle $\quad \mathcal{X} = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_\infty \leq \rho\}$.

**Setting II**

This talk covers the case where $\mathcal{X}$ is a hyper-ellipsoid.