Static Analysis for Machine Learning

MPRI 2-6: Abstract Interpretation, Application to Verification and Static Analysis

Caterina Urban

January 13th, 2024

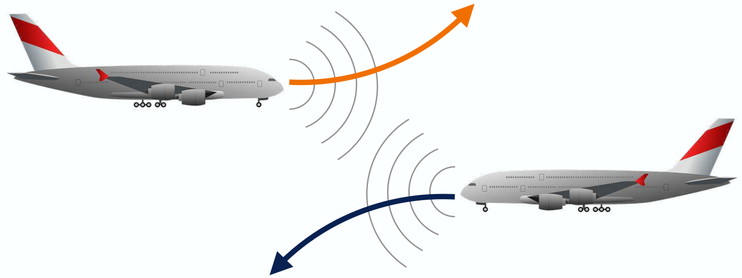
Year 2024-2025

Machine Learning in High-Stakes Systems

data

ML software

perform tasks that are impossible using explicit programming



act as surrogate model

automate decision-making

Machine Learning in High-Stakes Systems

STAT+

IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show

By Casey Ross *@caseymross and* Ike Swetlitz

July 25, 2018

A self-driving Uber ran a red light last December, contrary to company claims

Feds Say Self-Driving Uber SUV Did Not Recognize Jaywalking Pedestrian In Fatal Crash

Richard Gonzales November 7, 201910:57 PM ET

Machine Learning Development Pipeline

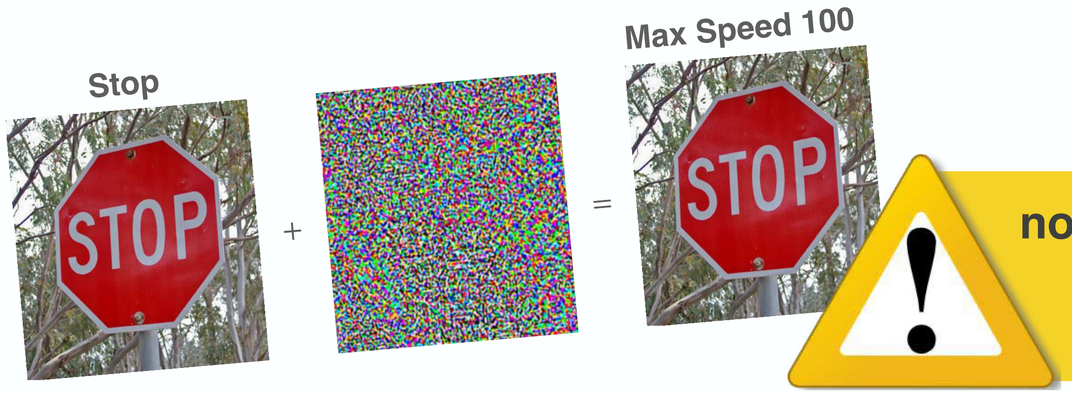
NEXT COURSE

TODAY

Static Analysis for Trained Models

Machine Learning Development Pipeline

Models Only Give Probabilistic Guarantees

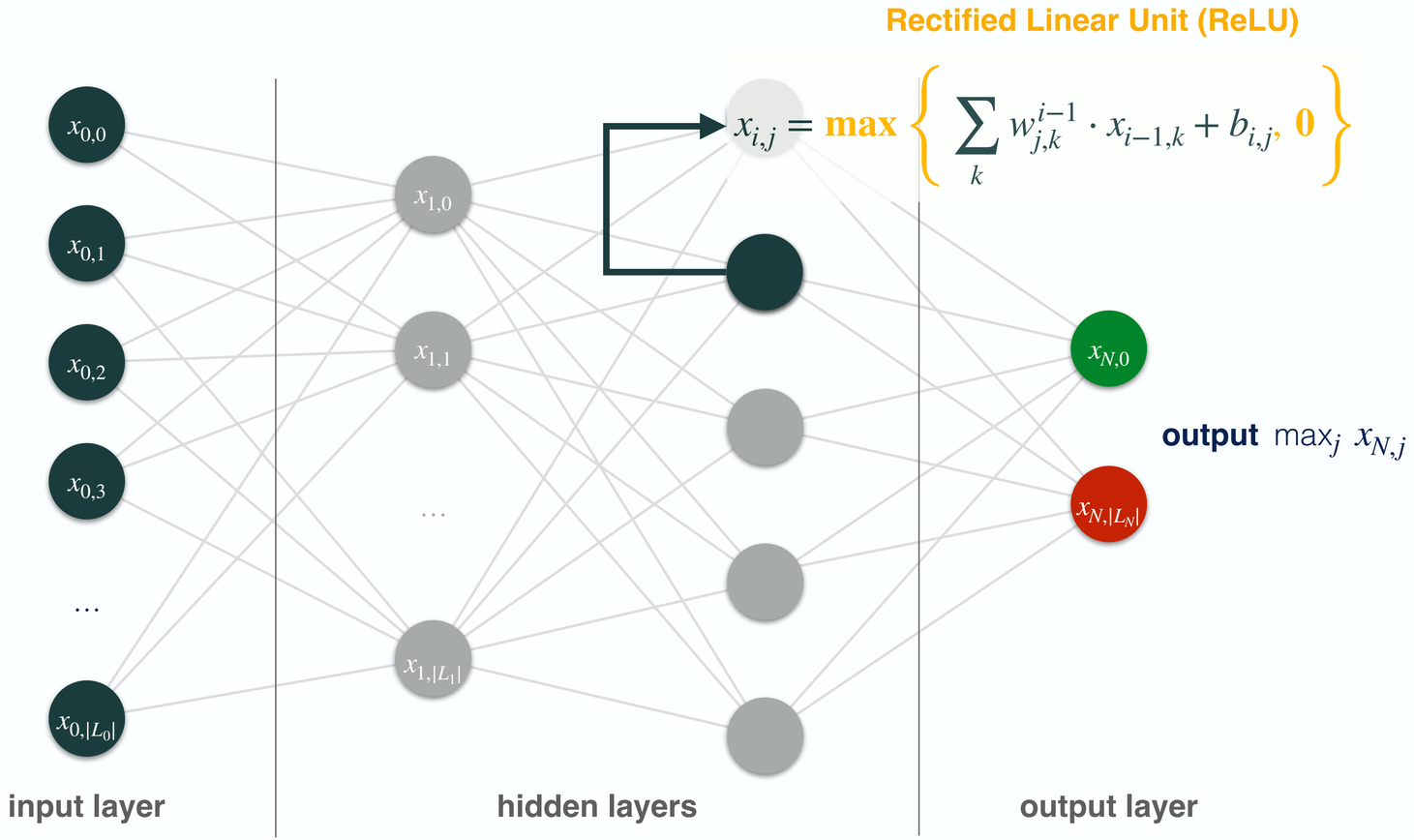


not sufficient for guaranteeing an acceptable failure rate under any circumstance

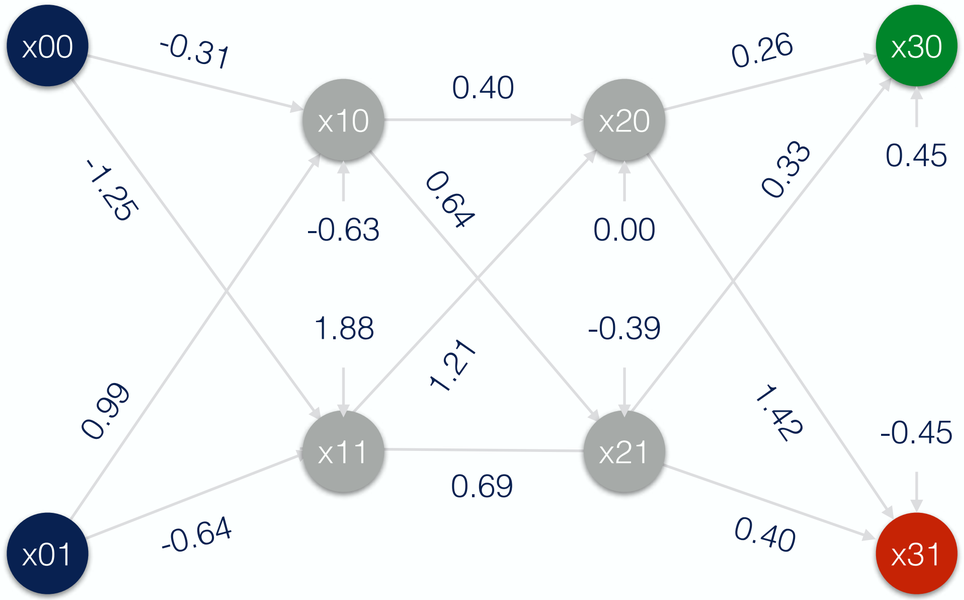
Neural Networks

Neural Networks

Feed-Forward ReLU-Activated Neural Networks

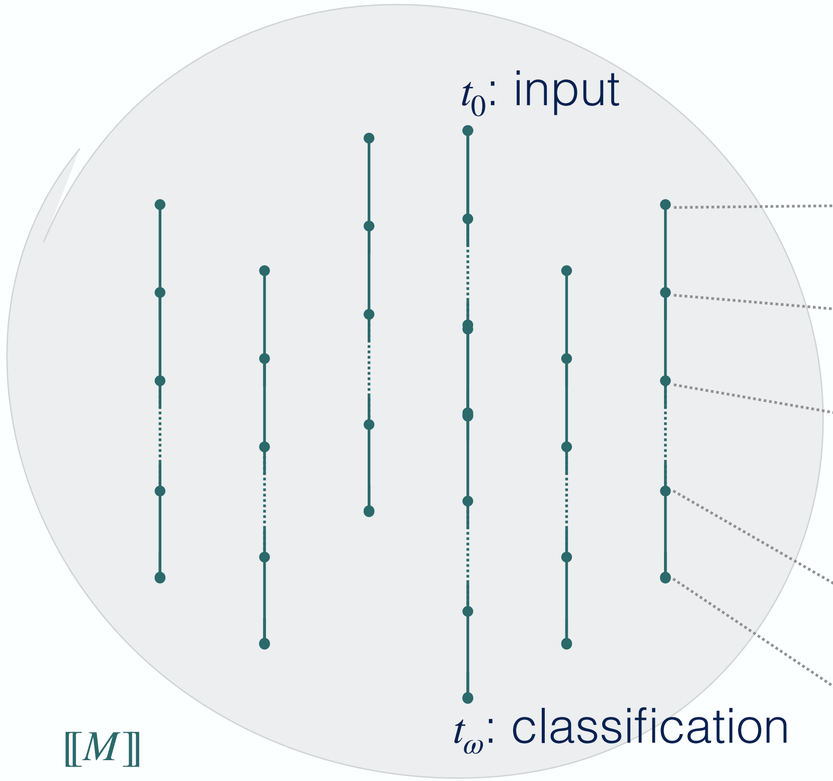


Neural Networks as Programs




```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return ‘ ’ if x31 < 30 else ‘ ’
```

Maximal Trace Semantics

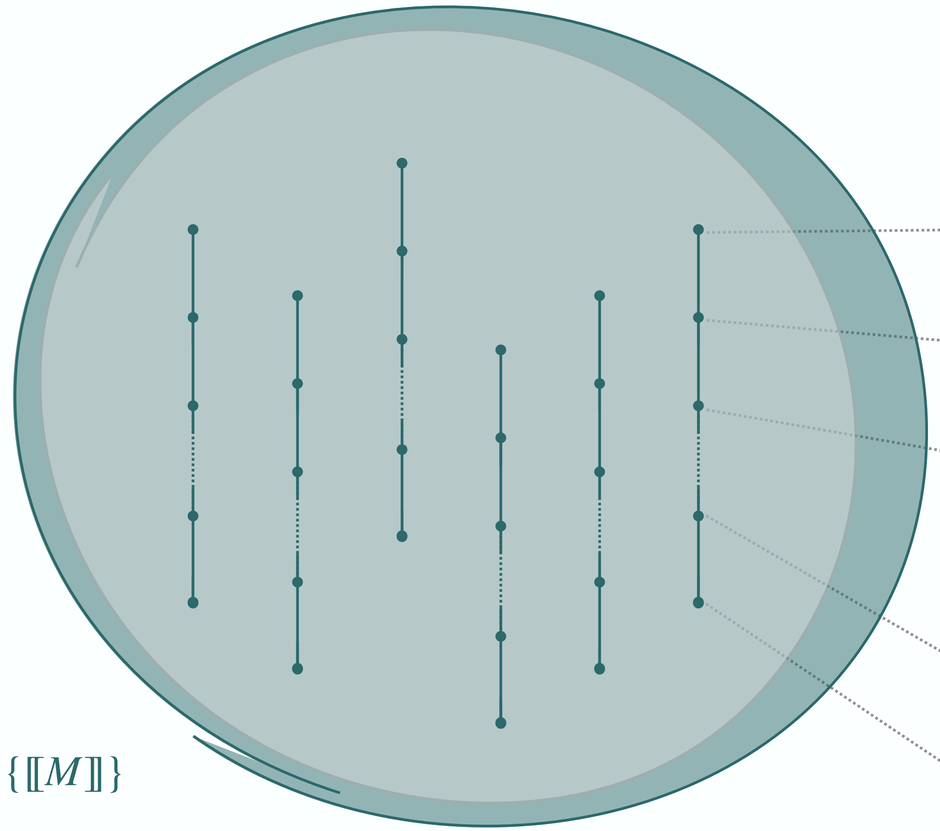


M

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '      ' if x31 < 30 else '      '
```

Neural Network Verification

Collecting Semantics



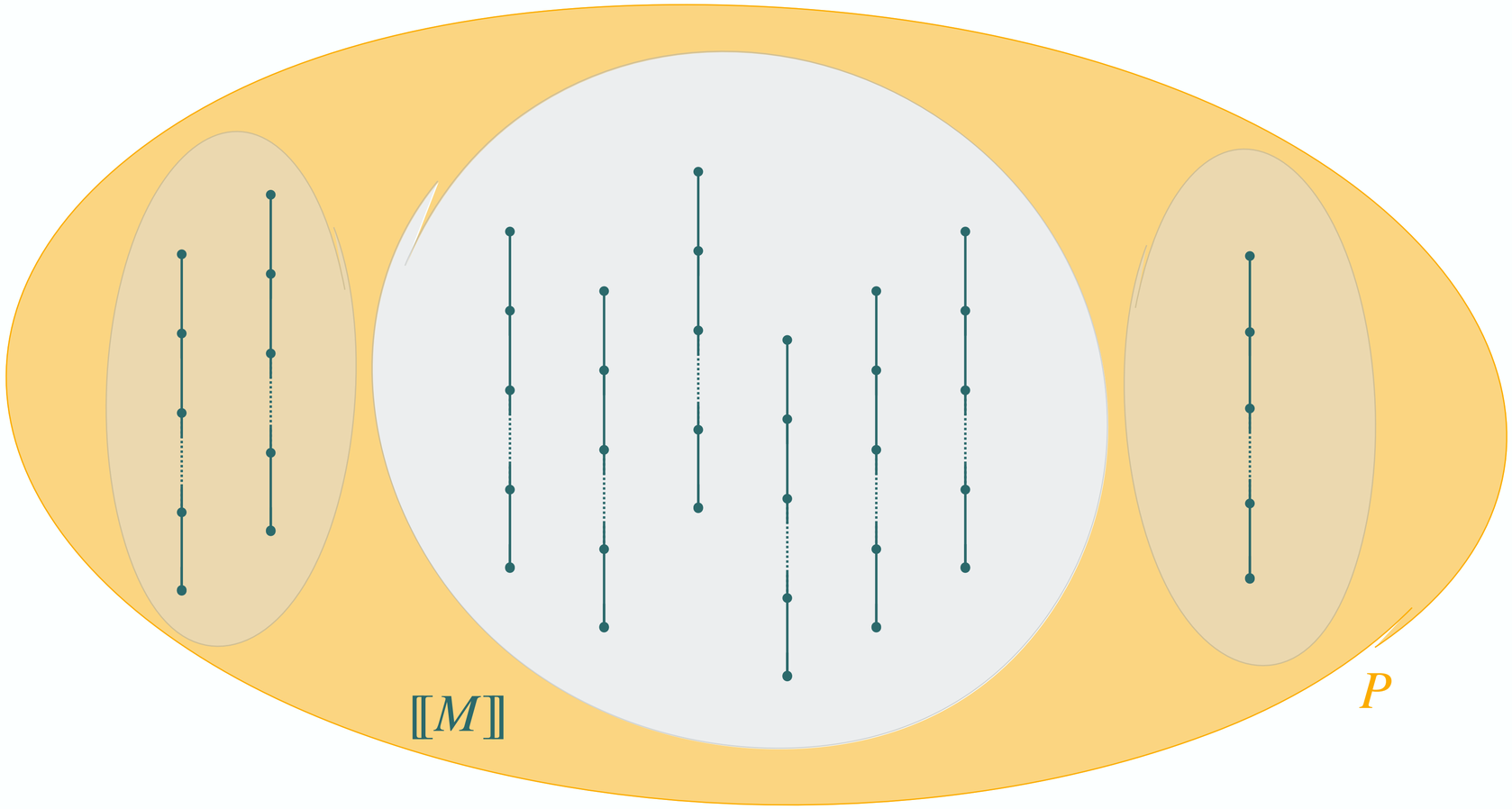
$\{[\![M]\!]\}$

M

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '  ' if x31 < 30 else '  '
```

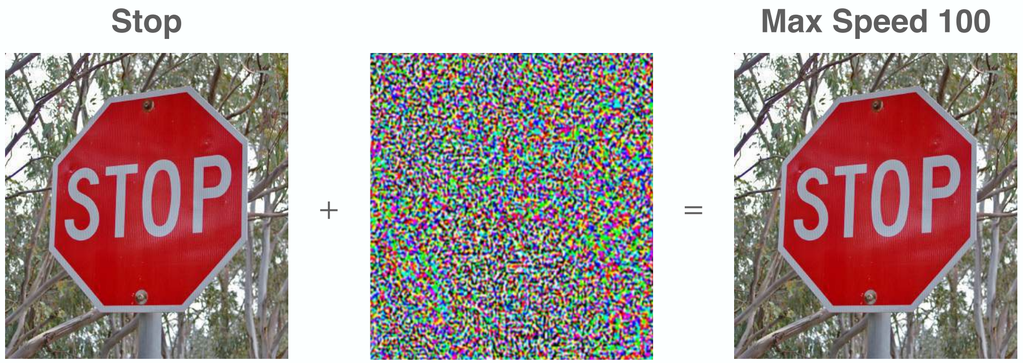
Property Verification

$$[\![M]\!] \in P \Leftrightarrow \{[\![M]\!]\} \subseteq P$$



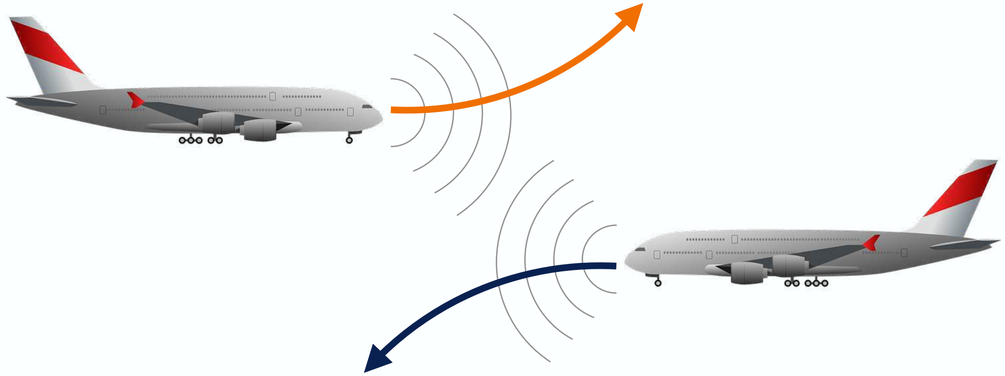
Stability

Goal G3 in [Kurd03]

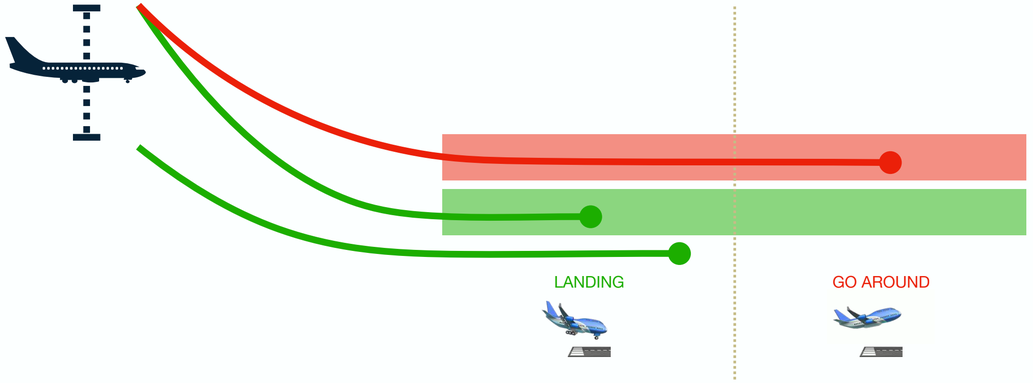


Safety

Goal G4 in [Kurd03]

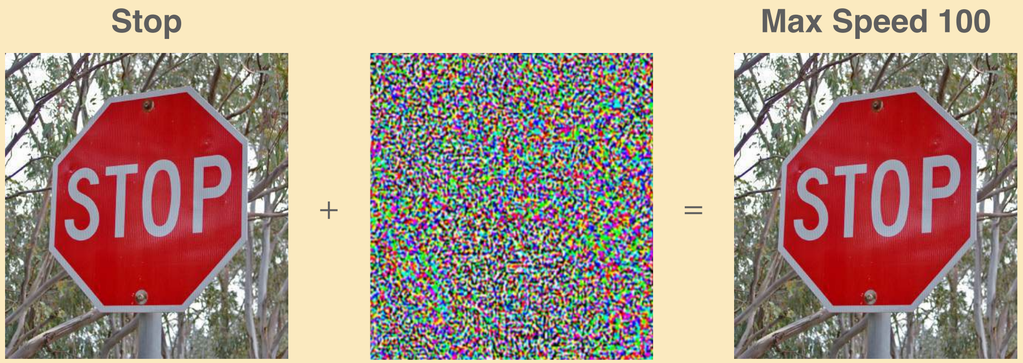


Hypersafety



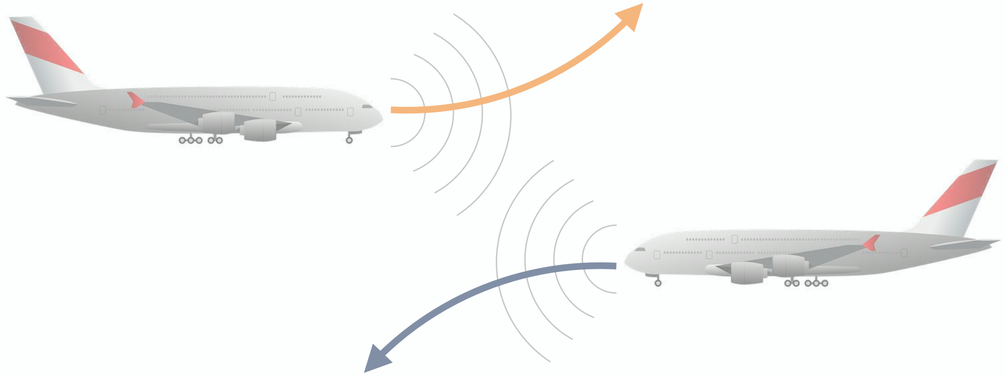
Stability

Goal G3 in [Kurd03]

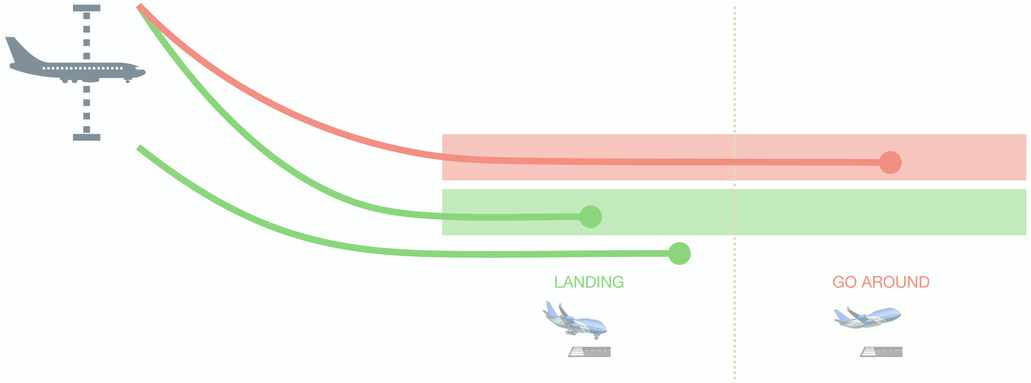


Safety

Goal G4 in [Kurd03]



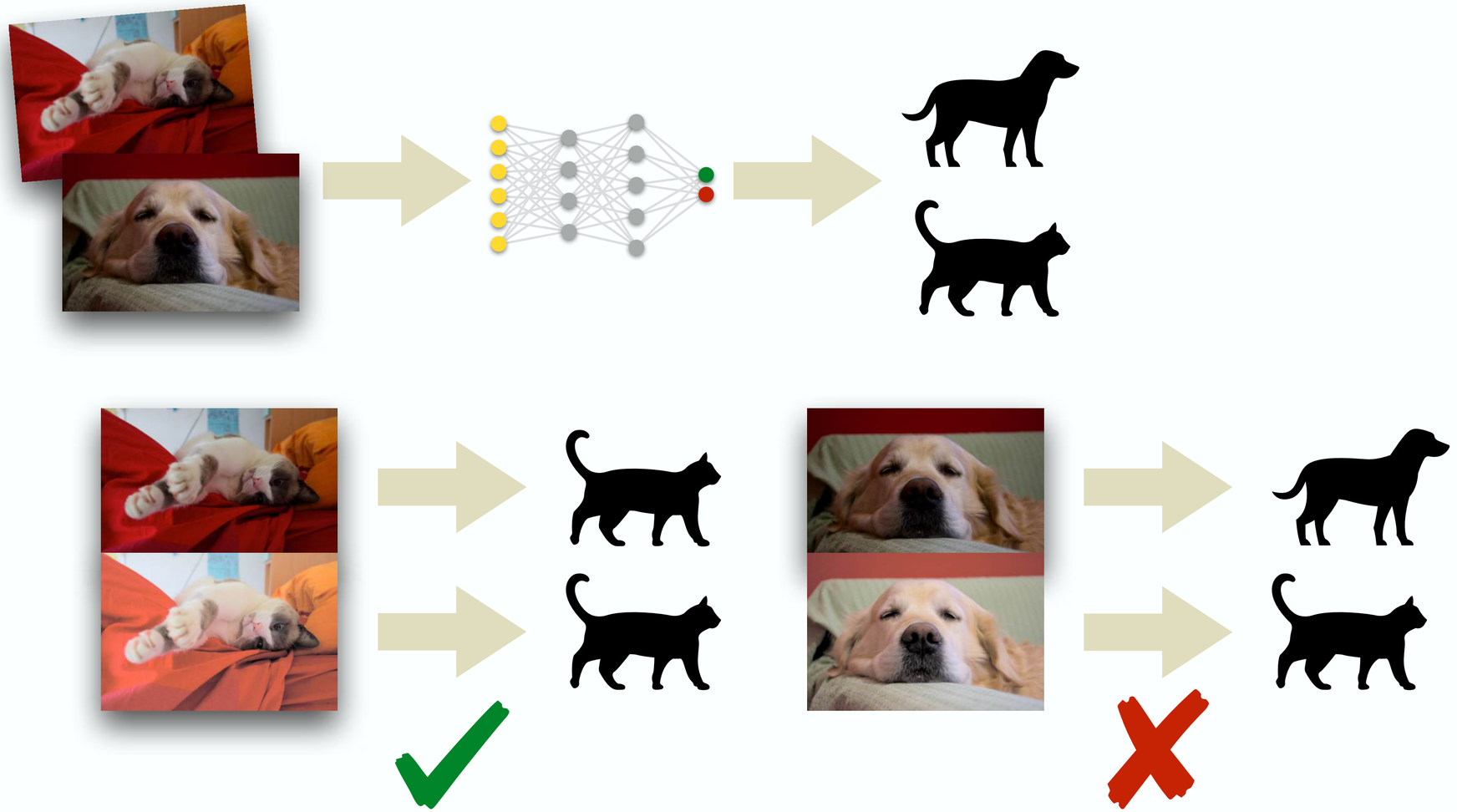
Hypersafety



Prediction Stability

Local Prediction Stability

Prediction is Unaffected by Input Perturbations



Local Prediction Stability

Distance-Based Perturbations

$$P_{\delta,\epsilon}(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{x}' \in \mathscr{R}^{|L_0|} \mid \delta(\mathbf{x}, \mathbf{x}') \leq \epsilon\}$$

Example (L_∞ distance): $P_{\infty,\epsilon}(\mathbf{x}) \stackrel{\text{def}}{=} \{\mathbf{x}' \in \mathscr{R}^{|L_0|} \mid \max_i |\mathbf{x}_i - \mathbf{x}'_i| \leq \epsilon\}$

$$\mathscr{R}^{\delta,\epsilon}_{\mathbf{x}} \stackrel{\text{def}}{=} \{[\![M]\!] \mid \text{STABLE}^{\delta,\epsilon}_{\mathbf{x}}([\![M]\!])\}$$

$\mathscr{R}^{\delta,\epsilon}_{\mathbf{x}}$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that are **stable** in the neighborhood $P_{\delta,\epsilon}(\mathbf{x})$ of a given input $\mathbf{x}$

$$\text{STABLE}^{\delta,\epsilon}_{\mathbf{x}}(T) \stackrel{\text{def}}{=} \forall t \in T\colon\ t_0 \in P_{\delta,\epsilon}(\mathbf{x}) \Rightarrow t_\omega = T(\mathbf{x})$$

classification of $\mathbf{x}$ in T

Theorem

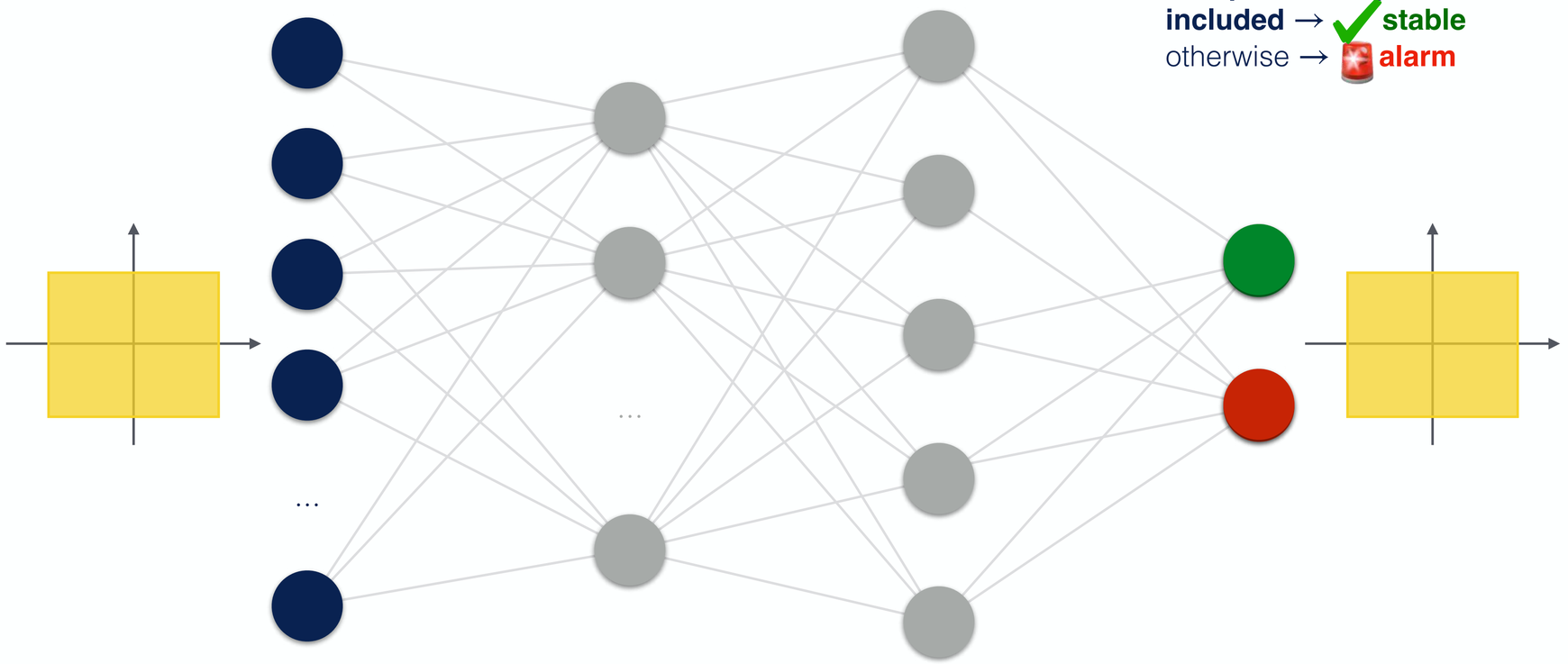
$$M \vDash \mathscr{R}^{\delta,\epsilon}_{\mathbf{x}} \Leftrightarrow \{[\![M]\!]\} \subseteq \mathscr{R}^{\delta,\epsilon}_{\mathbf{x}}$$

Corollary

$$M \vDash \mathscr{R}^{\delta,\epsilon}_{\mathbf{x}} \Leftrightarrow [\![M]\!] \subseteq \bigcup \mathscr{R}^{\delta,\epsilon}_{\mathbf{x}}$$

Forward Analysis

② check output for **inclusion** in **expected output**:
included → ✓ **stable**
otherwise → 🚨 **alarm**

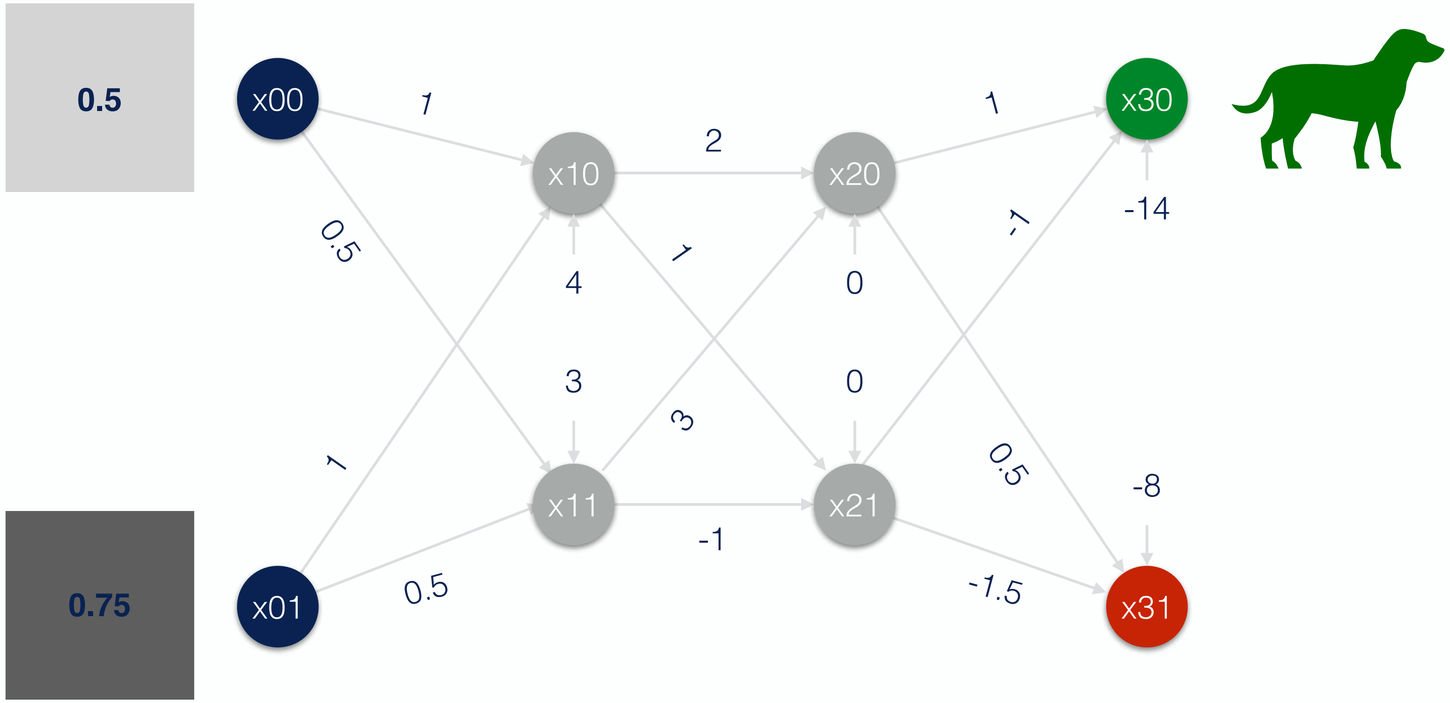


① proceed **forwards** **from an abstraction** of all possible perturbations

Theorem

$$[\![M]\!] \subseteq [\![M]\!]^{\natural} \subseteq \bigcup \mathscr{R}_{\mathbf{x}}^{\delta,\epsilon} \Rightarrow M \vDash \mathscr{R}_{\mathbf{x}}^{\delta,\epsilon}$$

Example

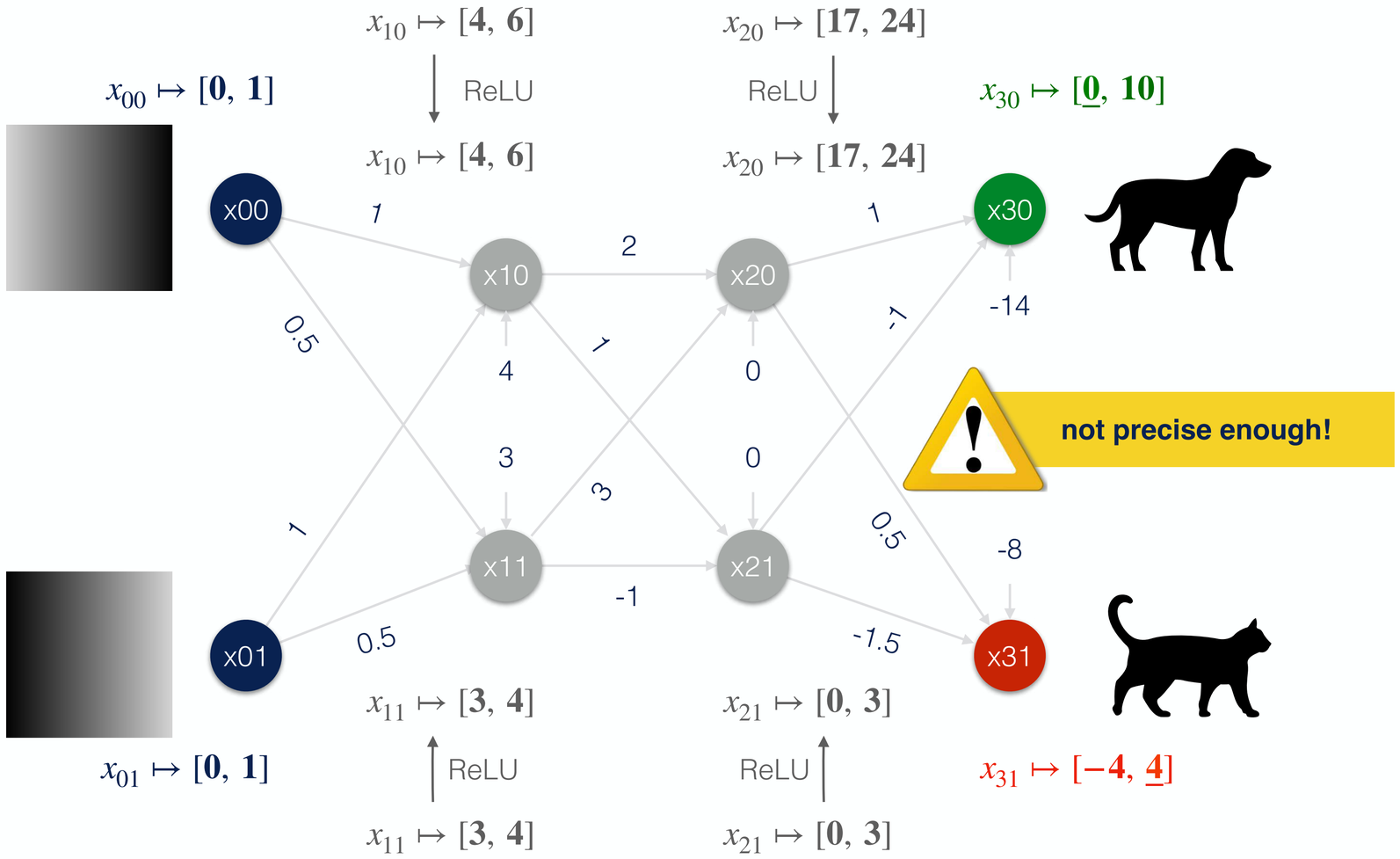


$$P(\langle 0.5,0.75 \rangle) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathscr{R} \times \mathscr{R} \mid 0 \leq \mathbf{x}_0 \leq 1 \wedge 0 \leq \mathbf{x}_1 \leq 1\}$$

Abstraction #1: Intervals

Interval Abstraction

$$x_{i,j} \mapsto [a, b]$$
$$a, b \in \mathcal{R}$$



Abstraction #2: Symbolic

Symbolic Abstraction [Li19]

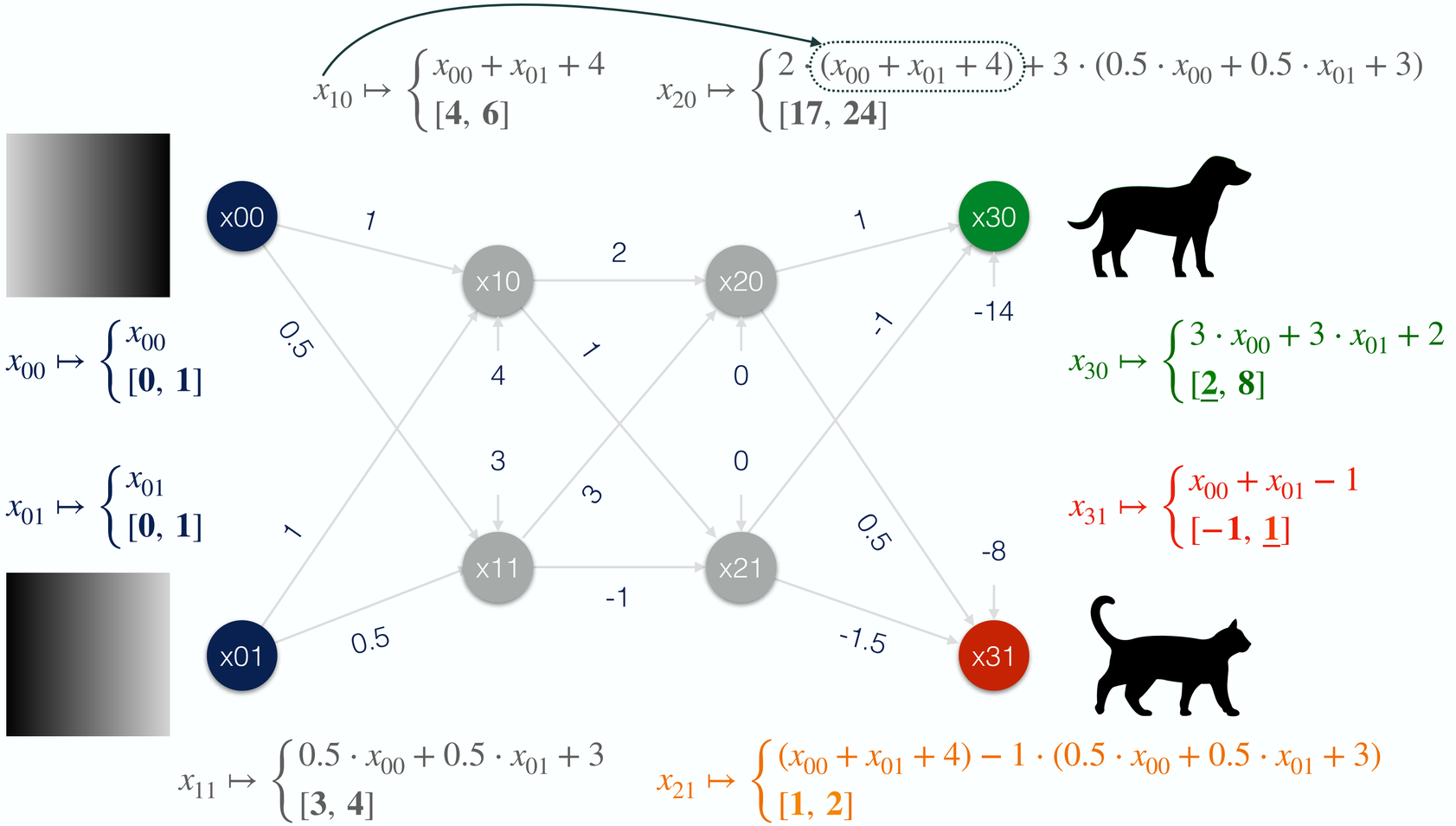
represent each neuron as a linear combination of the inputs and the previous ReLUs

$$x_{i,j} \mapsto \begin{cases} \sum_{k=0}^{i-1} \mathbf{c}_k \cdot \mathbf{x}_k + \mathbf{c} & \mathbf{c}_k, \mathbf{c} \in \mathscr{R}^{|\mathbf{X}_k|} \\ [a,\ b] & a, b \in \mathscr{R} \end{cases}$$

$$\begin{array}{l} x_{i-1,0} \mapsto \mathbf{E_{i-1,0}} \\ \cdots \\ x_{i-1,j} \mapsto \mathbf{E_{i-1,j}} \\ \cdots \end{array} \xrightarrow{x_{i,j} = \sum_k w_{j,k}^{i-1} \cdot x_{i-1,k} + b_{i,j}} x_{i,j} \mapsto \sum_k w_{j,k}^{i-1} \cdot \mathbf{E_{i-1,k}} + b_{i,j}$$

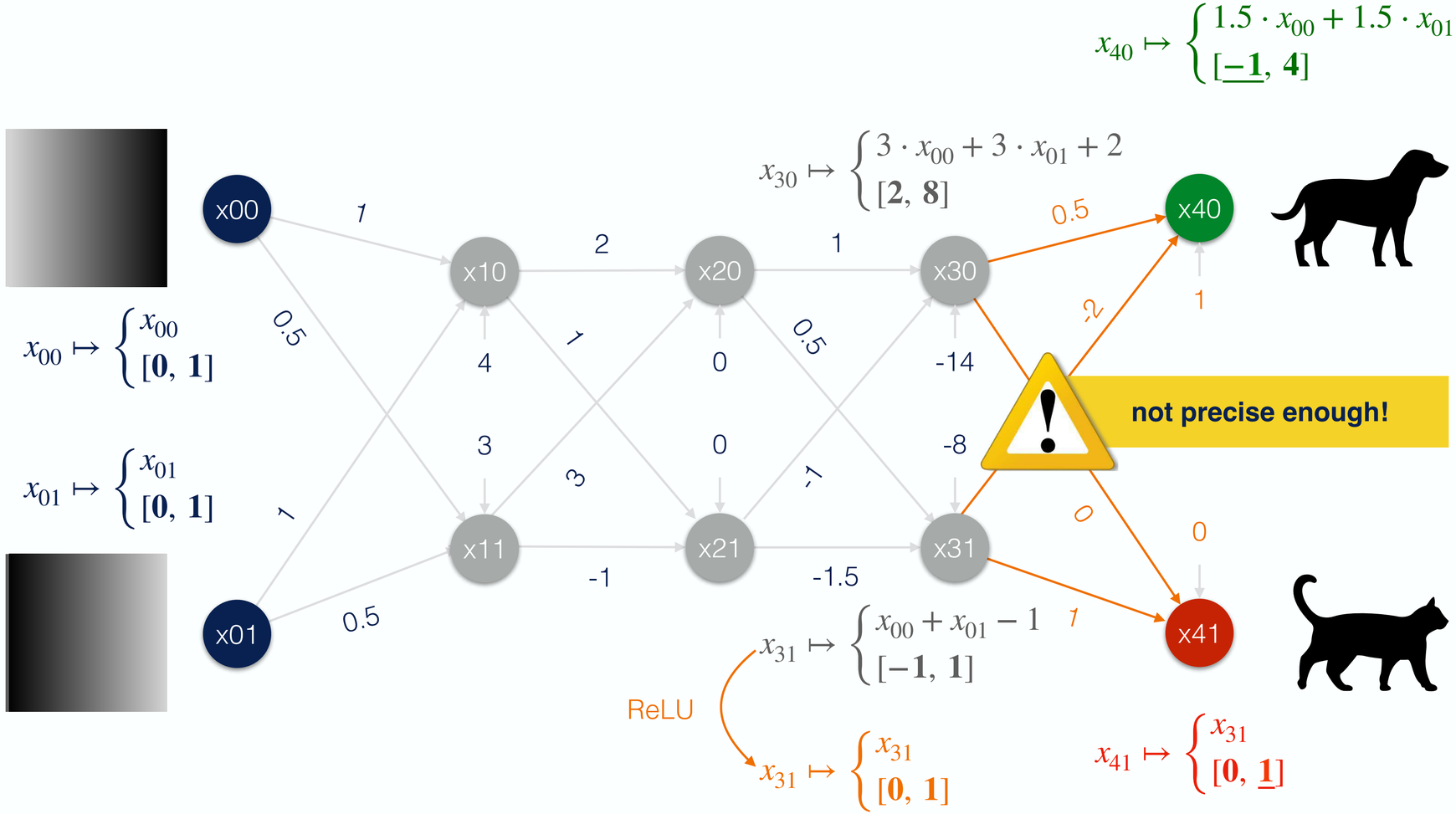
$$x_{i,j} \mapsto \begin{cases} \mathbf{E_{i,j}} \\ \mathbf{[a, b]} \end{cases} \xrightarrow{\text{ReLU}} \begin{array}{ll} x_{i,j} \mapsto \begin{cases} \mathbf{E_{i,j}} \\ \mathbf{[a, b]} \end{cases} & 0 \leq a \\ x_{i,j} \mapsto \begin{cases} \mathbf{x_{i,j}} \\ \mathbf{[0, b]} \end{cases} & a < 0 \wedge 0 < b \\ x_{i,j} \mapsto \begin{cases} \mathbf{0} \\ \mathbf{[0, 0]} \end{cases} & b \leq 0 \end{array}$$

Symbolic Abstraction [Li19]



Symbolic Abstraction [Li19]

Modified Example



Abstraction #3: DeepPoly

DeepPoly Abstraction [Singh19]

maintain symbolic lower- and upper-bounds for each neuron + convex ReLU approximations

$$x_{i+1,j} \mapsto \begin{cases} [\sum_k c_{i,k} \cdot x_{i,k} + c,\ \sum_k d_{i,k} \cdot x_{i,k} + d] & c_{i,k}, c, d_{i,k}, d \in \mathscr{R} \\ [a,\ b] & a, b \in \mathscr{R} \end{cases}$$

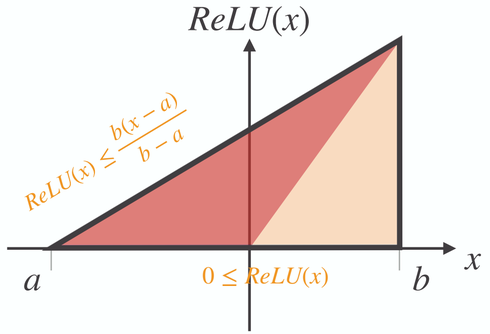
$x_{i,j} \mapsto \begin{cases} [\mathbf{L_{i,j}}, \mathbf{U_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases}$

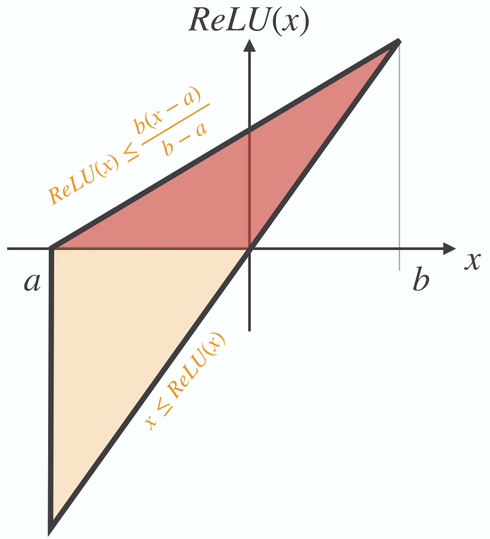
ReLU $0 \leq a$: $x_{i,j} \mapsto \begin{cases} [\mathbf{L_{i,j}}, \mathbf{U_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases}$

ReLU $b \leq -a$: $x_{i,j} \mapsto \begin{cases} \left[\mathbf{0}, \frac{\mathbf{b}(\mathbf{x_{i,j}} - \mathbf{a})}{\mathbf{b} - \mathbf{a}}\right] \\ [\mathbf{0}, \mathbf{b}] \end{cases}$

ReLU $a < 0 \wedge 0 < b$: $x_{i,j} \mapsto \begin{cases} \left[\mathbf{x_{i,j}}, \frac{\mathbf{b}(\mathbf{x_{i,j}} - \mathbf{a})}{\mathbf{b} - \mathbf{a}}\right] \\ [\mathbf{a}, \mathbf{b}] \end{cases}$

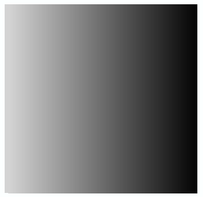
ReLU $b \leq 0$: $x_{i,j} \mapsto \begin{cases} [\mathbf{0}, \mathbf{0}] \\ [0, 0] \end{cases}$





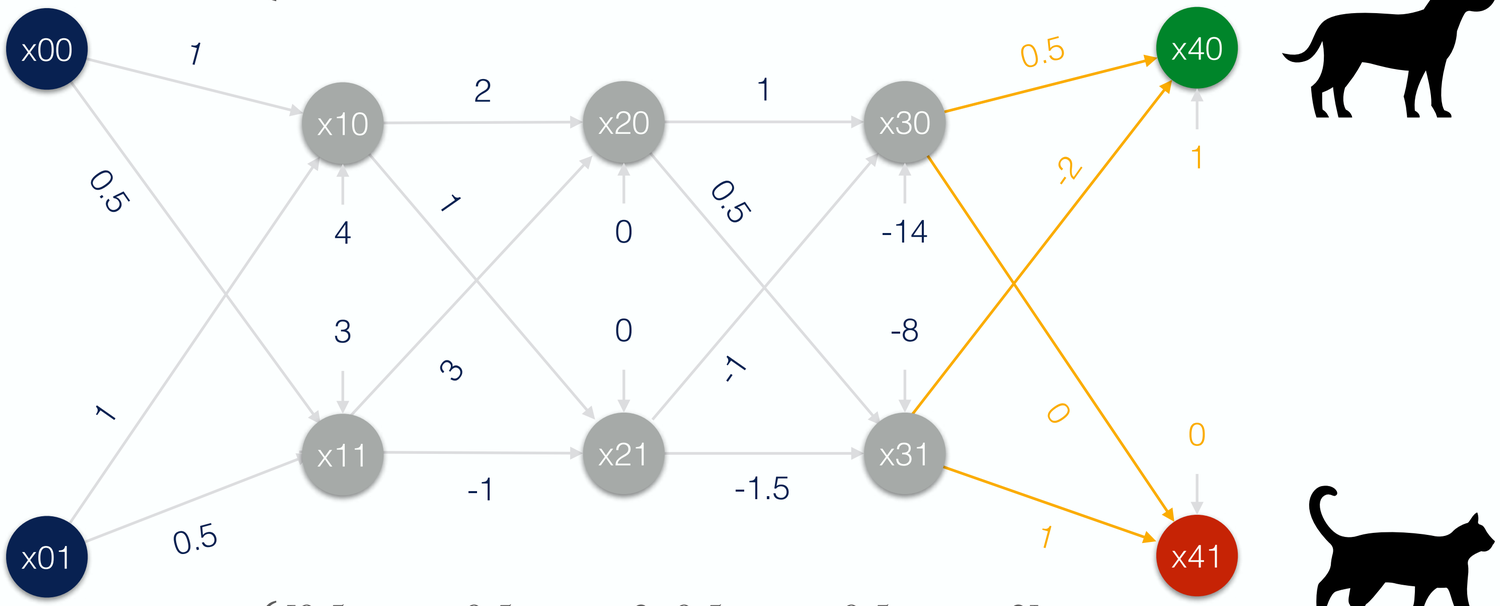
DeepPoly Abstraction [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00}, x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$



$$x_{01} \mapsto \begin{cases} [x_{01}, x_{01}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

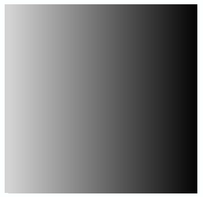
$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4}, \mathbf{6}] \end{cases}$$



$$x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3}, \mathbf{4}] \end{cases}$$

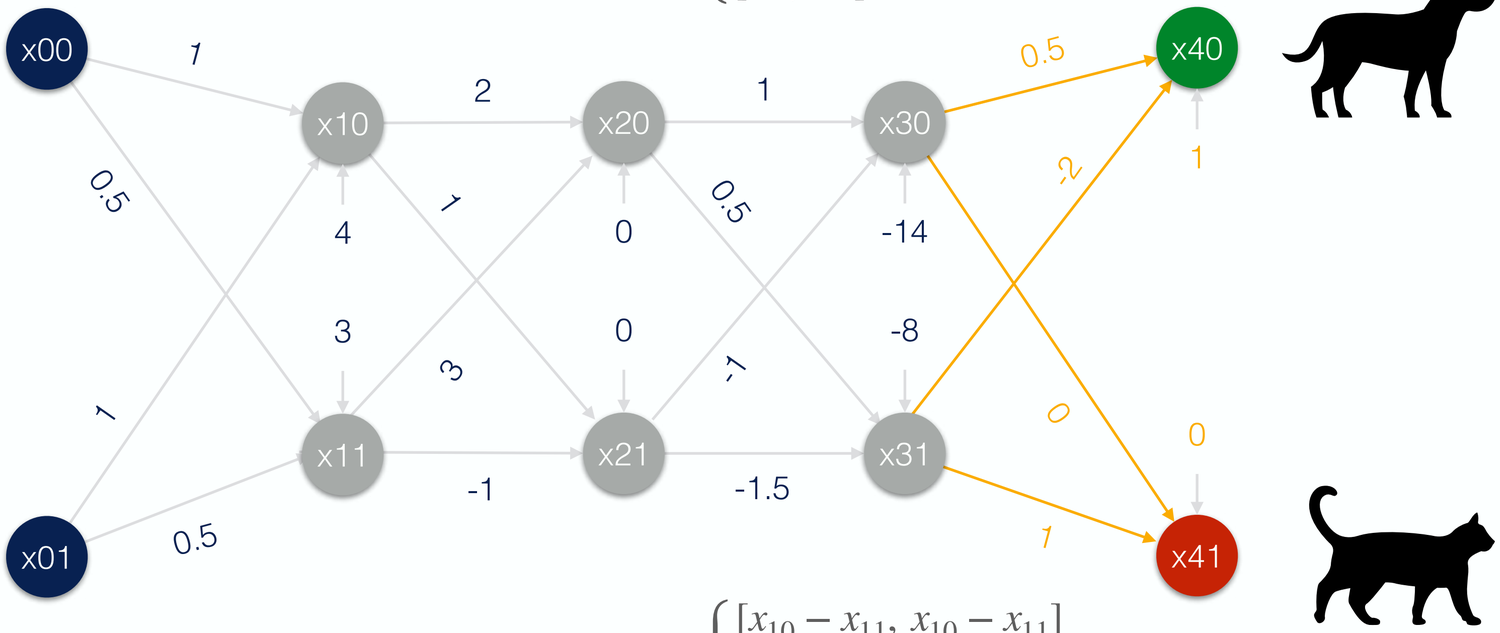
DeepPoly Abstraction [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00}, x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$



$$x_{01} \mapsto \begin{cases} [x_{01}, x_{01}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

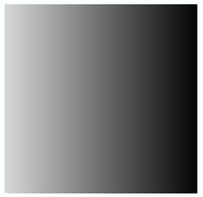
$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17}, \mathbf{24}] \end{cases}$$



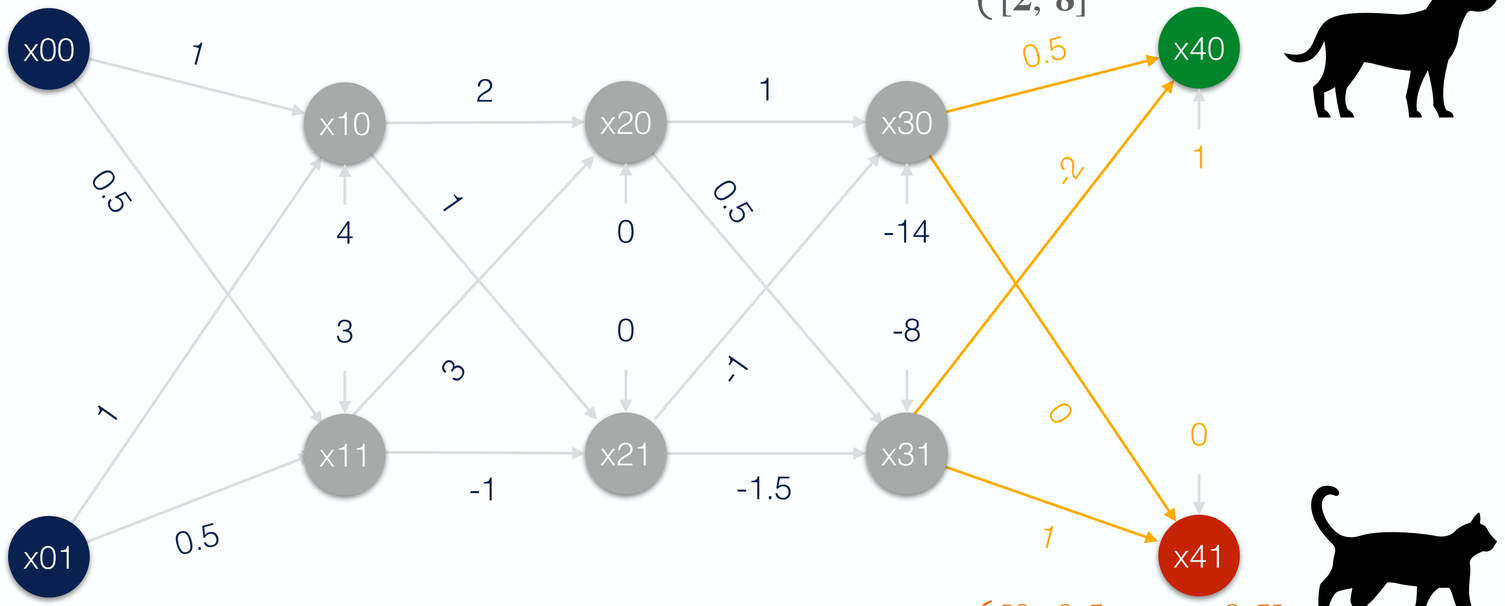
$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1}, \mathbf{2}] \end{cases}$$

DeepPoly Abstraction [Singh19]

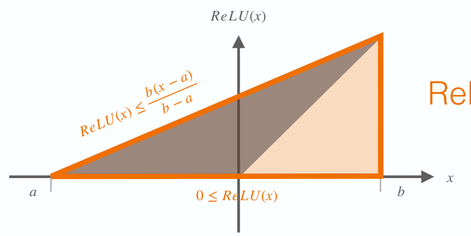
$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$



$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$



$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2},\ \mathbf{8}] \end{cases}$$



ReLU

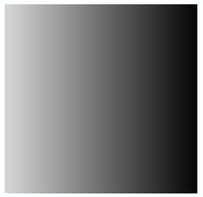
$$x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot x_{31} + 0.5] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} [0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8,\ 0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8] \\ [-\mathbf{1},\ \mathbf{1}] \end{cases}$$

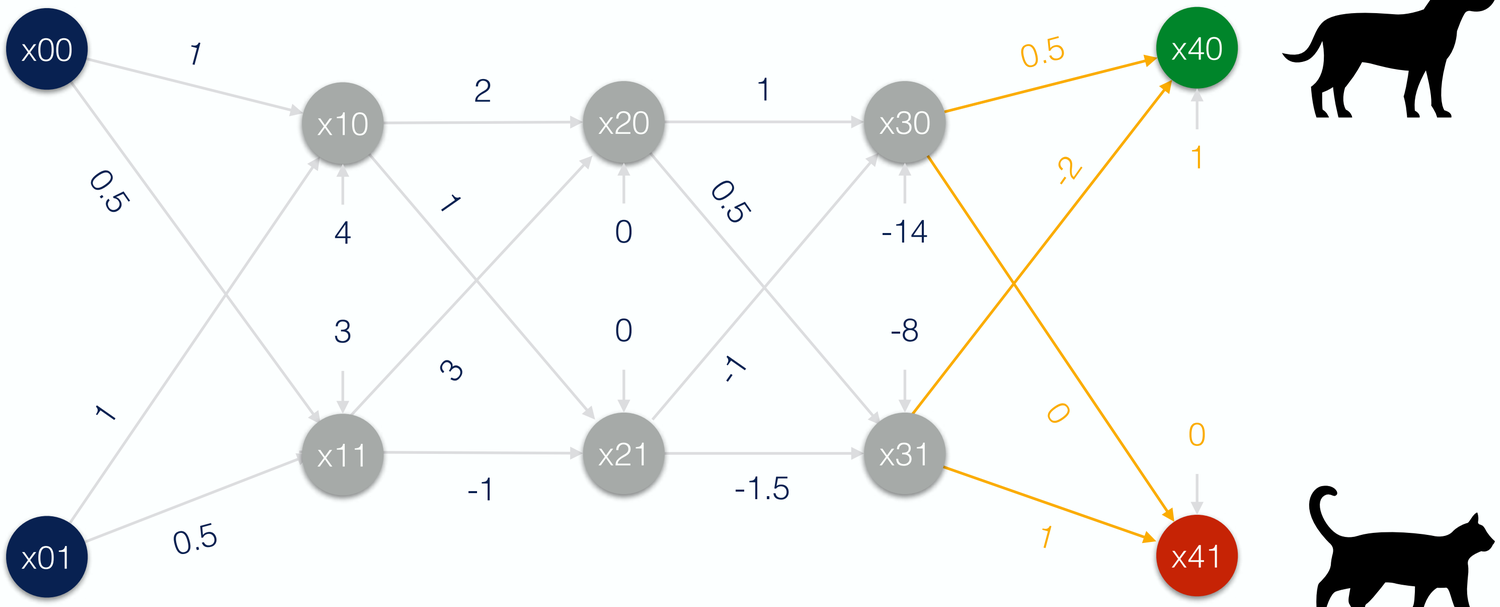
DeepPoly Abstraction [Singh19]

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \end{cases}$$

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$



$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$



DeepPoly Abstraction [Singh19]

Back-Substitution

$x_{00} \mapsto [\mathbf{0}, \mathbf{1}]$

$x_{01} \mapsto [\mathbf{0}, \mathbf{1}]$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4}, \mathbf{6}] \end{cases}$$

$$x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3}, \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17}, \mathbf{24}] \end{cases}$$

$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1}, \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2}, \mathbf{8}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [x_{10} - x_{11} + 1,\ 0.5 \cdot x_{10} + 2 \cdot x_{11} - 6] \\ \end{cases}$$

$$\mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 2,\ 1.5 \cdot x_{00} + 1.5 \cdot x_{11} + 2] \\ [\underline{\mathbf{2}}, \mathbf{5}] \end{cases}$$

DeepPoly Abstraction [Singh19]

Partial Back-Substitution

$x_{00} \mapsto [\mathbf{0}, \mathbf{1}]$

$x_{01} \mapsto [\mathbf{0}, \mathbf{1}]$

$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01} + 4,\ x_{00} + x_{01} + 4] \\ [\mathbf{4}, \mathbf{6}] \end{cases}$$

$$x_{11} \mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3] \\ [\mathbf{3}, \mathbf{4}] \end{cases}$$

$$x_{20} \mapsto \begin{cases} [2 \cdot x_{10} + 3 \cdot x_{11},\ 2 \cdot x_{10} + 3 \cdot x_{11}] \\ [\mathbf{17}, \mathbf{24}] \end{cases}$$

$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \\ [\mathbf{1}, \mathbf{2}] \end{cases}$$

$$x_{30} \mapsto \begin{cases} [x_{20} - x_{21} - 14,\ x_{20} - x_{21} - 14] \\ [\mathbf{2}, \mathbf{8}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} [0,\ 0.5 \cdot (0.5 \cdot x_{20} - 1.5 \cdot x_{21} - 8) + 0.5] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ [\underline{\mathbf{0}}, \mathbf{5}] \end{cases}$$

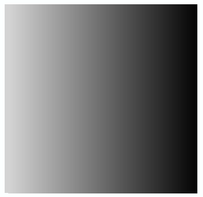
$$\mapsto \begin{cases} [x_{21} + 1,\ 0.5 \cdot x_{20} - 0.5 \cdot x_{21} - 6] \\ [\mathbf{2}, \underline{\mathbf{5.5}}] \end{cases}$$

$$\mapsto \begin{cases} [x_{10} - x_{11} + 1,\ 0.5 \cdot x_{10} + 2 \cdot x_{11} - 6] \\ [\underline{\mathbf{1}}, \mathbf{5}] \end{cases}$$

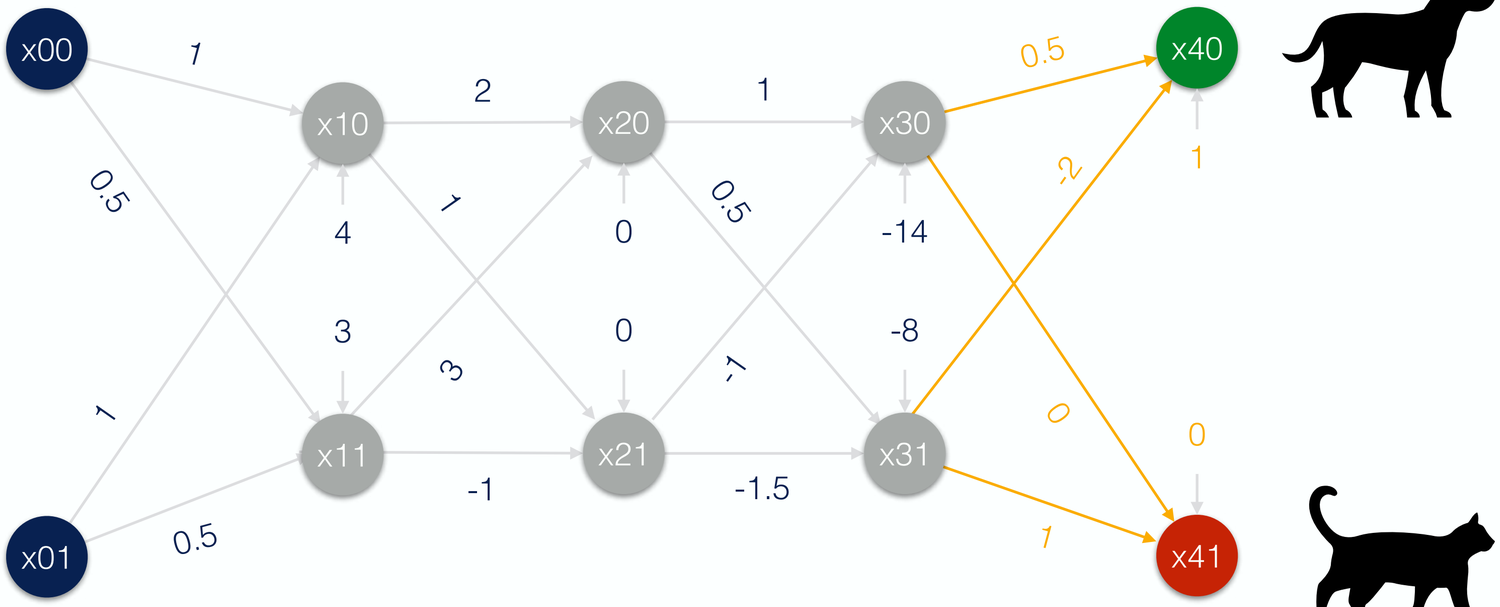
$$\mapsto \begin{cases} [0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 2,\ 1.5 \cdot x_{00} + 1.5 \cdot x_{11} + 2] \\ [\mathbf{2}, \mathbf{5}] \end{cases}$$

DeepPoly Abstraction [Singh19]

$$x_{00} \mapsto \begin{cases} [x_{00}, x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$



$$x_{01} \mapsto \begin{cases} [x_{01}, x_{01}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

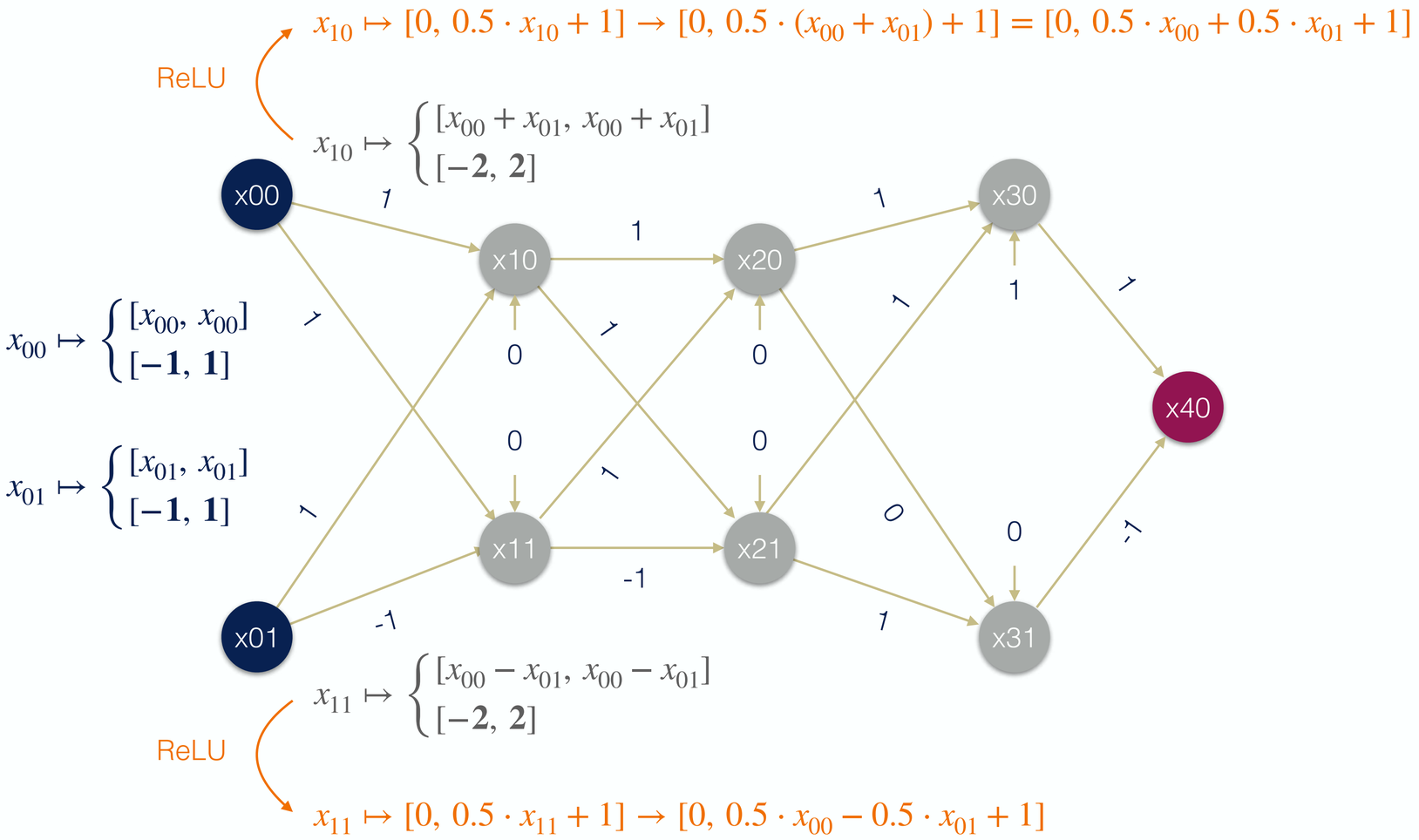


$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1,\ 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ [\mathbf{2}, \mathbf{5}] \end{cases}$$

$$x_{41} \mapsto \begin{cases} [x_{31}, x_{31}] \\ [\mathbf{0}, \underline{\mathbf{1}}] \end{cases}$$

DeepPoly Abstraction [Singh19]

Maintaining Symbolic Bounds with Respect to the Inputs ("à la Symbolic")

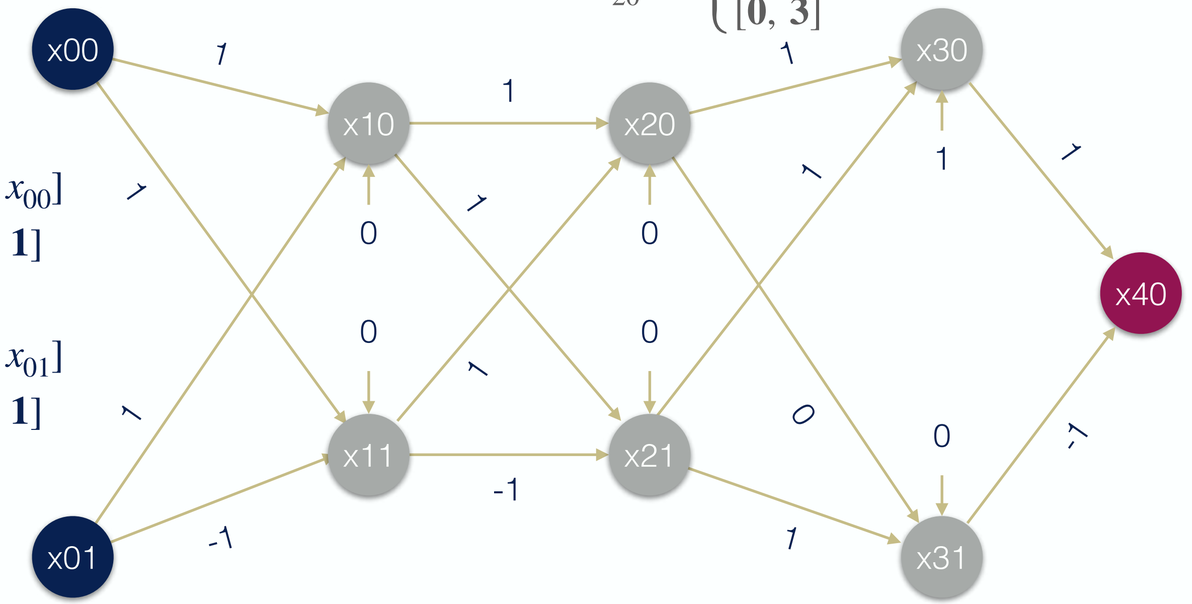


DeepPoly Abstraction [Singh19]

Maintaining Symbolic Bounds with Respect to the Inputs ("à la Symbolic")

$$x_{10} \mapsto [0,\ 0.5 \cdot x_{10} + 1] \rightarrow [0,\ 0.5 \cdot (x_{00} + x_{01}) + 1] = [0,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 1]$$

$$x_{20} \mapsto \begin{cases} [x_{10} + x_{11},\ x_{10} + x_{11}] \rightarrow [0,\ x_{00} + 2] \\ [\mathbf{0},\ \mathbf{3}] \end{cases}$$



$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11},\ x_{10} - x_{11}] \rightarrow [-0.5 \cdot x_{00} + 0.5 \cdot x_{01} - 1,\ 0.5 \cdot x_{00} - 0.5 \cdot x_{01} + 1] \\ [\mathbf{-2},\ \mathbf{2}] \end{cases}$$

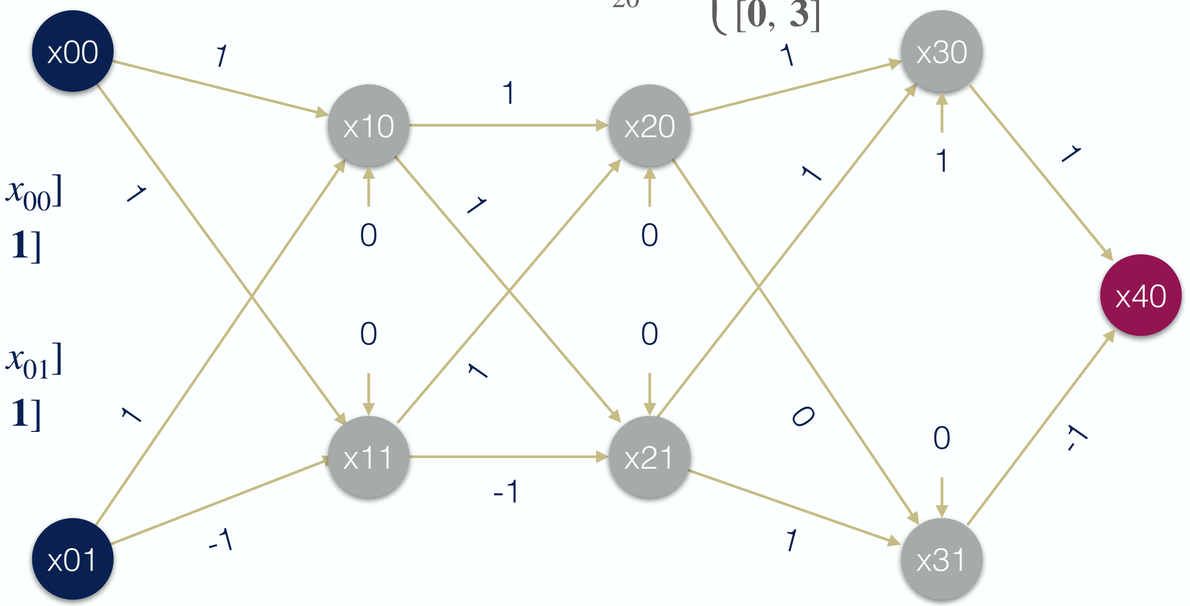
$$x_{11} \mapsto [0,\ 0.5 \cdot x_{11} + 1] \rightarrow [0,\ 0.5 \cdot x_{00} - 0.5 \cdot x_{01} + 1]$$

DeepPoly Abstraction [Singh19]

Maintaining Symbolic Bounds with Respect to the Inputs ("à la Symbolic")

$$x_{00} \mapsto \begin{cases} [x_{00}, x_{00}] \\ [\mathbf{-1}, \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01}, x_{01}] \\ [\mathbf{-1}, \mathbf{1}] \end{cases}$$



$$x_{20} \mapsto \begin{cases} [x_{10} + x_{11}, x_{10} + x_{11}] \rightarrow [0, x_{00} + 2] \\ [\mathbf{0}, \mathbf{3}] \end{cases}$$

$$x_{21} \mapsto \begin{cases} [x_{10} - x_{11}, x_{10} - x_{11}] \rightarrow [-0.5 \cdot x_{00} + 0.5 \cdot x_{01} - 1, 0.5 \cdot x_{00} - 0.5 \cdot x_{01} + 1] \\ [\mathbf{-2}, \mathbf{2}] \end{cases}$$

ReLU

$$x_{21} \mapsto [0, 0.5 \cdot x_{21} + 1] \rightarrow [0, 0.25 \cdot x_{00} - 0.25 \cdot x_{01} + 1.5]$$

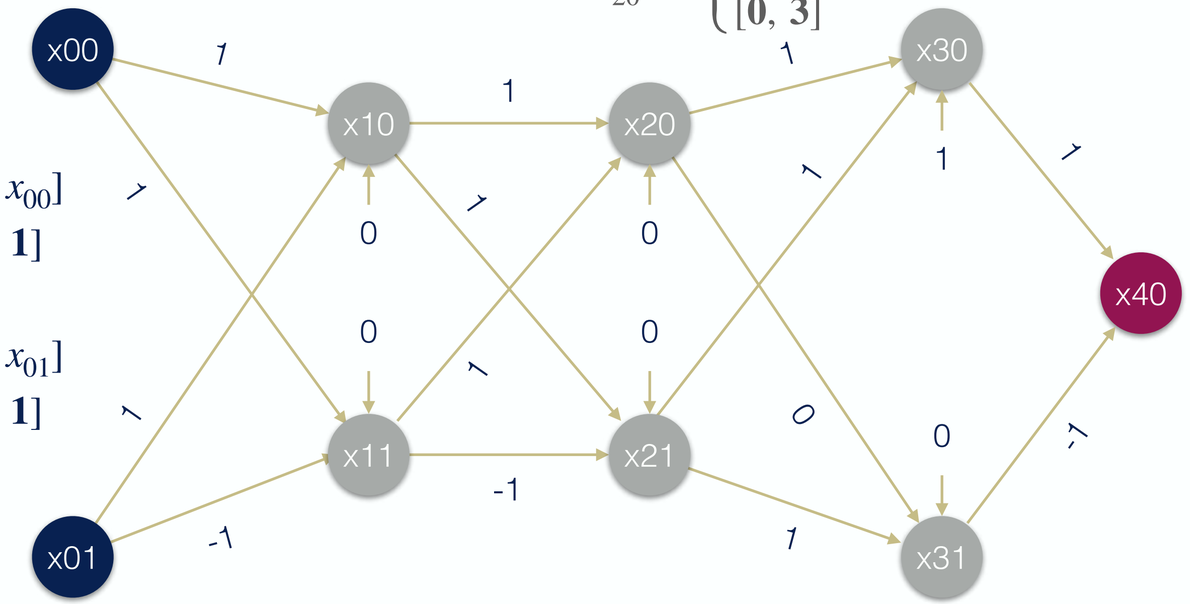
DeepPoly Abstraction [Singh19]

Maintaining Symbolic Bounds with Respect to the Inputs ("à la Symbolic")

$$x_{30} \mapsto \begin{cases} [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] \rightarrow [1,\ 1.25 \cdot x_{00} - 0.25 \cdot x_{01} + 4.5] \\ [\mathbf{1},\ \underline{\mathbf{6}}] \end{cases}$$

← $[\mathbf{1}, \mathbf{5}.\mathbf{5}]$ with back-substitution

$$x_{20} \mapsto \begin{cases} [x_{10} + x_{11},\ x_{10} + x_{11}] \rightarrow [0,\ x_{00} + 2] \\ [\mathbf{0},\ \mathbf{3}] \end{cases}$$



$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [\mathbf{-1},\ \mathbf{1}] \end{cases}$$

$$x_{31} \mapsto \begin{cases} [x_{21},\ x_{21}] \rightarrow [0,\ 0.25 \cdot x_{00} - 0.25 \cdot x_{01} + 1.5] \\ [\mathbf{0},\ \mathbf{2}] \end{cases}$$

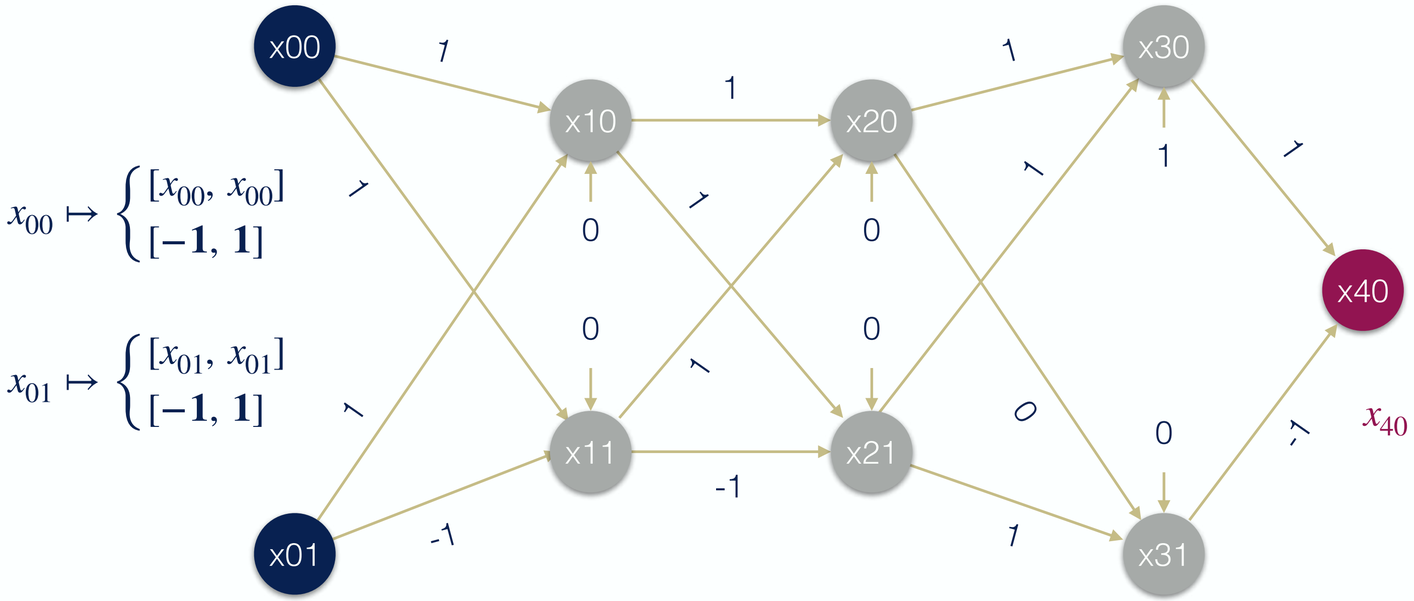
$$x_{21} \mapsto [0,\ 0.5 \cdot x_{21} + 1] \rightarrow [0,\ 0.25 \cdot x_{00} - 0.25 \cdot x_{01} + 1.5]$$

DeepPoly Abstraction [Singh19]

Maintaining Symbolic Bounds with Respect to the Inputs ("à la Symbolic")

$$x_{30} \mapsto \begin{cases} [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] \rightarrow [1,\ 1.25 \cdot x_{00} - 0.25 \cdot x_{01} + 4.5] \\ [\mathbf{1},\ \underline{\mathbf{6}}] \end{cases}$$

← $[\mathbf{1}, \mathbf{5}.\mathbf{5}]$ with back-substitution



$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [-\mathbf{1},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [-\mathbf{1},\ \mathbf{1}] \end{cases}$$

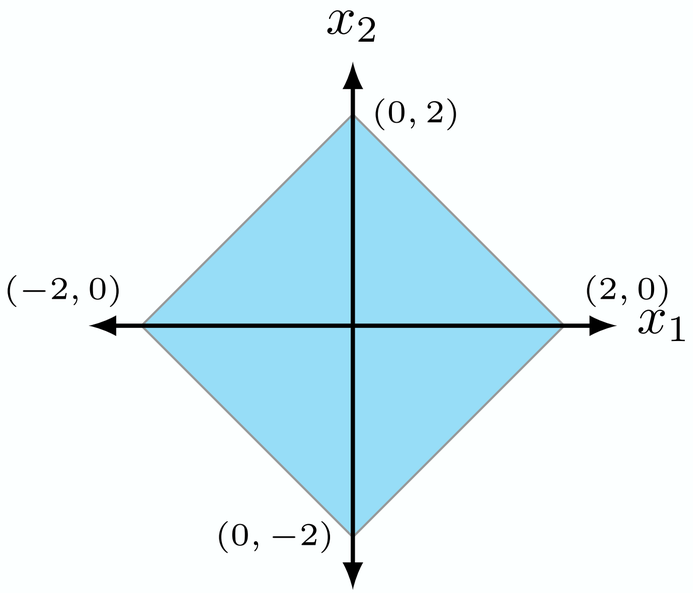
$$x_{40} \mapsto \begin{cases} \cdots \\ [-\mathbf{1},\ \underline{\mathbf{6}}] \end{cases}$$

← $[\mathbf{1}, \mathbf{4}]$ with back-substitution

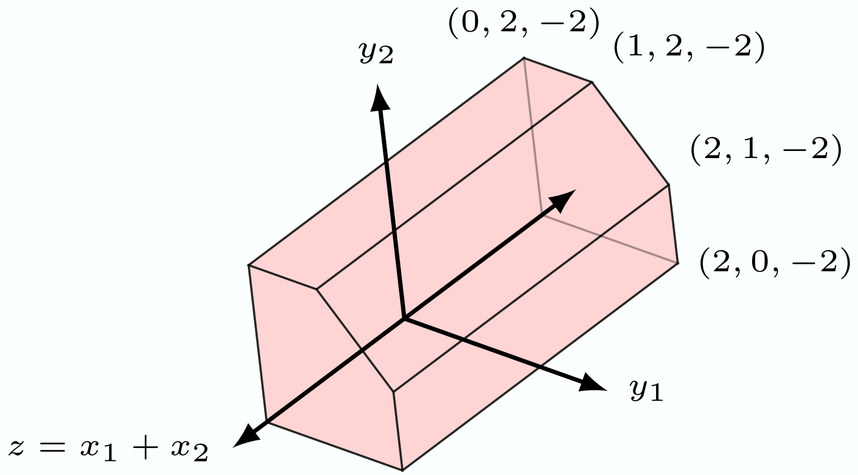
$$x_{31} \mapsto \begin{cases} [x_{21},\ x_{21}] \rightarrow [0,\ 0.25 \cdot x_{00} - 0.25 \cdot x_{01} + 1.5] \\ [\mathbf{0},\ \mathbf{2}] \end{cases}$$

Going Farther: Multi-Neuron Abstractions

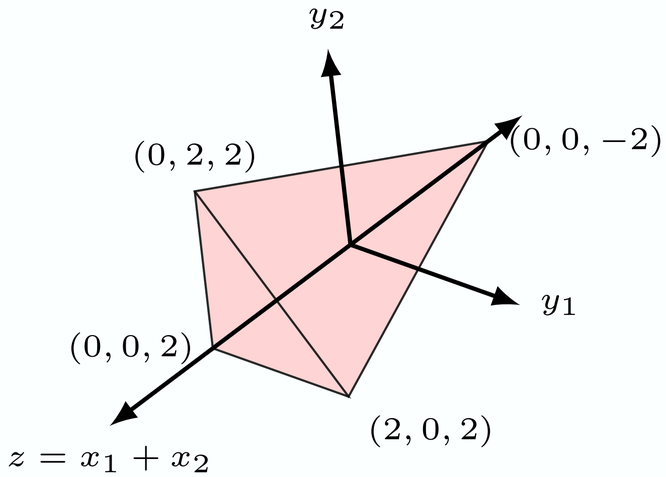
Multi-Neuron Abstractions



(a) Input shape



(b) 1-ReLU



(c) 2-ReLU

Singh, Ganvir, Püschel and Vechev. Beyond the Single Neuron Convex Barrier for Neural Network Certification. In NeurIPS, 2019

Other Static Analysis Methods

- **T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev**. *AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation*. In S&P, 2018.
  the first use of abstract interpretation for verifying neural network stability

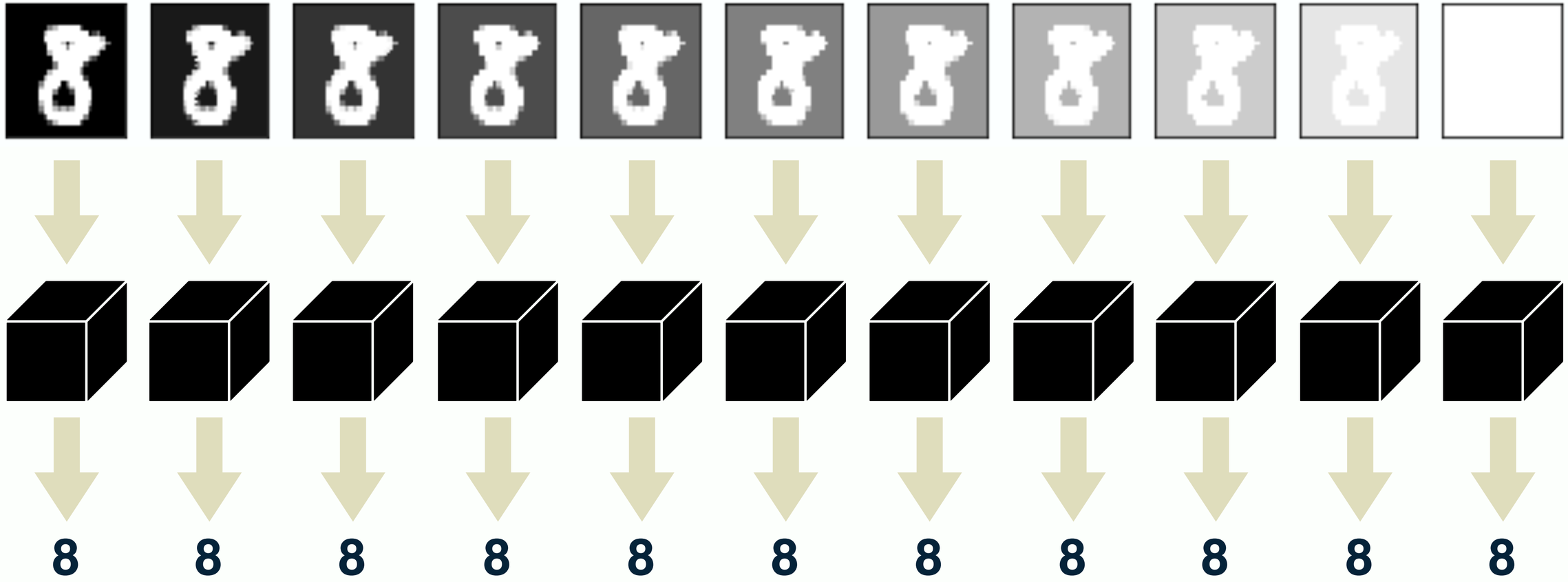
- **G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev**. *Fast and Effective Robustness Certification*. In NeurIPS, 2018.
  a custom zonotope domain for certifying neural network stability

- **M. N. Müller, G. Makarchuk, G. Singh, M. Püschel, and M. Vechev**. *PRIMA: General and Precise Neural Network Certification via Scalable Convex Hull Approximations*. In POPL, 2022.
  a multi-neuron abstraction via a convex-hull approximation algorithm

Saliency Map Stability

Local Prediction Stability

Not Enough!

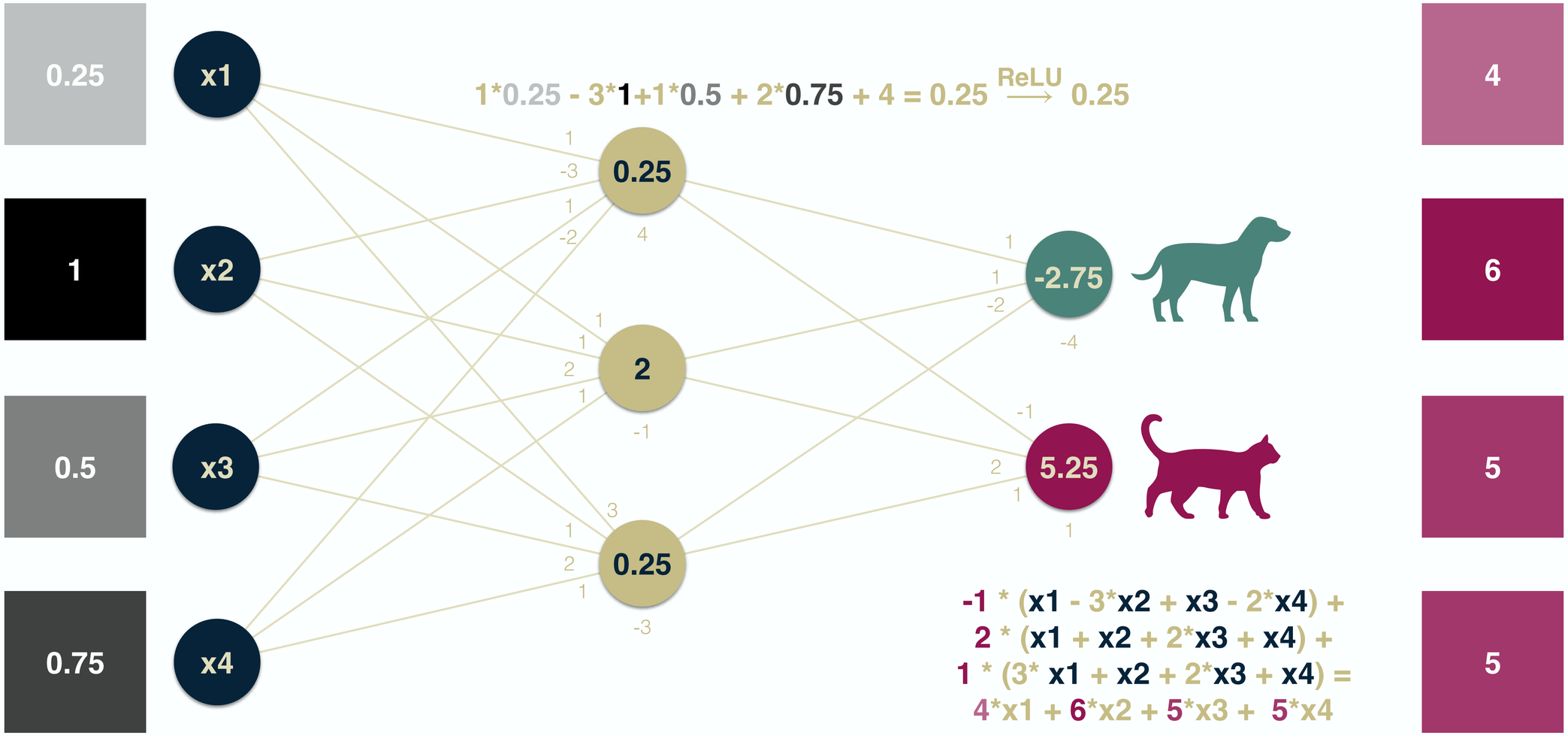


Local Stability Verification [Munakata23]

Saliency Map Stability

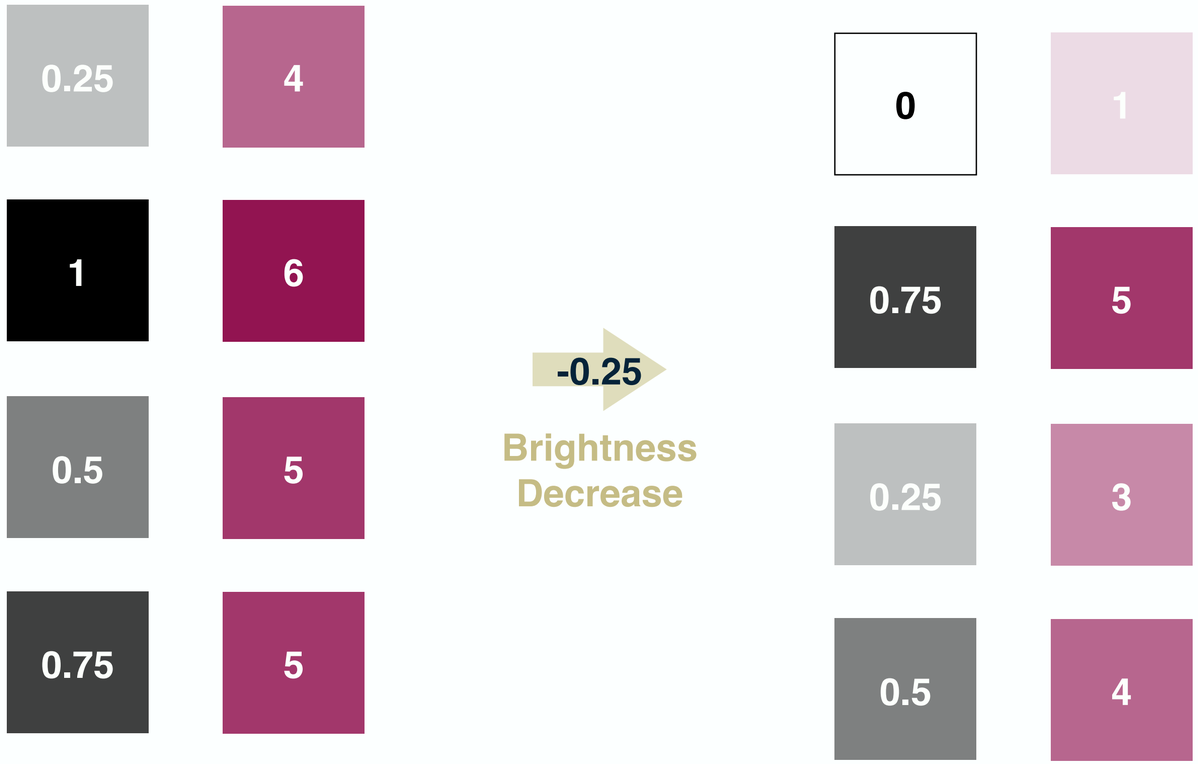
(A Very Small) Example

Saliency Maps



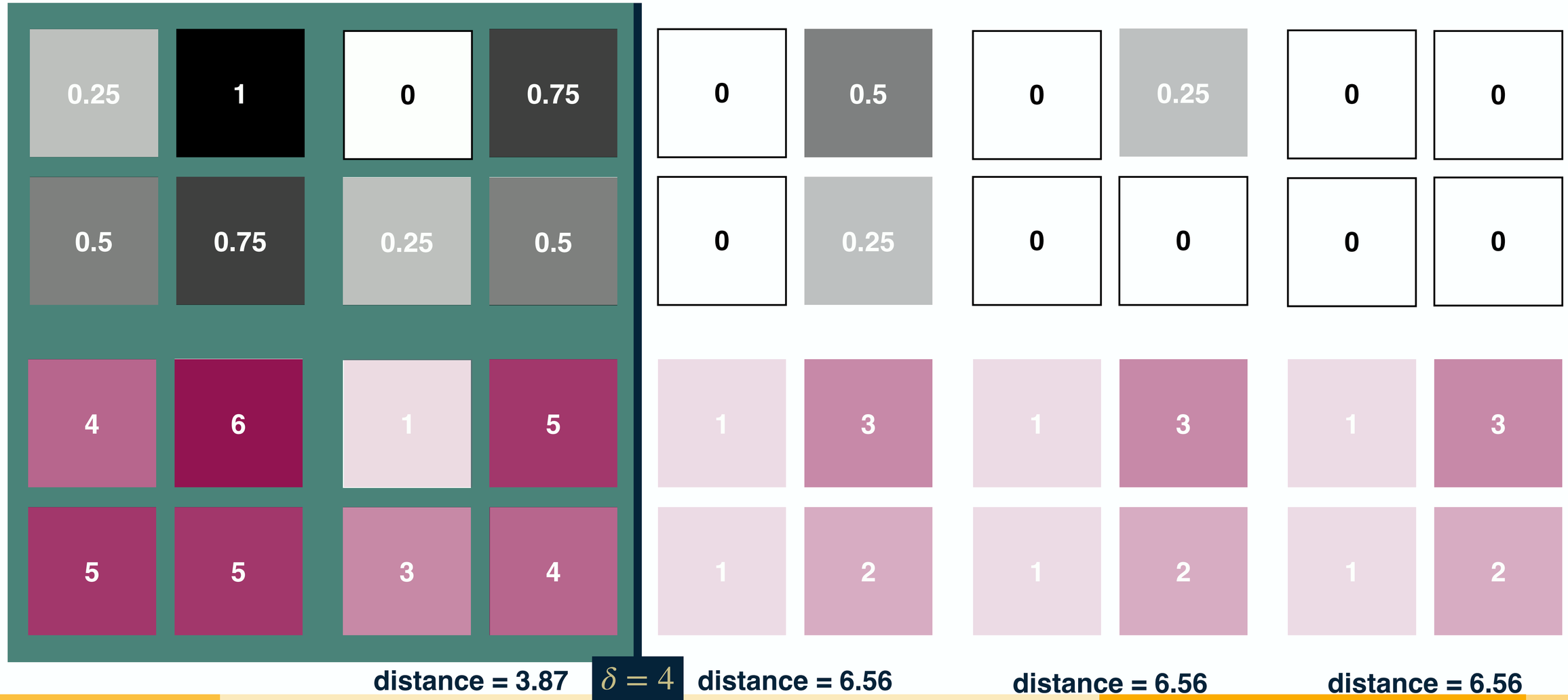
(A Very Small) Example

Semantic Perturbations



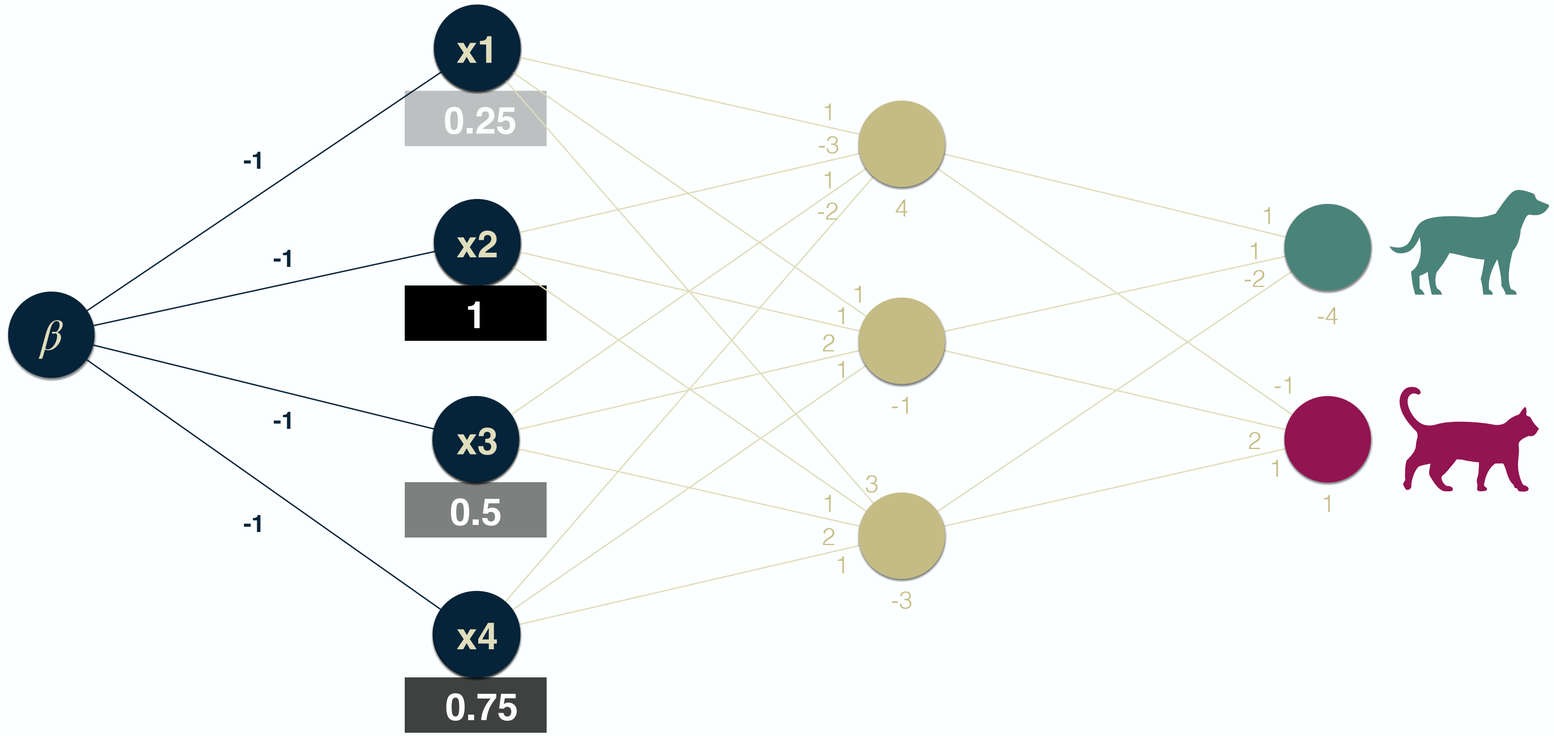
(A Very Small) Example

Saliency Map Stability



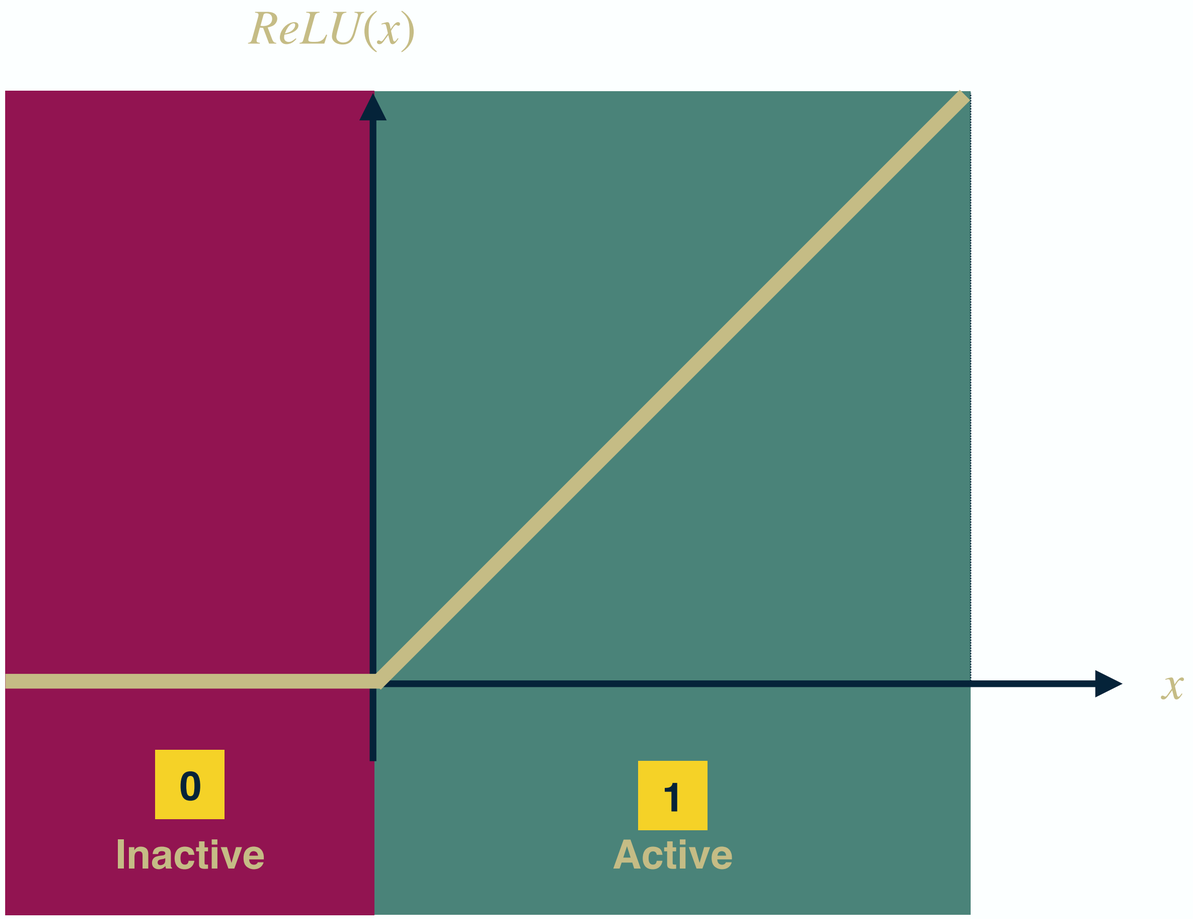
(A Very Small) Example

Encoding Semantic Perturbations [Mohapatra20]



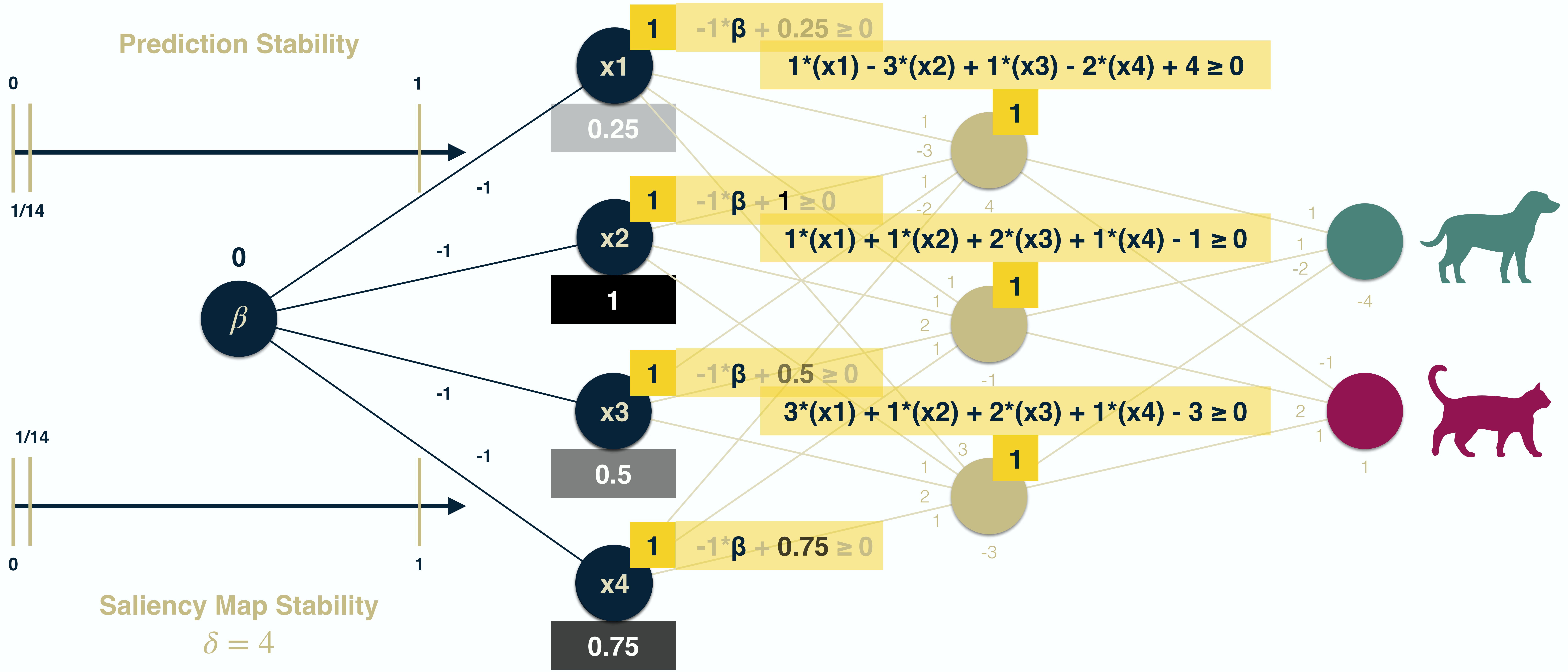
Naïve Breadth-First Search

Activation Patterns



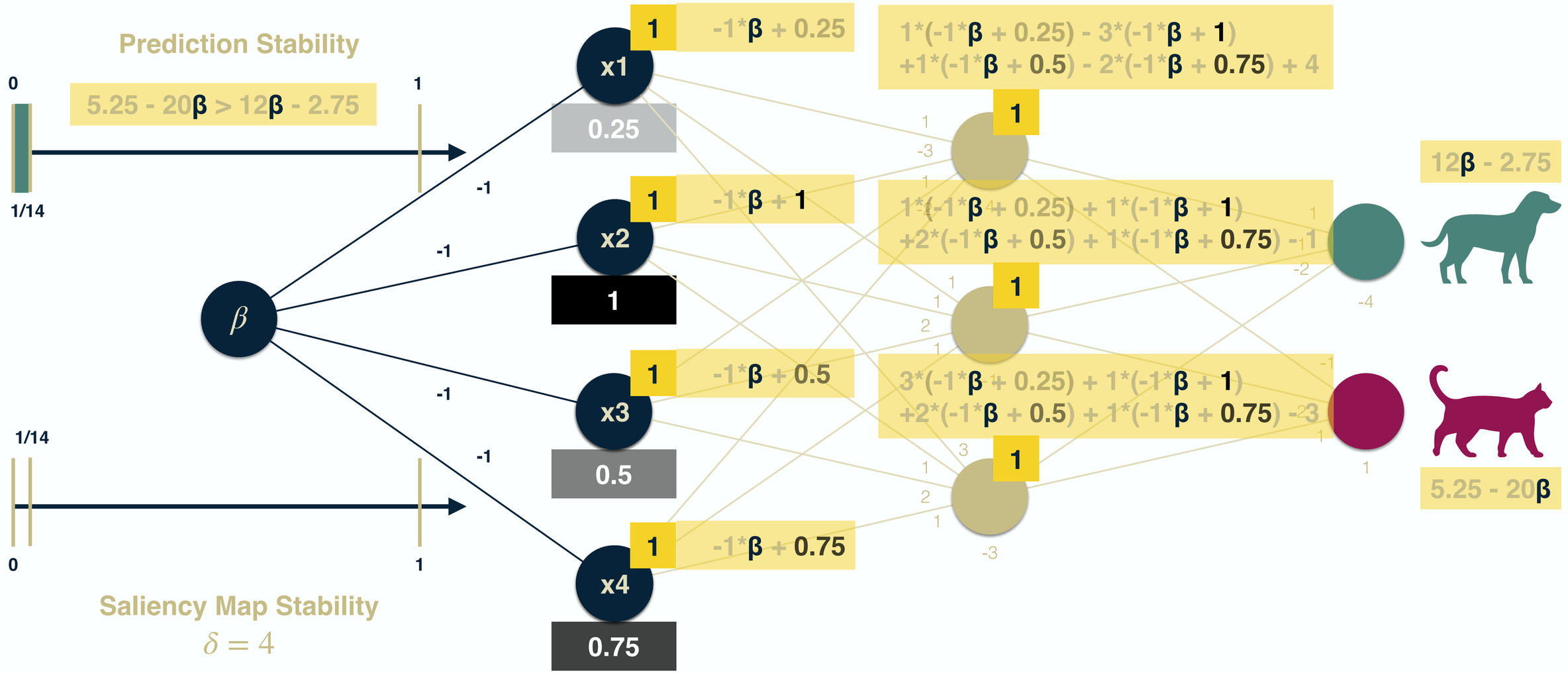
(A Very Small) Example

Activation Patterns



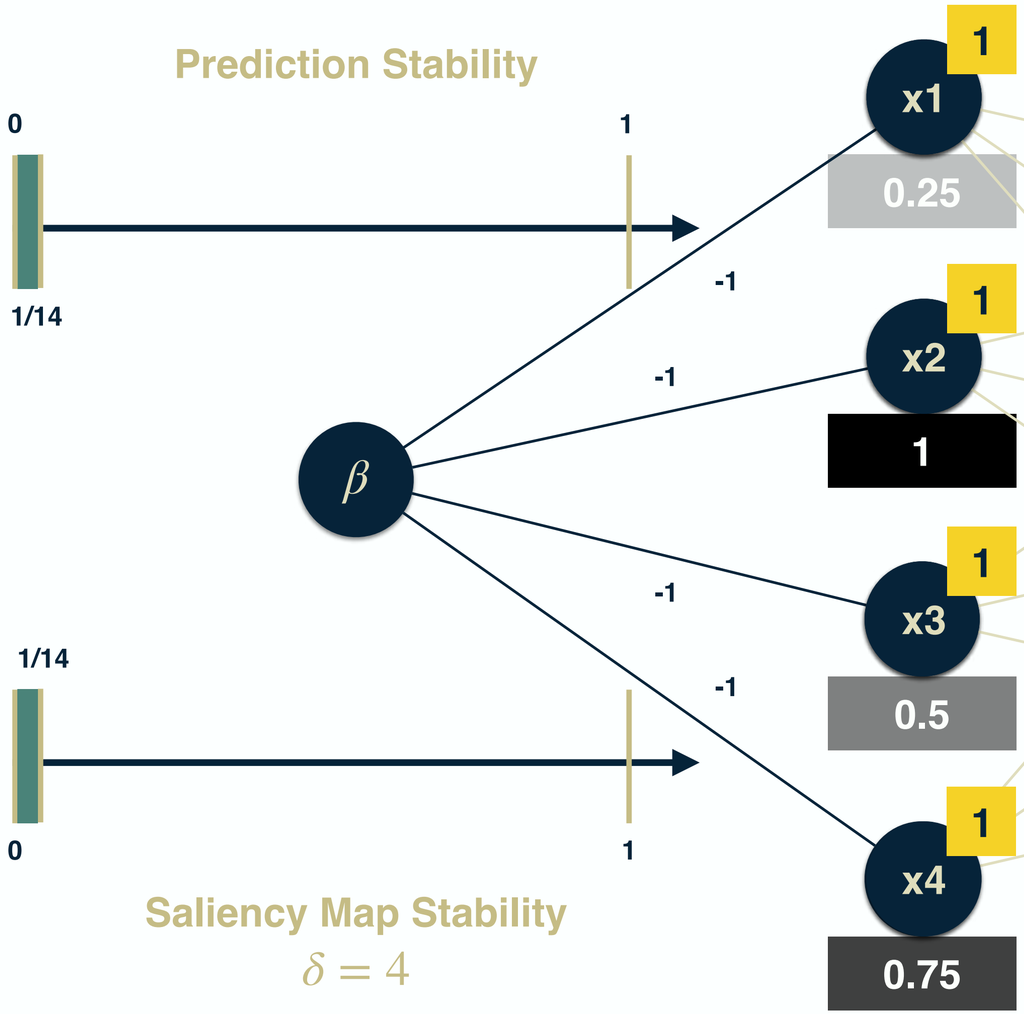
(A Very Small) Example

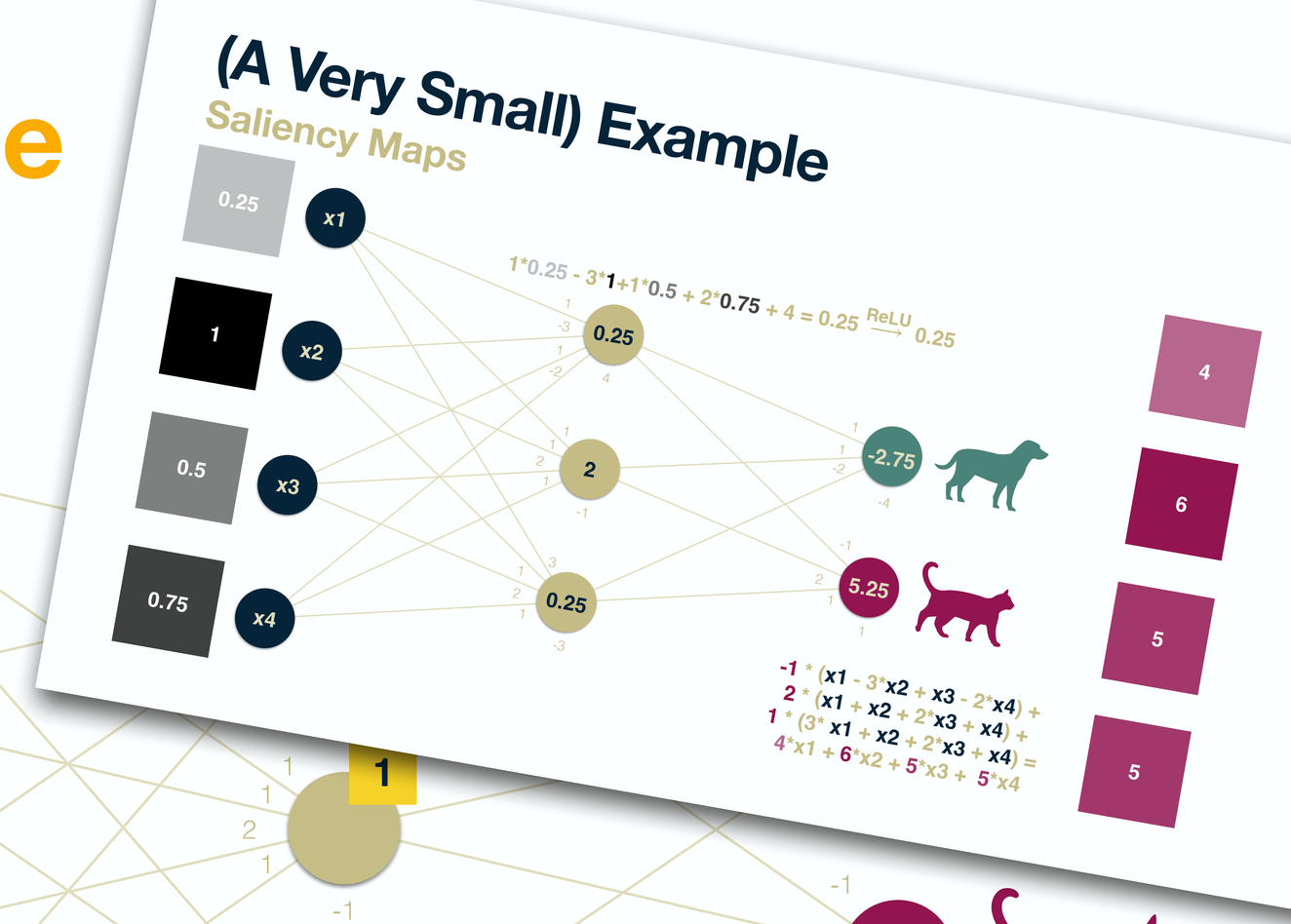
Prediction Stability

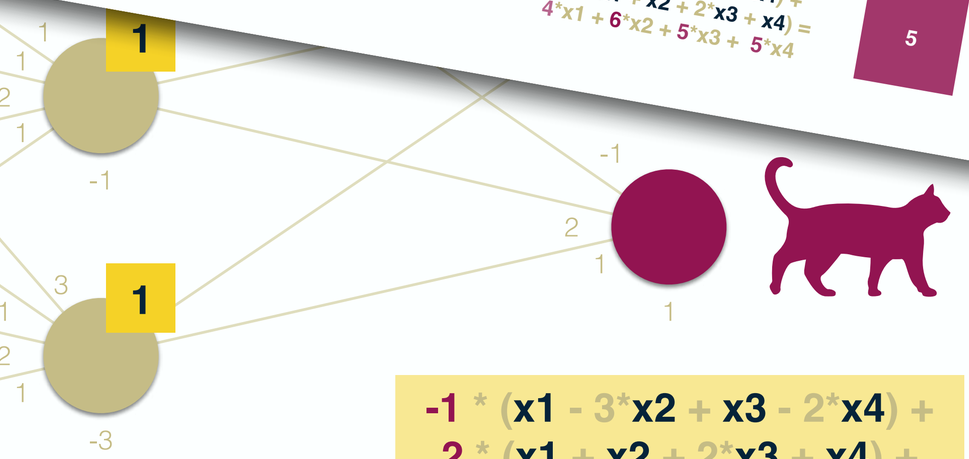


(A Very Small) Example

Saliency Map Stability

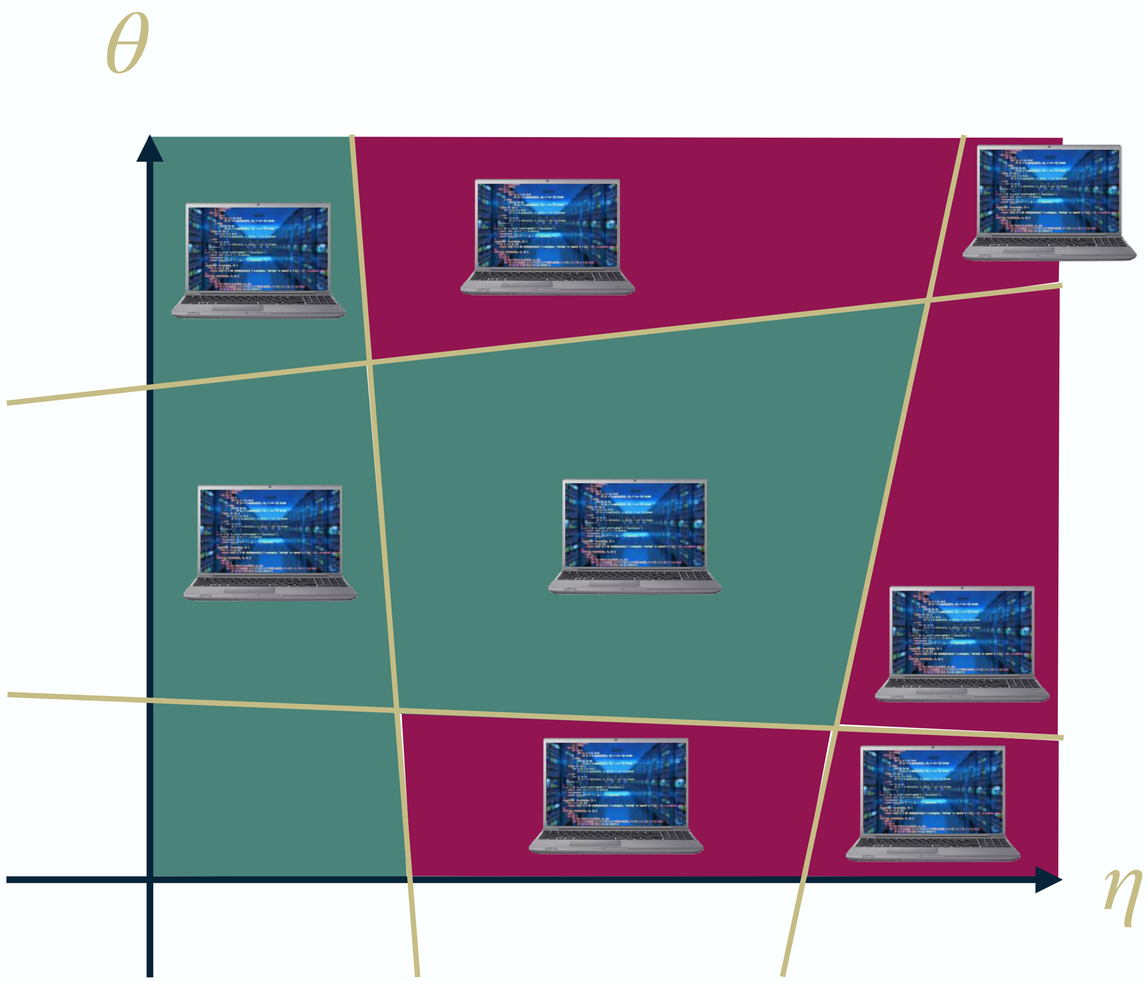






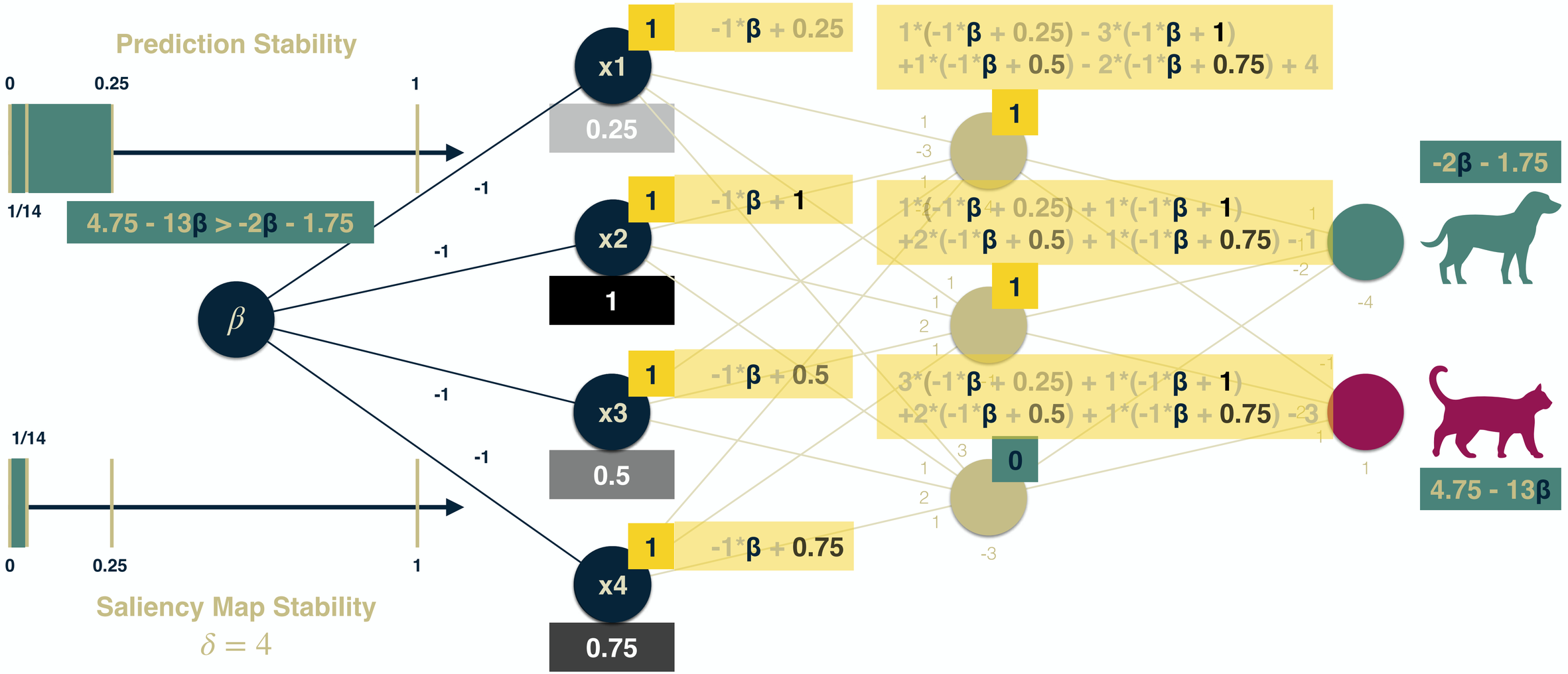
$$-1 * (x1 - 3*x2 + x3 - 2*x4) + 2 * (x1 + x2 + 2*x3 + x4) + 1 * (3* x1 + x2 + 2*x3 + x4) = 4*x1 + 6*x2 + 5*x3 + 5*x4$$

Naïve Breadth-First Search



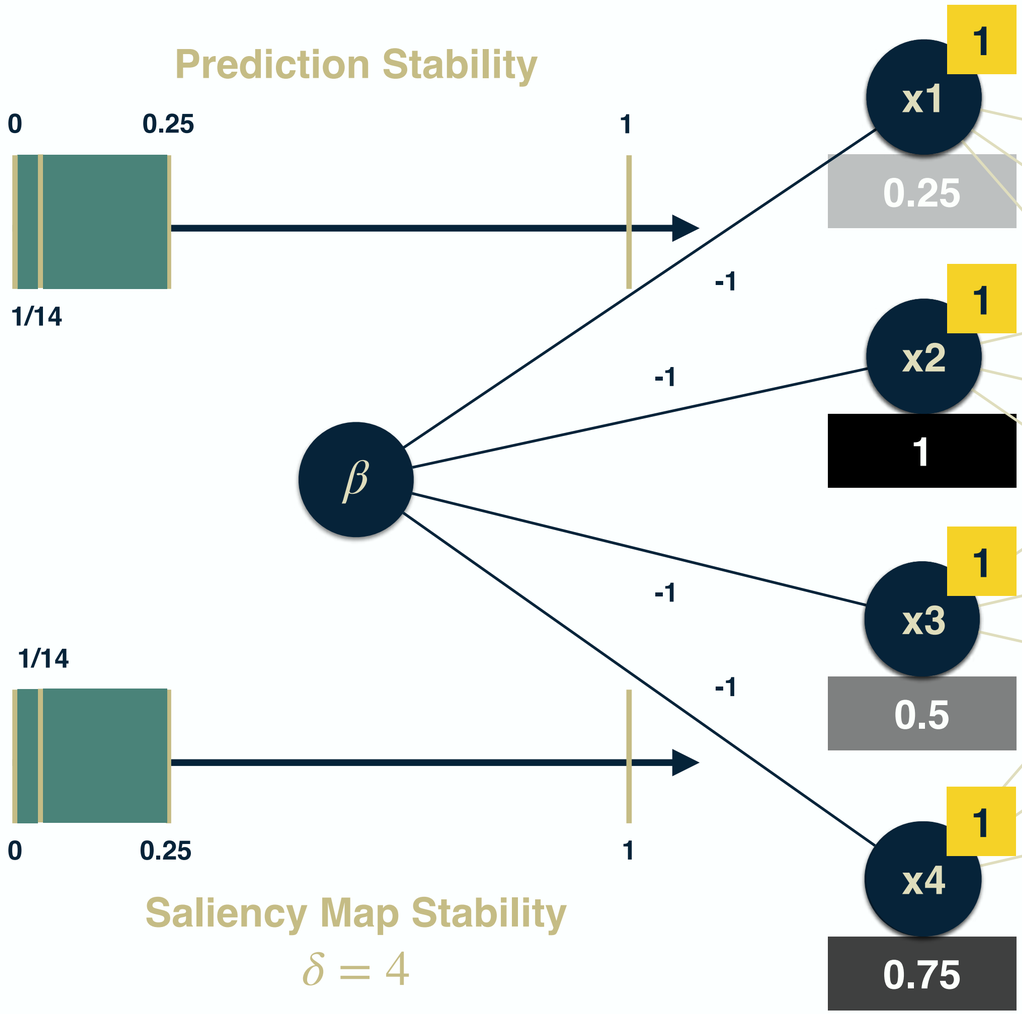
(A Very Small) Example

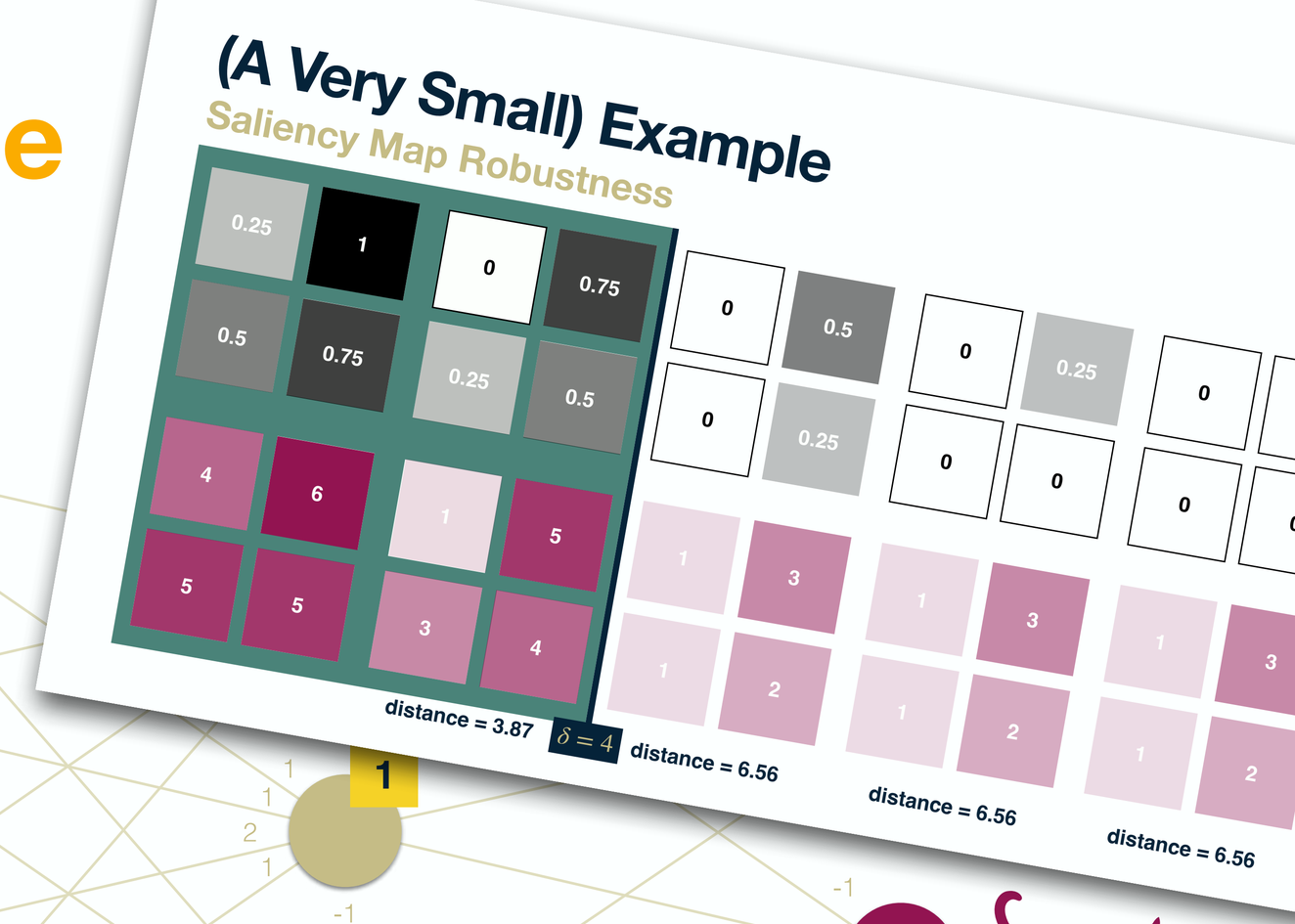
Prediction Stability

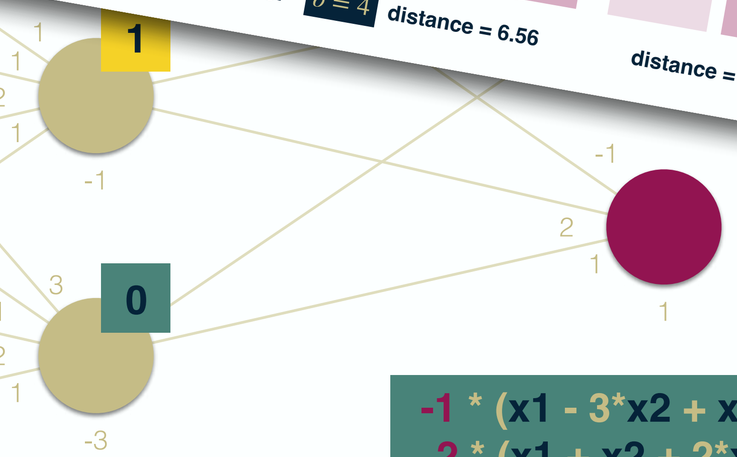


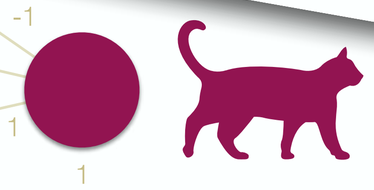
(A Very Small) Example

Saliency Map Stability





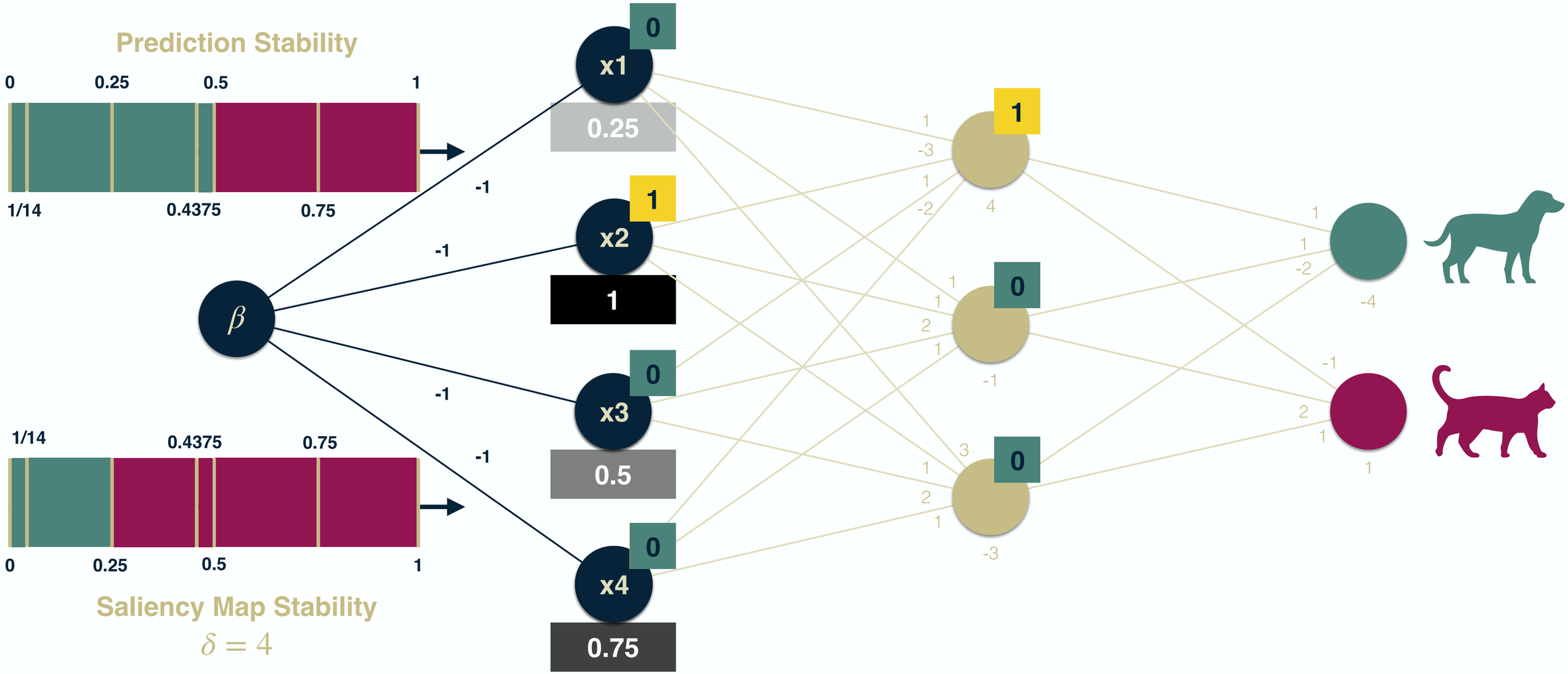




-1 * (x1 - 3*x2 + x3 - 2*x4) +
2 * (x1 + x2 + 2*x3 + x4) +
0 * (3* x1 + x2 + 2*x3 + x4) =
1*x1 + 5*x2 + 3*x3 + 4*x4

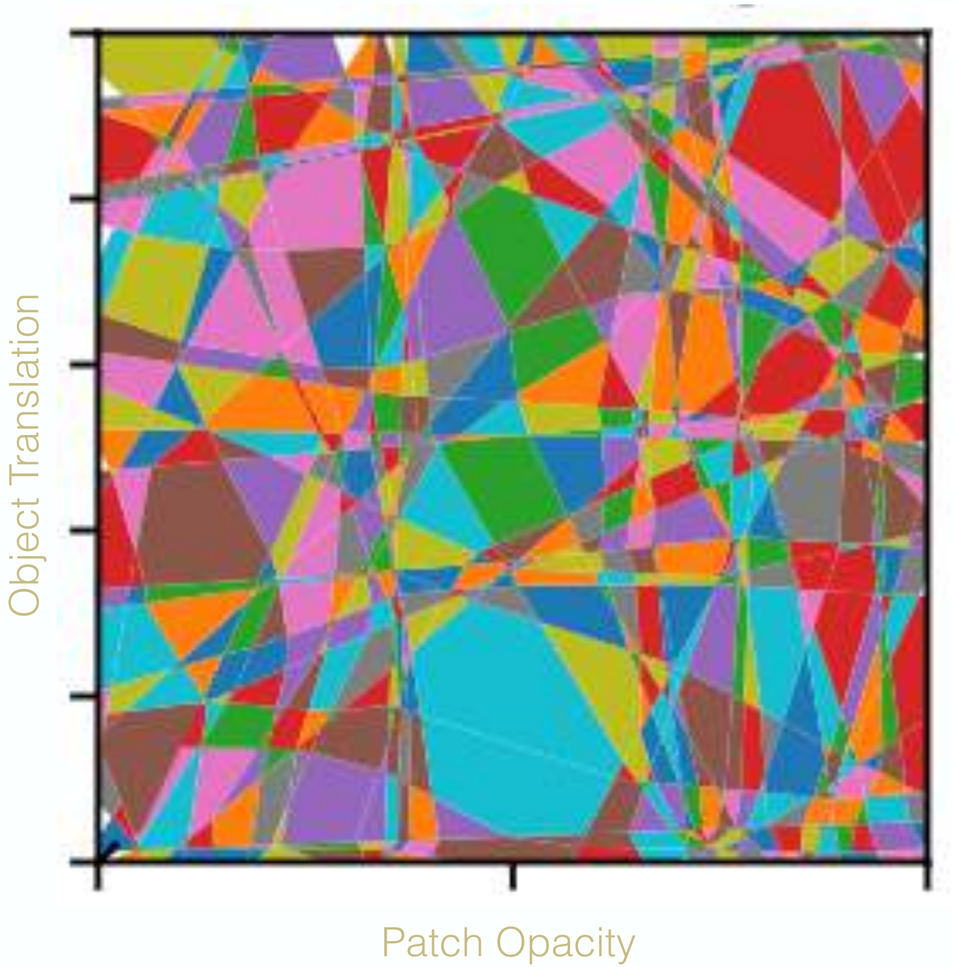
(A Very Small) Example

Naïve Breadth-First Search

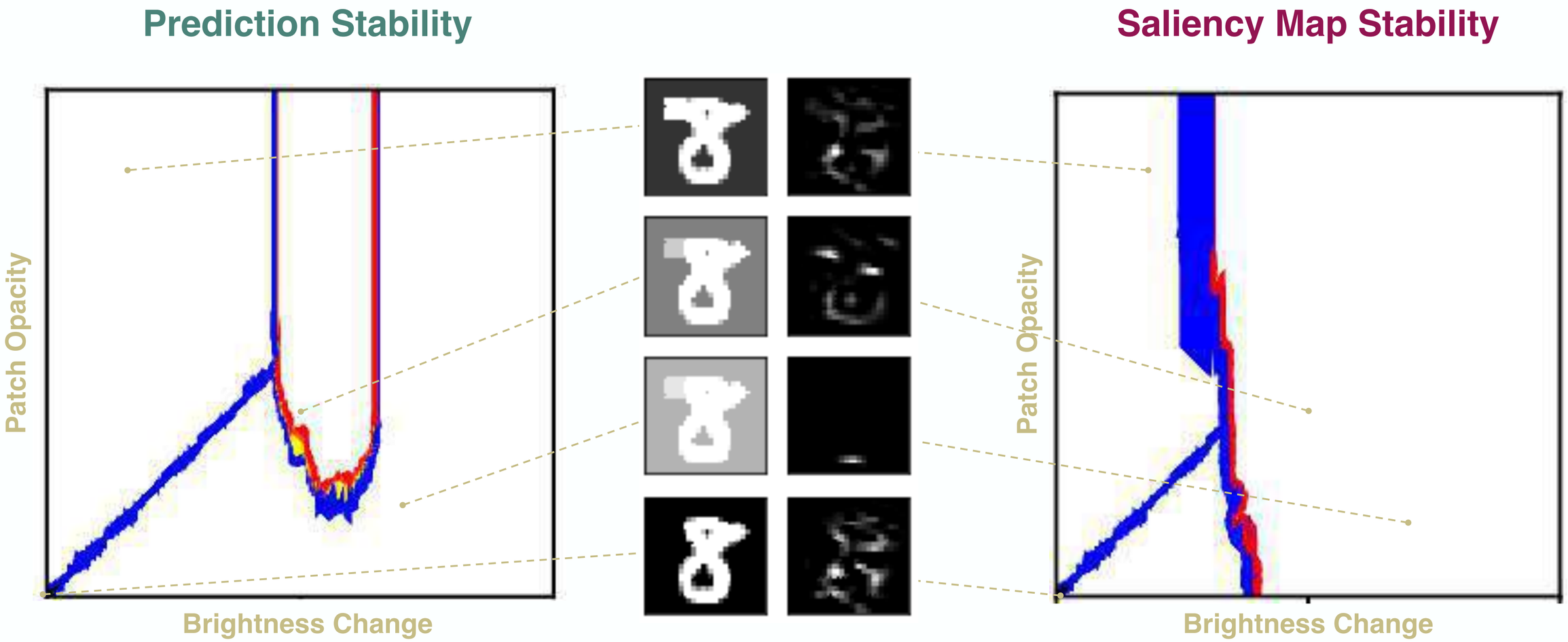


Naïve Breadth-First Search

Too Many Activation Patterns!

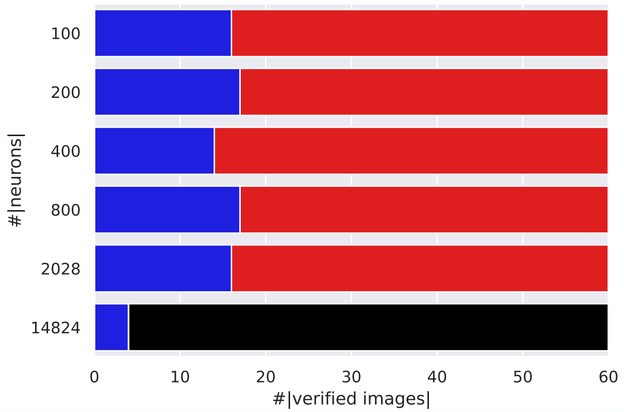


Geometric Boundary Search

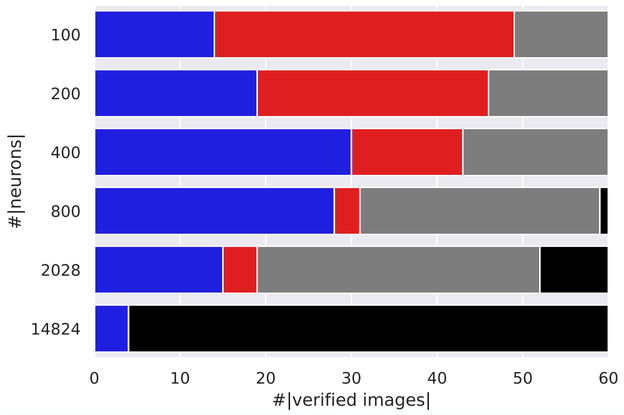


Geometric Boundary Search

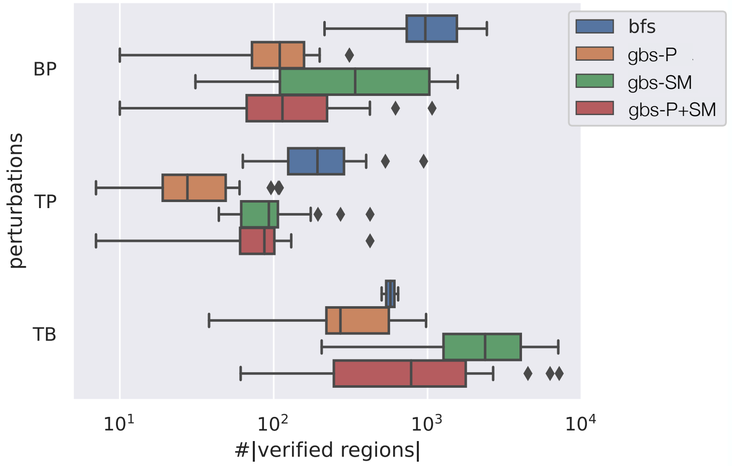
Experimental Results

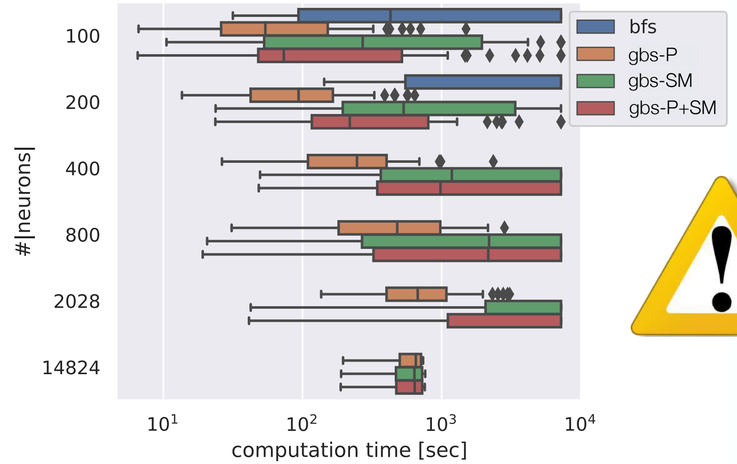


Prediction (P) Stability



Saliency Map (SM) Stability





EXPONENTIAL INCREASE

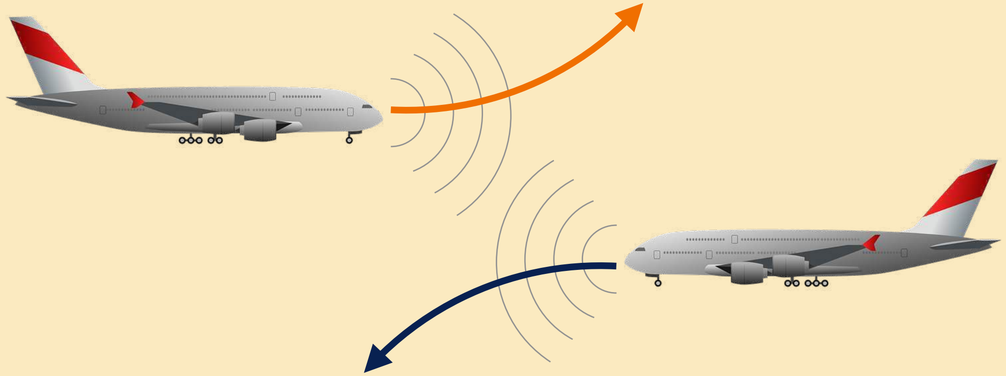
Abstract (Boundary) Search

Stability

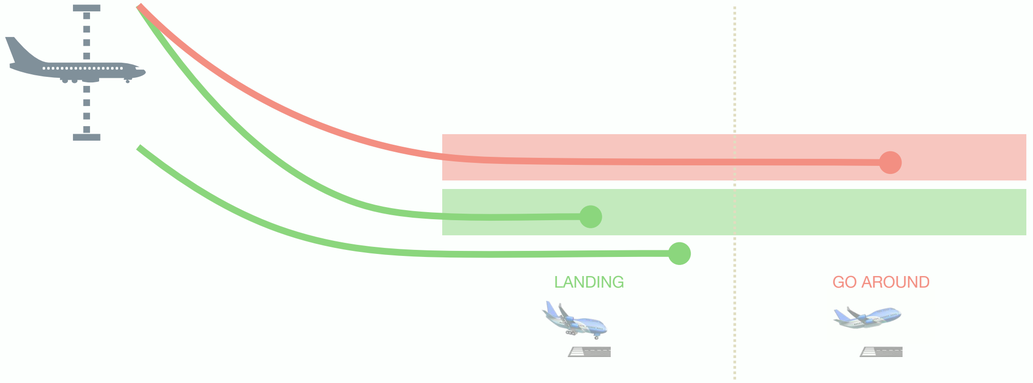
Goal G3 in [Kurd03]

Safety

Goal G4 in [Kurd03]



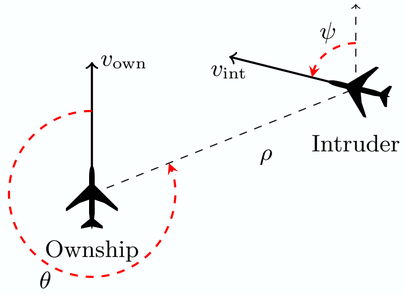
Hypersafety



ACAS Xu [Julian16][Katz17]

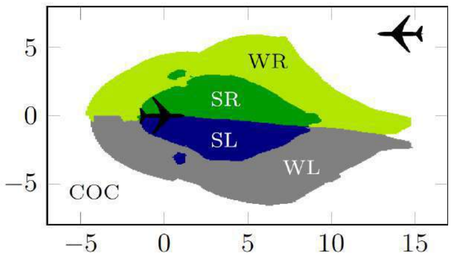
Airborne Collision Avoidance System for Unmanned Aircraft

implemented using **45 feed-forward fully-connected ReLU networks**



5 input **sensor measurements**

- ρ: distance from ownship to intruder
- θ: angle to intruder relative to ownship heading direction
- ψ: heading angle to intruder relative to ownship heading direction
- v_{own}: speed of ownship
- v_{int}: speed of intruder

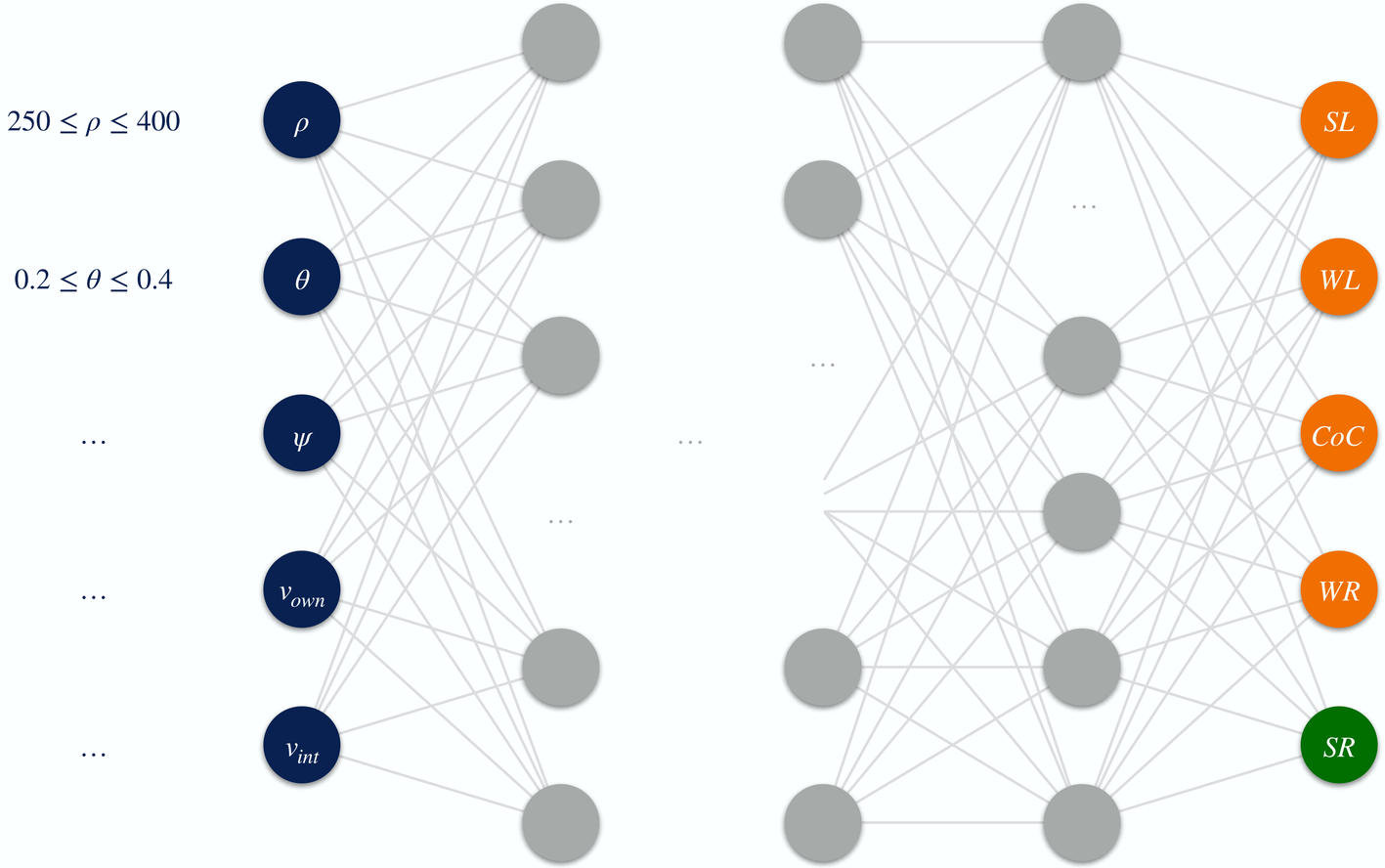


5 output **horizontal advisories**

- Strong Left
- Weak Left
- Clear of Conflict
- Weak Right
- Strong Right

ACAS Xu Properties [Katz17]

Example: “if intruder is **near** and approaching **from the left**, go **Strong Right**”



Safety

Input-Output Properties

$\mathbf{I}$: input specification

$\mathbf{O}$: output specification

$$\mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \stackrel{\text{def}}{=} \{[\![M]\!] \mid \text{SAFE}^{\mathbf{I}}_{\mathbf{O}}([\![M]\!])\}$$

$\mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **satisfy** the input and output specification $\mathbf{I}$ and $\mathbf{O}$

$$\text{SAFE}^{\mathbf{I}}_{\mathbf{O}}(T) \stackrel{\text{def}}{=} \forall t \in T\colon\ t_0 \vDash \mathbf{I} \Rightarrow t_\omega \vDash \mathbf{O}$$

Theorem

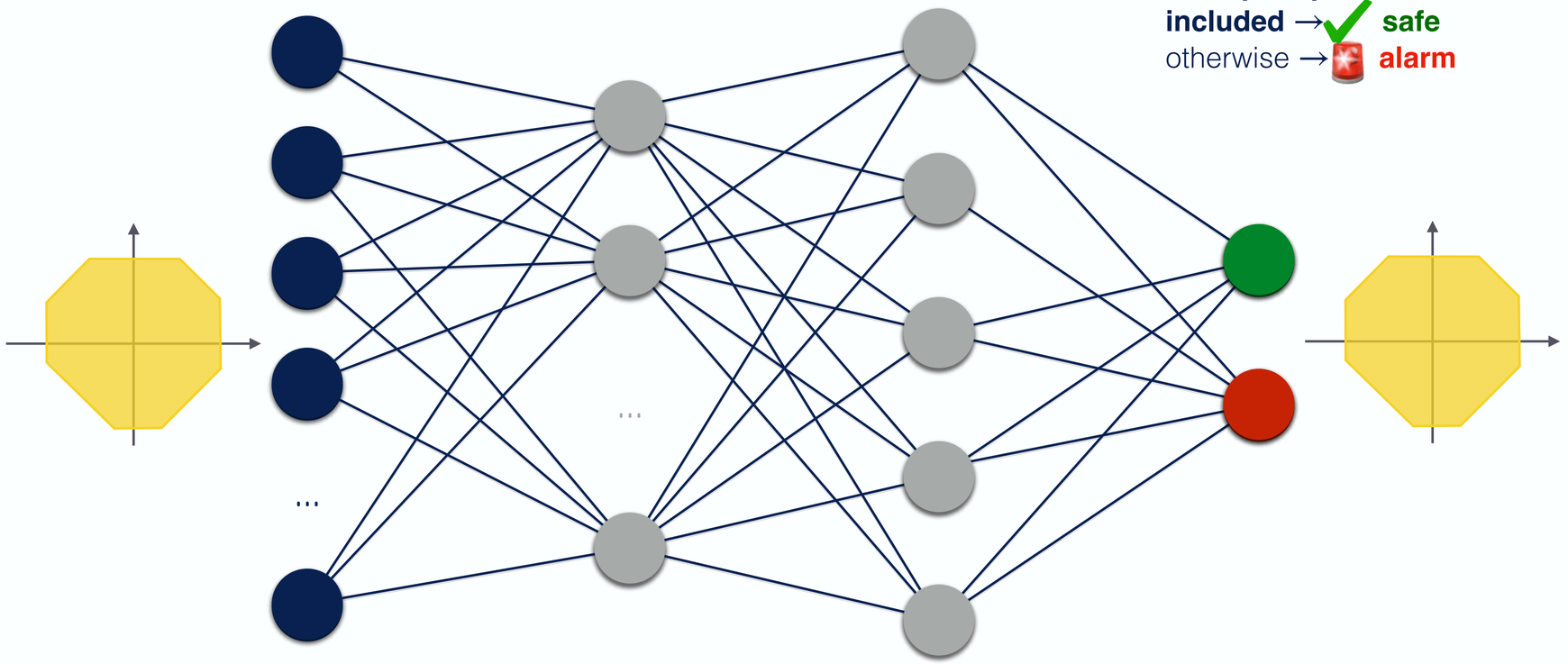
$$M \vDash \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Leftrightarrow \{[\![M]\!]\} \subseteq \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

Corollary

$$M \vDash \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Leftrightarrow [\![M]\!] \subseteq \bigcup \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

Forward Analysis

② check output for **inclusion** in **output specification** $\mathbf{O}$:
included → ✓ **safe**
otherwise → 🚨 **alarm**

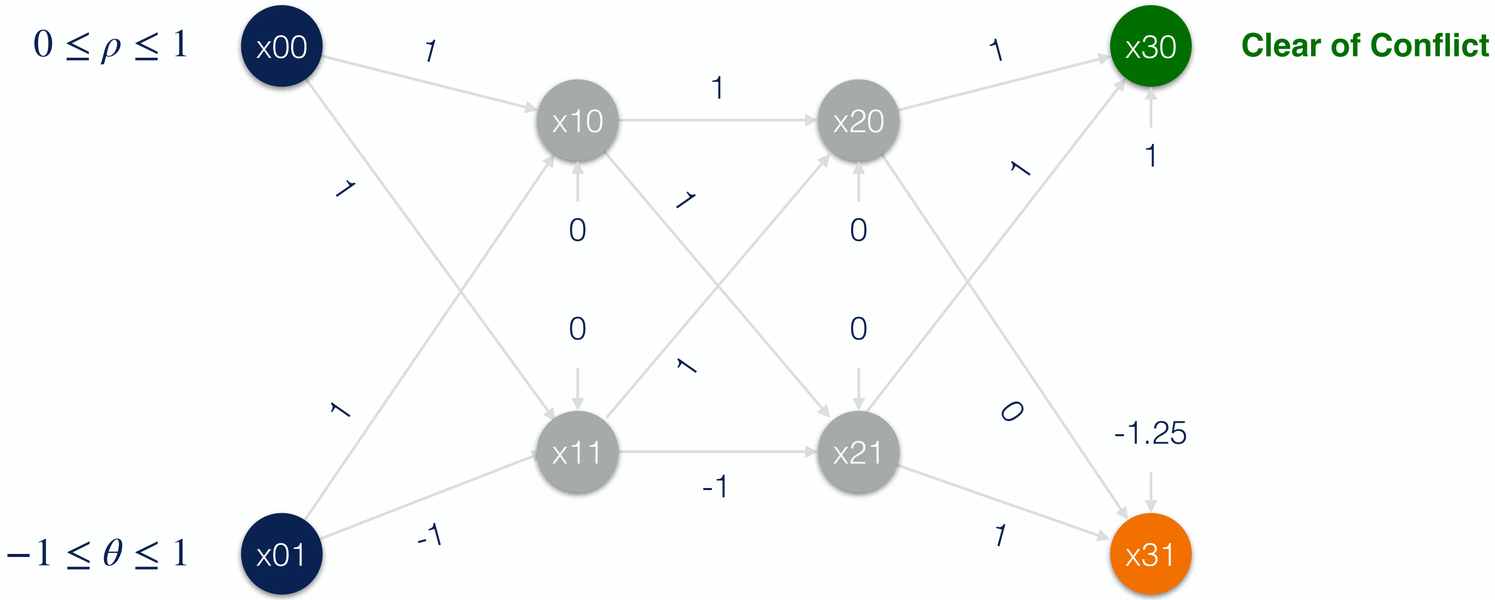


① proceed **forwards** **from an abstraction** of the input specification $\mathbf{I}$

Theorem

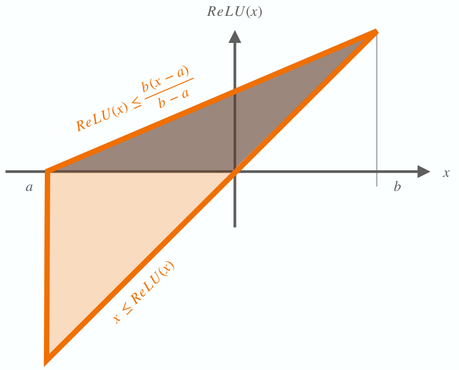
$$[\![M]\!] \subseteq [\![M]\!]^\natural \subseteq \bigcup \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Rightarrow M \vDash \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

Example


$0 \leq \rho \leq 1$
$-1 \leq \theta \leq 1$
x00
x01
x10
x11
x20
x21
x30
x31
1
1
1
1
1
1
1
1
0
0
0
0
1
1
-1
-1
0
1
-1.25
Clear of Conflict

Abstraction #3: DeepPoly

DeepPoly Abstraction [Singh19]



$$x_{10} \mapsto \begin{cases} [x_{10},\ \frac{2}{3} \cdot x_{10} + \frac{2}{3}] \\ [-\mathbf{1},\ \mathbf{2}] \end{cases}$$

ReLU

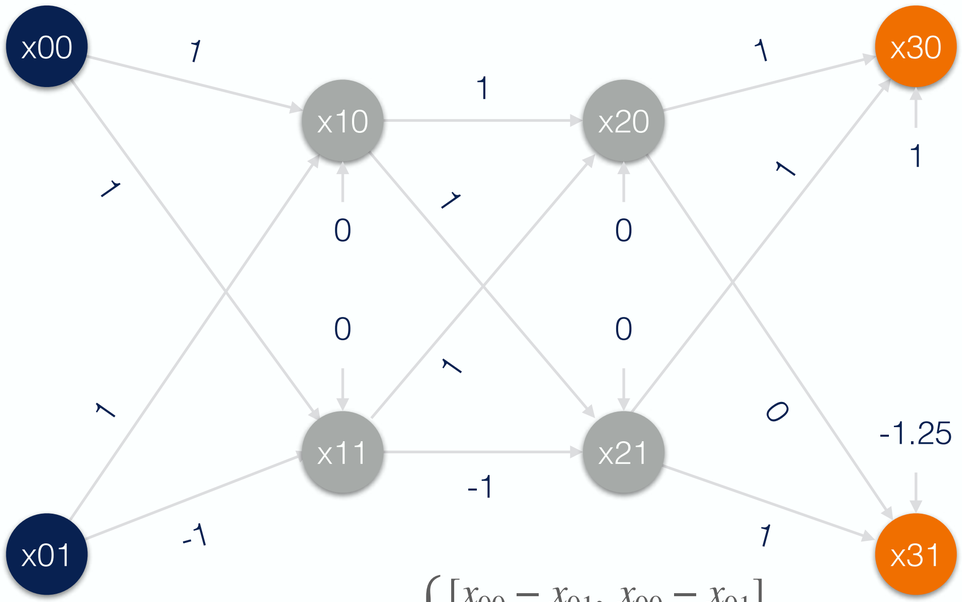
$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01},\ x_{00} + x_{01}] \\ [-\mathbf{1},\ \mathbf{2}] \end{cases}$$

$$0 \leq \rho \leq 1$$

$$x_{00} \mapsto \begin{cases} [x_{00},\ x_{00}] \\ [\mathbf{0},\ \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} [x_{01},\ x_{01}] \\ [-\mathbf{1},\ \mathbf{1}] \end{cases}$$

$$-1 \leq \theta \leq 1$$



Clear of Conflict

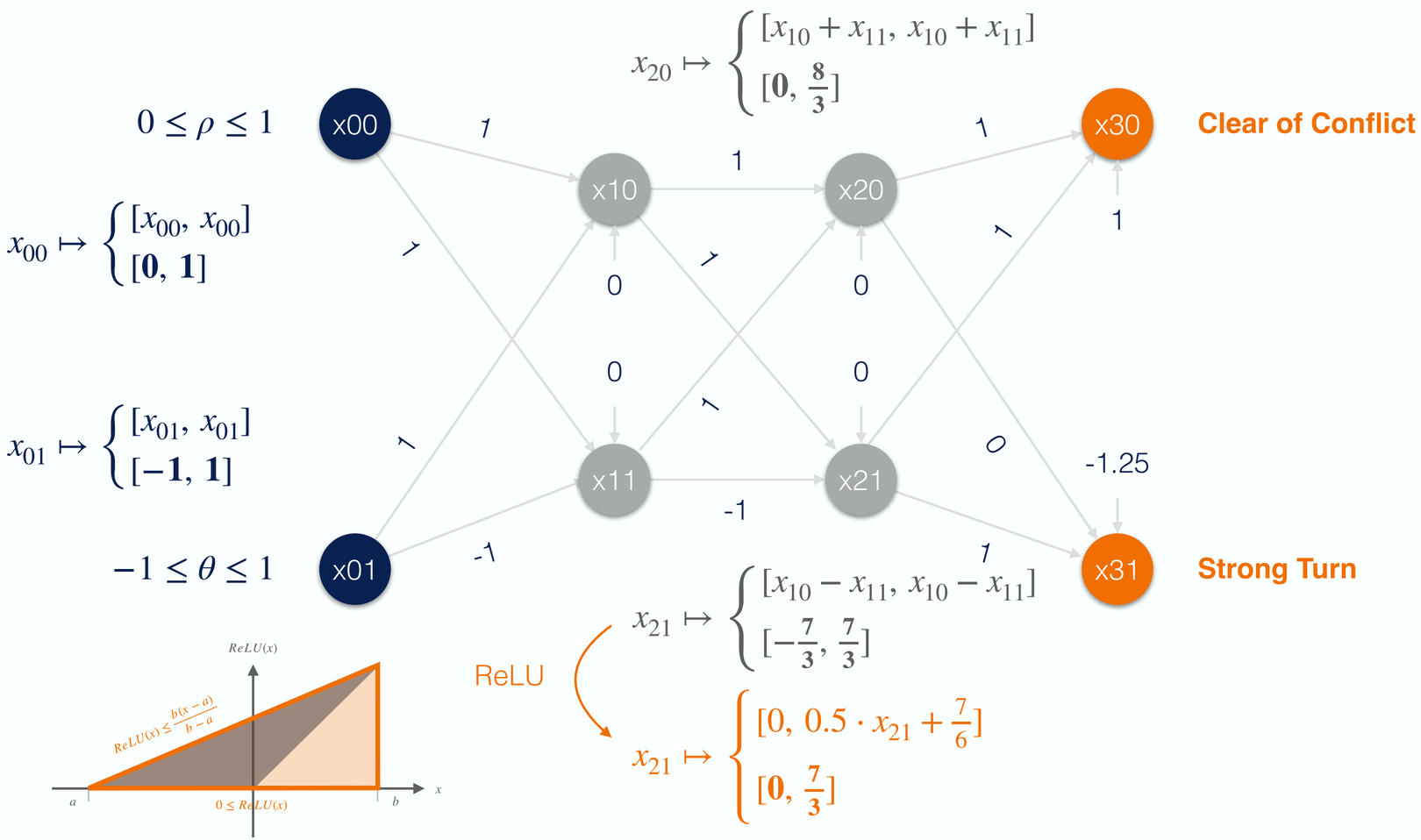
Strong Turn

$$x_{11} \mapsto \begin{cases} [x_{00} - x_{01},\ x_{00} - x_{01}] \\ [-\mathbf{1},\ \mathbf{2}] \end{cases}$$

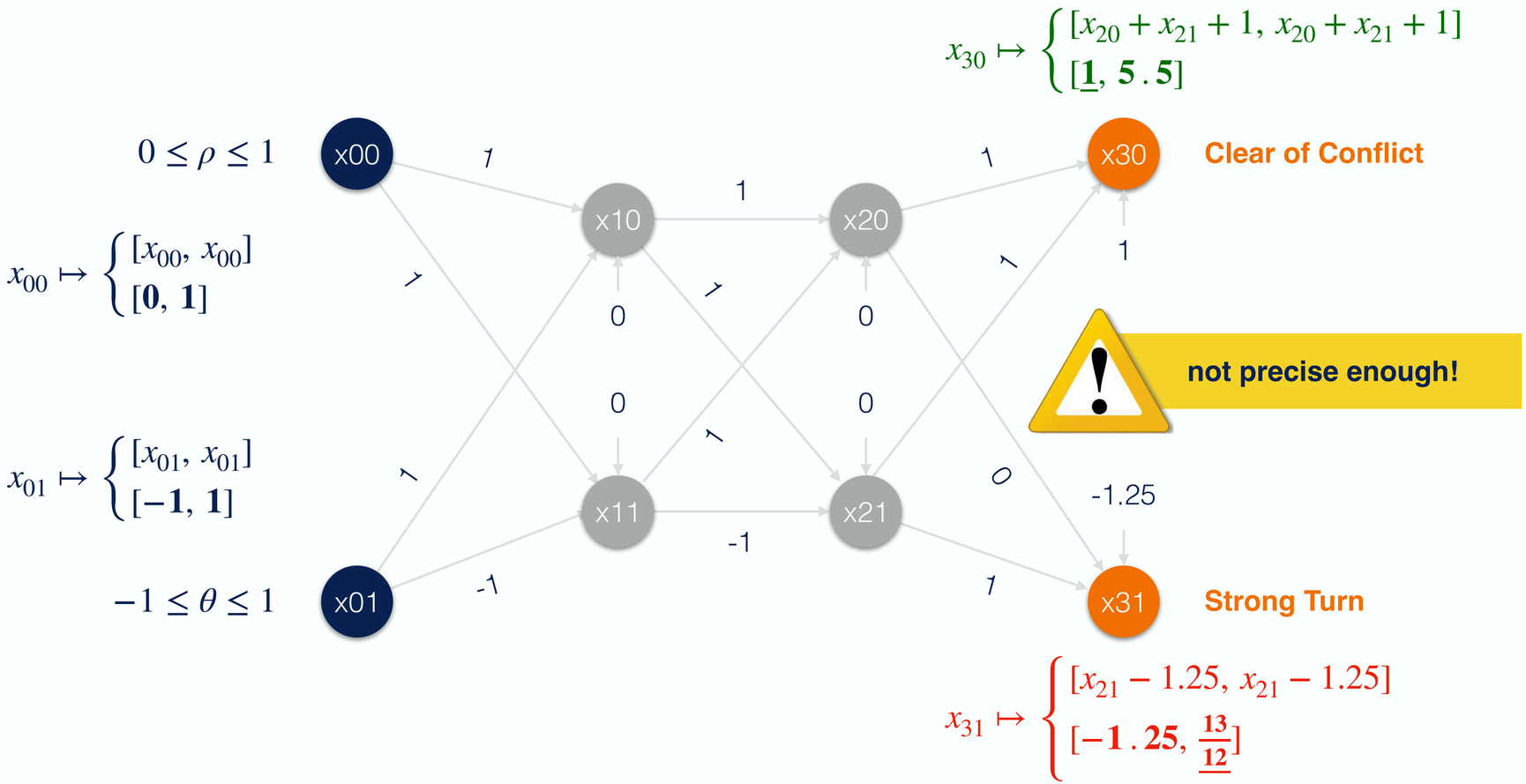
ReLU

$$x_{11} \mapsto \begin{cases} [x_{11},\ \frac{2}{3} \cdot x_{11} + \frac{2}{3}] \\ [-\mathbf{1},\ \mathbf{2}] \end{cases}$$

DeepPoly Abstraction [Singh19]

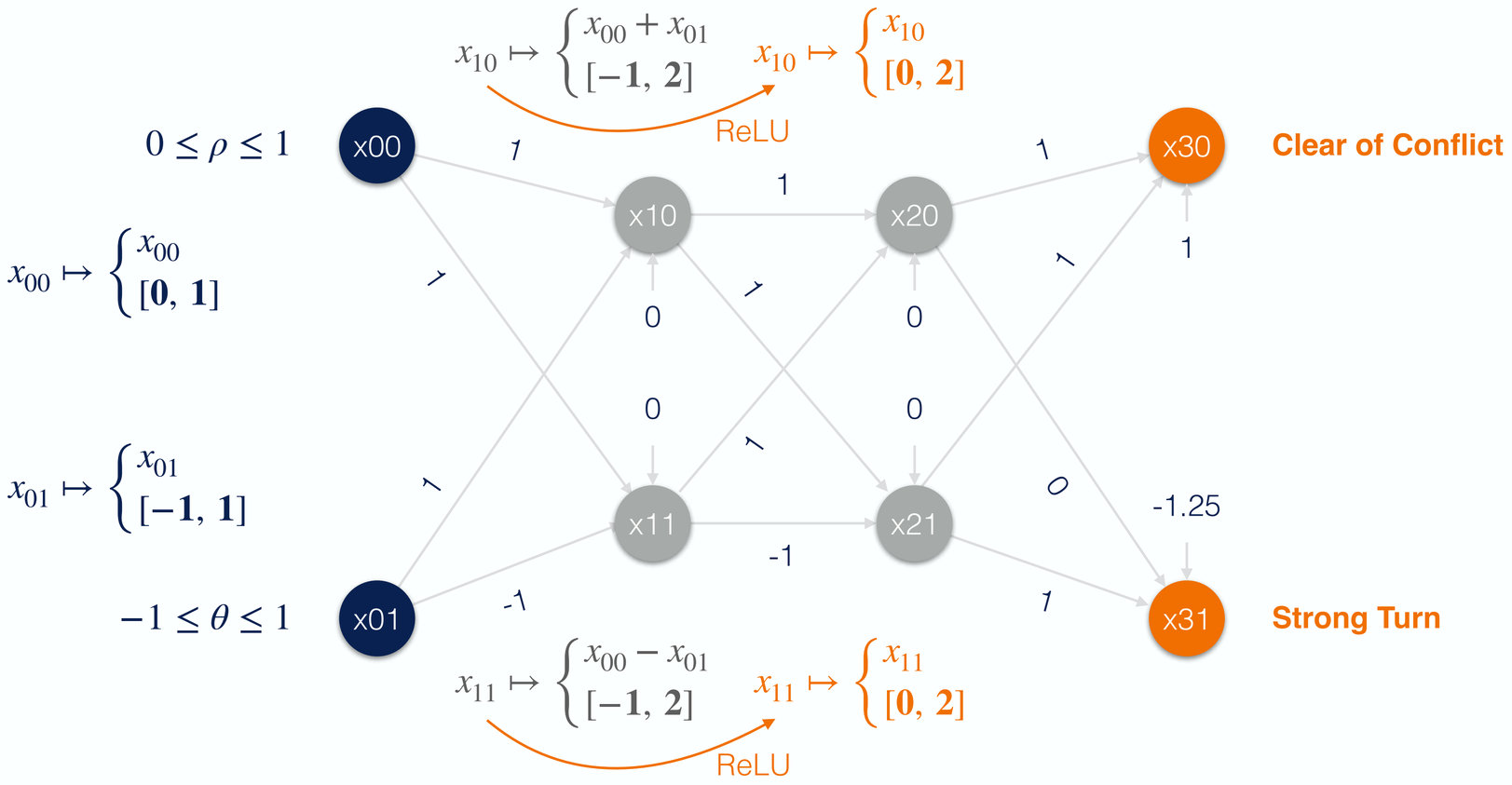


DeepPoly Abstraction [Singh19]

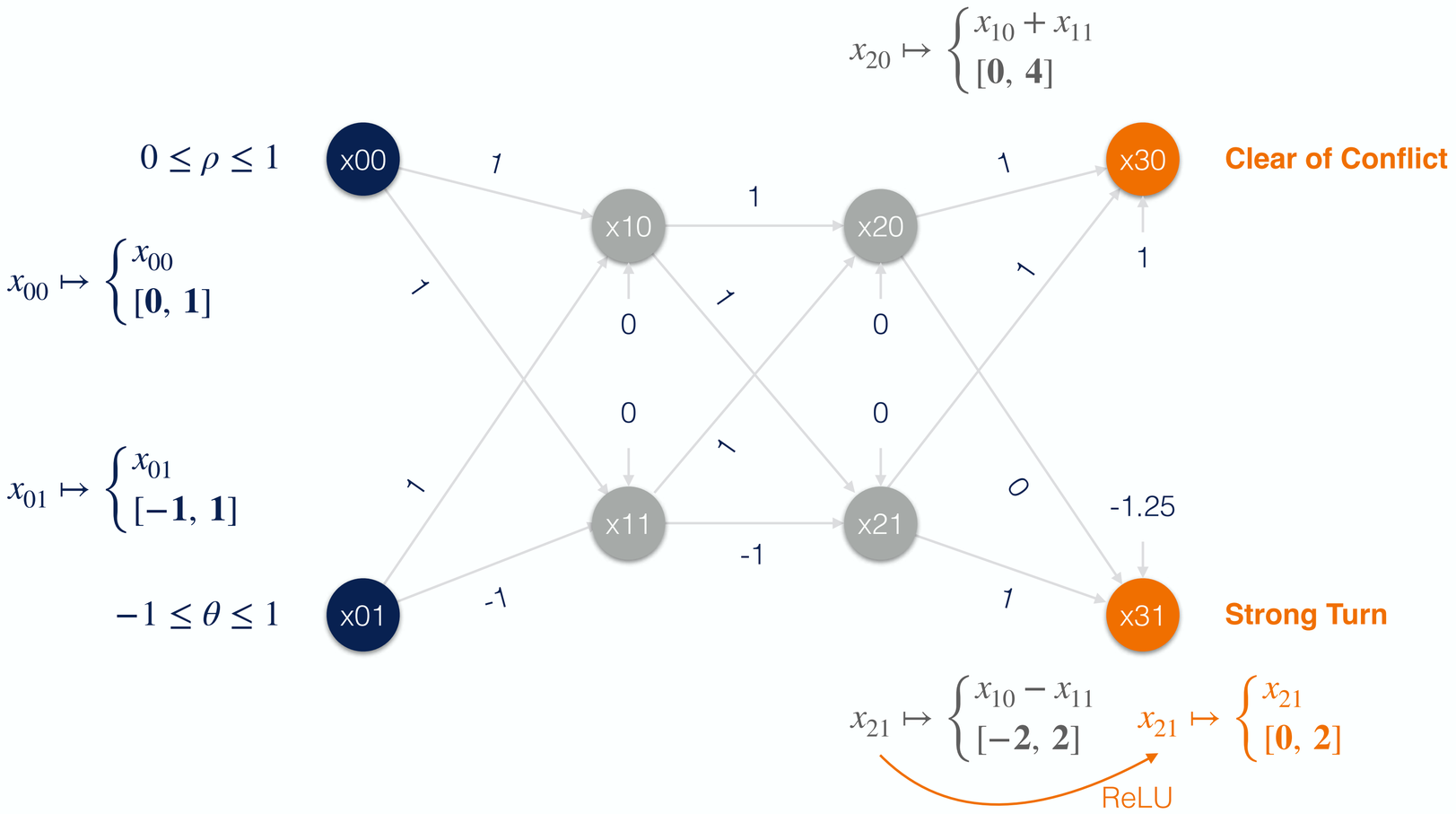


Abstraction #2: Symbolic

Symbolic Abstraction [Li19]



Symbolic Abstraction [Li19]

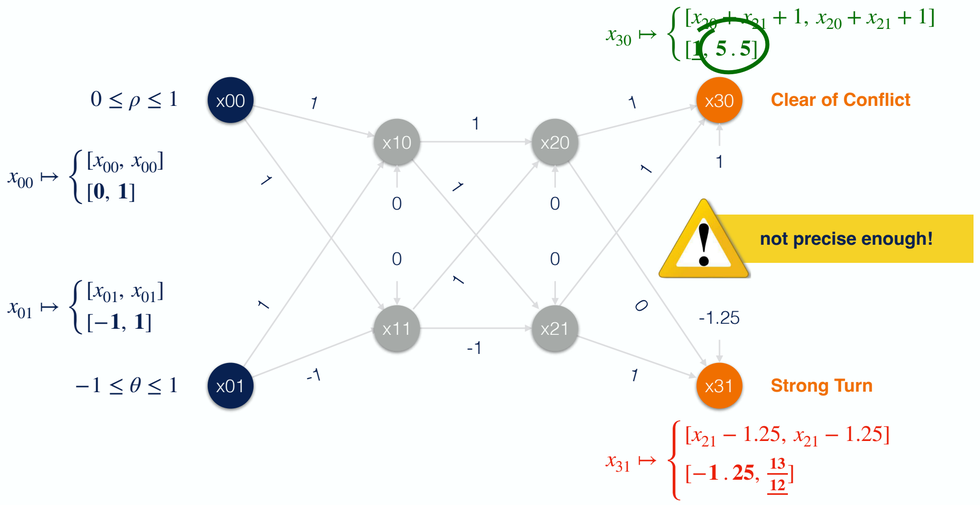


Symbolic Abstraction [Li19]

$$x_{30} \mapsto \begin{cases} x_{10} + x_{11} + x_{21} + 1 \\ [\underline{\mathbf{1}}, \mathbf{7}] \end{cases}$$

Clear of Conflict

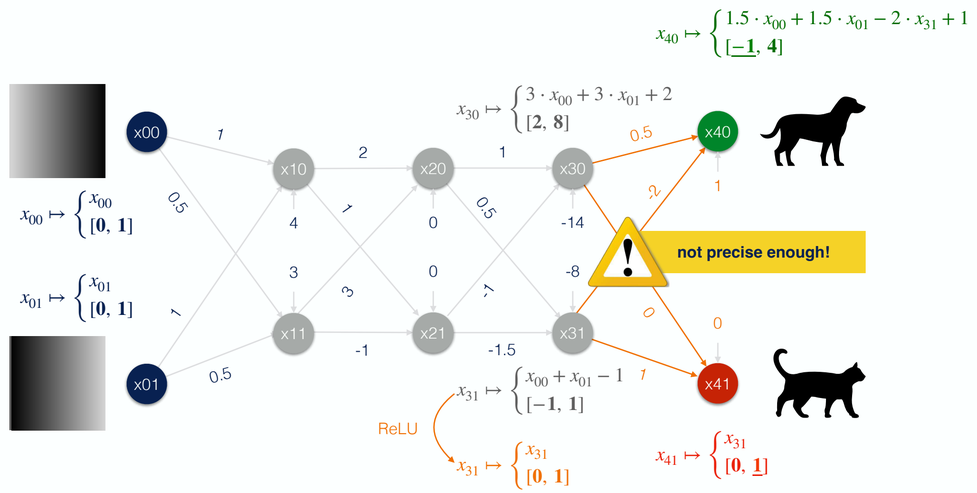
DeepPoly Abstraction



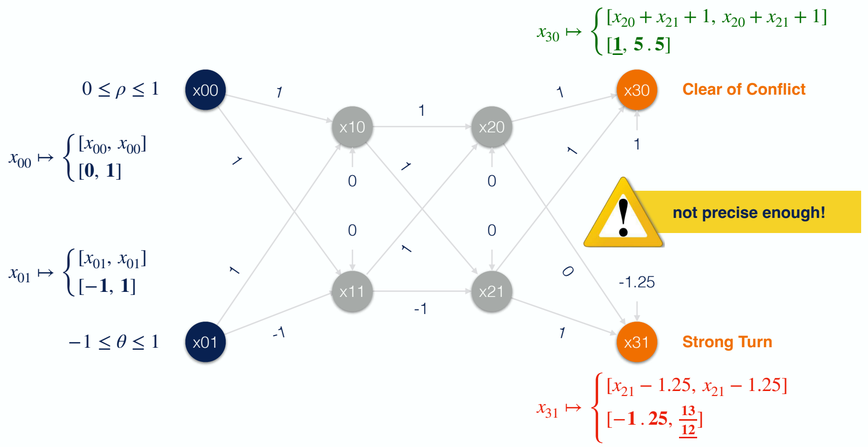
$$x_{21} - 1.25$$
$$[-\mathbf{1.25}, \underline{\mathbf{0.75}}]$$

Symbolic Abstraction

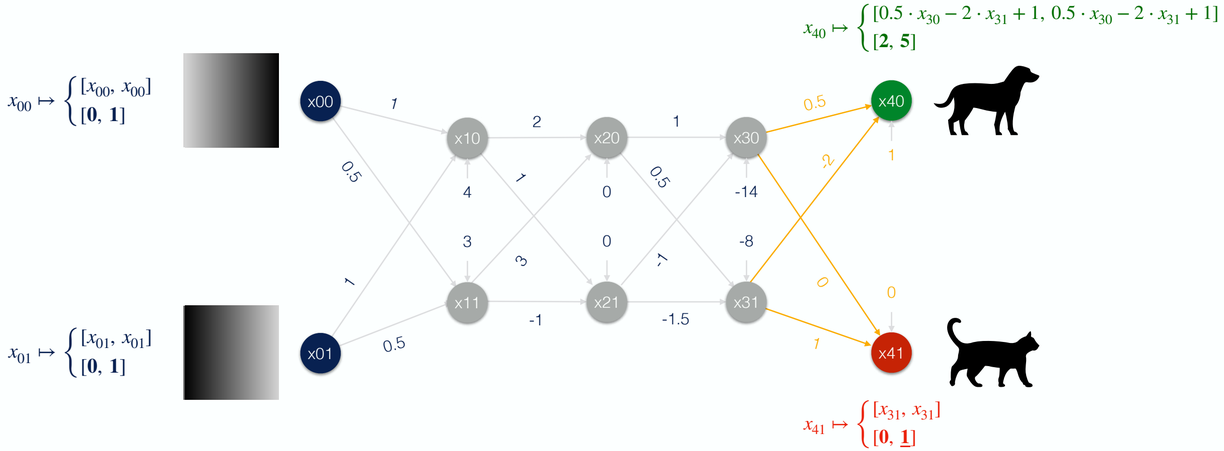
Modified Example



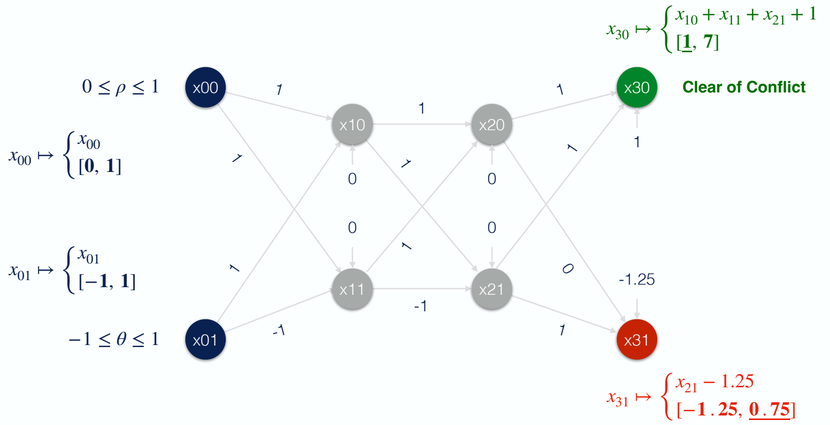
DeepPoly Abstraction



DeepPoly Abstraction



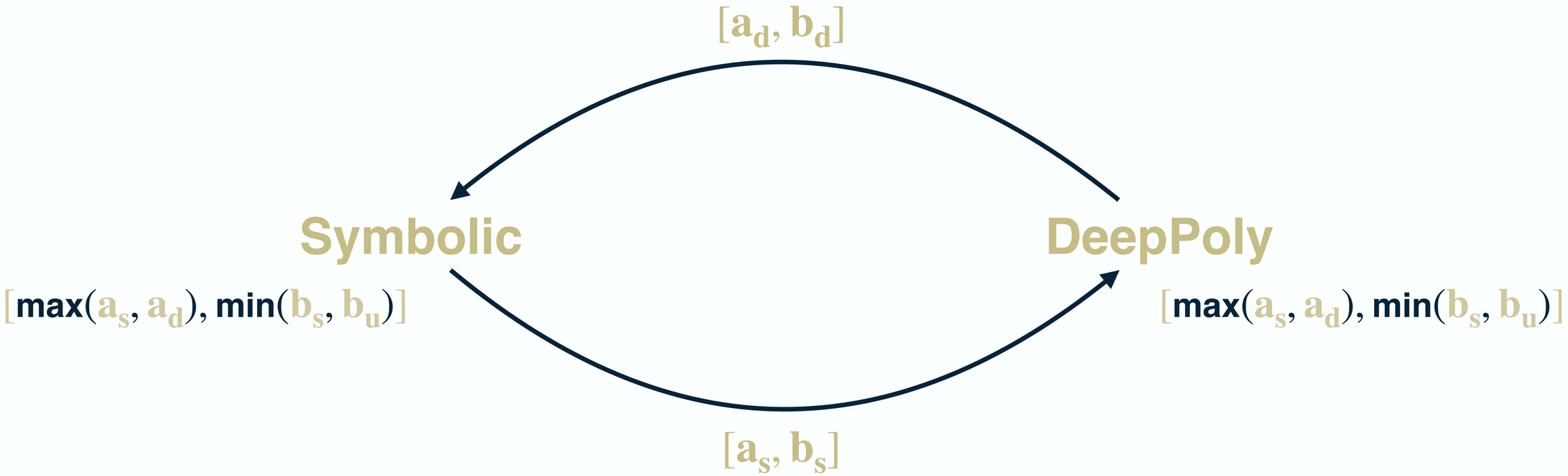
Symbolic Abstraction



Abstraction #4: Product

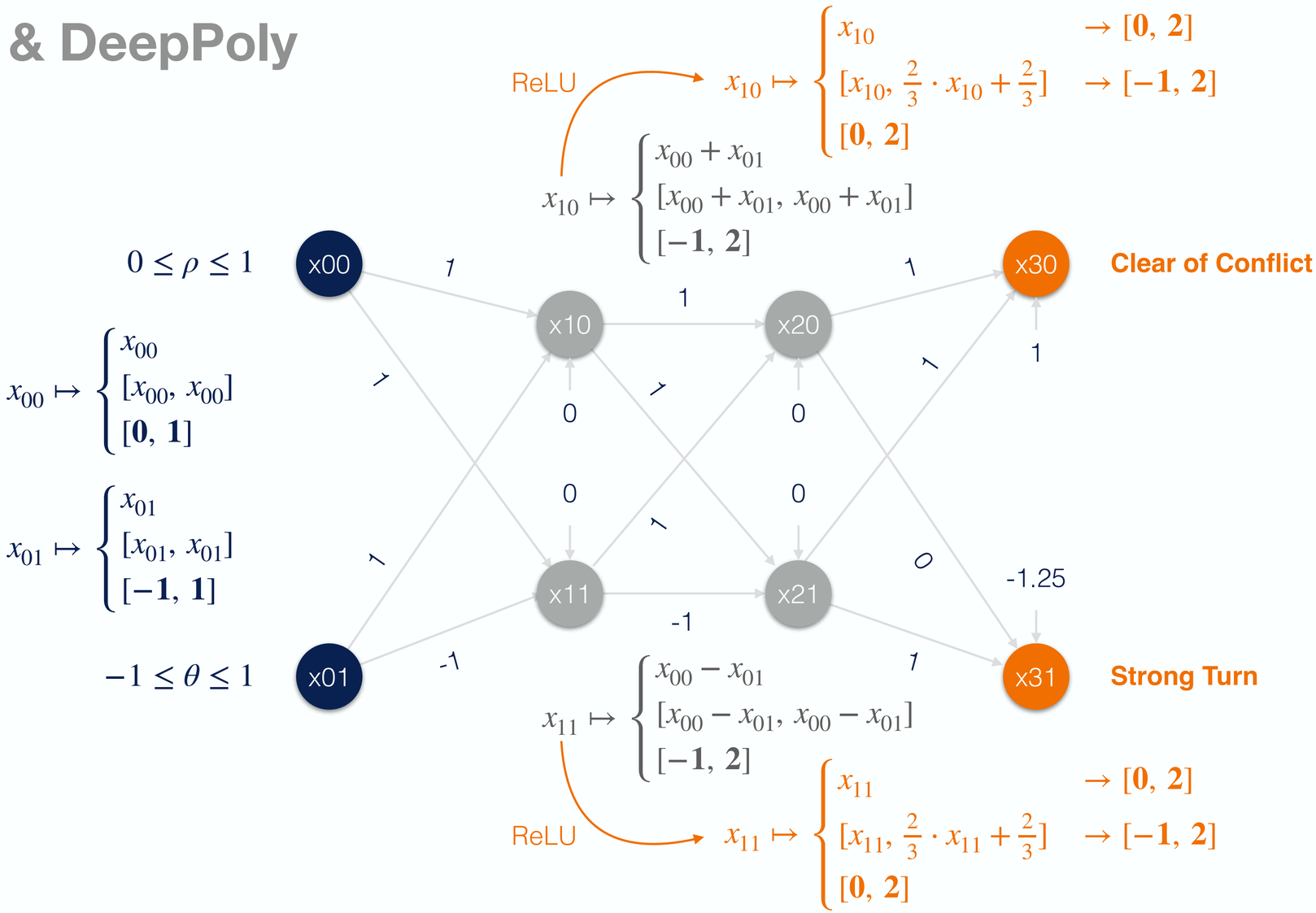
Product Abstraction [Mazzucato21]

Symbolic & DeepPoly



Product Abstraction [Mazzucato21]

Symbolic & DeepPoly



Product Abstraction [Mazzucato21]

Symbolic & DeepPoly

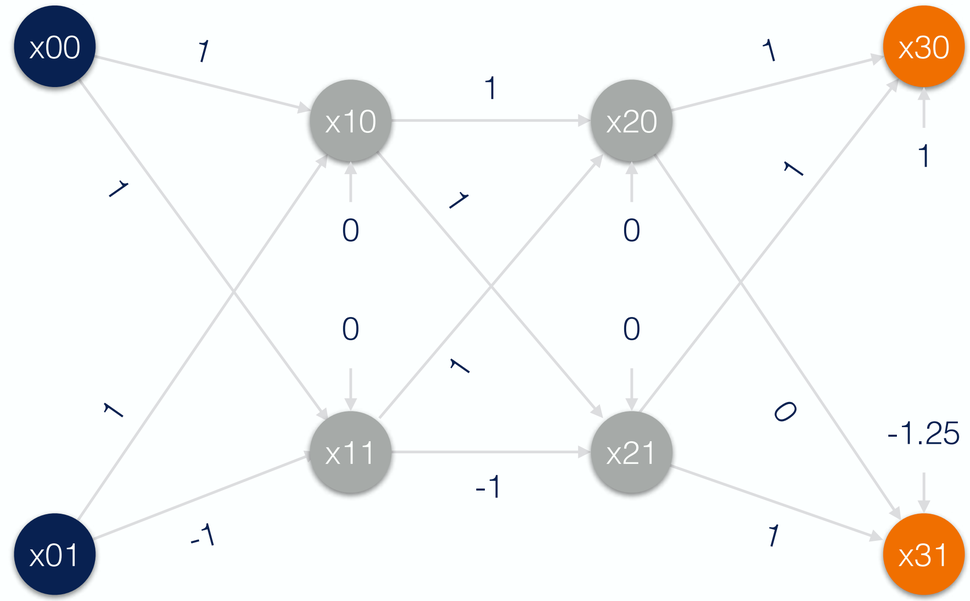
$$x_{20} \mapsto \begin{cases} x_{10} + x_{11} & \to [\mathbf{0}, \mathbf{4}] \\ [x_{10} + x_{11},\ x_{10} + x_{11}] & \to [\mathbf{0}, \frac{\mathbf{8}}{\mathbf{3}}] \\ [\mathbf{0}, \frac{\mathbf{8}}{\mathbf{3}}] \end{cases}$$

$0 \leq \rho \leq 1$

$$x_{00} \mapsto \begin{cases} x_{00} \\ [x_{00},\ x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} x_{01} \\ [x_{01},\ x_{01}] \\ [-\mathbf{1}, \mathbf{1}] \end{cases}$$

$-1 \leq \theta \leq 1$



$$x_{21} \mapsto \begin{cases} x_{10} - x_{11} & \to [-\mathbf{2}, \mathbf{2}] \\ [x_{10} - x_{11},\ x_{10} - x_{11}] & \to [-\frac{\mathbf{7}}{\mathbf{3}}, \frac{\mathbf{7}}{\mathbf{3}}] \\ [-\mathbf{2}, \mathbf{2}] \end{cases}$$

ReLU

$$x_{21} \mapsto \begin{cases} x_{21} & \to [\mathbf{0}, \mathbf{2}] \\ [0,\ 0.5 \cdot x_{21} + 0.5] & \to [\mathbf{0}, \frac{\mathbf{5}}{\mathbf{3}}] \\ [\mathbf{0}, \frac{\mathbf{5}}{\mathbf{3}}] \end{cases}$$

Product Abstraction [Mazzucato21]

Symbolic & DeepPoly

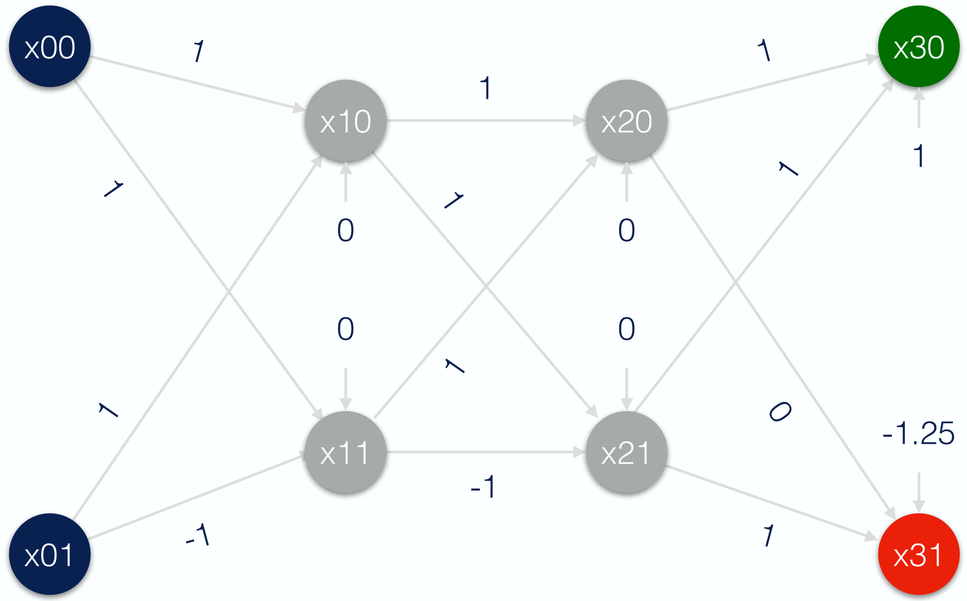
$$x_{30} \mapsto \begin{cases} x_{10} + x_{11} + x_{21} + 1 & \rightarrow [\mathbf{1}, \frac{\mathbf{20}}{\mathbf{3}}] \\ [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] & \rightarrow [\mathbf{1}, \mathbf{4.5}] \\ [\underline{\mathbf{1}}, \mathbf{4.5}] \end{cases}$$

$$0 \leq \rho \leq 1$$

$$x_{00} \mapsto \begin{cases} x_{00} \\ [x_{00},\ x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

$$x_{01} \mapsto \begin{cases} x_{01} \\ [x_{01},\ x_{01}] \\ [\mathbf{-1}, \mathbf{1}] \end{cases}$$

$$-1 \leq \theta \leq 1$$



Clear of Conflict

$$x_{31} \mapsto \begin{cases} x_{21} - 1.25 & \rightarrow [\mathbf{-1.25}, \frac{\mathbf{5}}{\mathbf{12}}] \\ [x_{21} - 1.25,\ x_{21} - 1.25] & \rightarrow [\mathbf{-1.25}, \frac{\mathbf{5}}{\mathbf{12}}] \\ [\mathbf{-1.25}, \underline{\frac{\mathbf{5}}{\mathbf{12}}}] \end{cases}$$

Going Farther: Complete Methods

Star Sets

Complete Abstraction

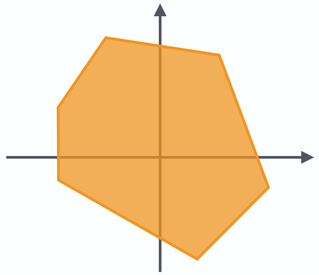
use union of efficient representations of bounded convex polyhedra

Follow-up Work

H.-D. Tran et al. - Verification of Deep Convolutional Neural Networks Using ImageStars (CAV 2020)

$\Theta \stackrel{\text{def}}{=} \langle c, V, P \rangle$

$c \in \mathscr{R}^n$: center
$V = \{v_1, \ldots, v_m\}$: basis vectors in $\mathscr{R}^n$
$P\colon \mathscr{R}^m \rightarrow \{ \bot , \top \}$: predicate



$$[\![\Theta]\!] = \{x \mid x = c + \sum_{i=1}^{m} \alpha_i v_i \text{ such that } P(\alpha_1, \ldots, \alpha_m) = \top \}$$

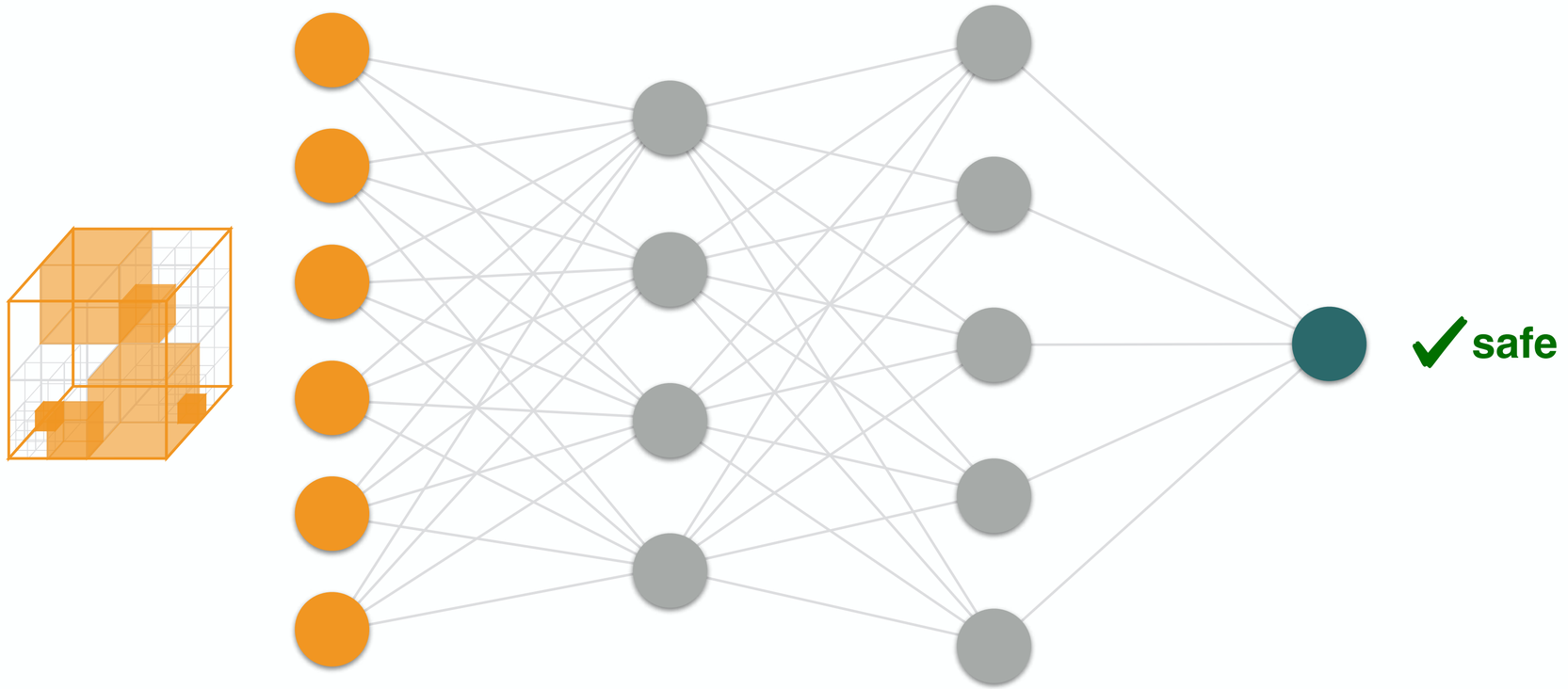
- fast and cheap **affine mapping operations** → neural network layers
- inexpensive **intersections with half-spaces** → ReLU activations

H.-D. Tran et al. - Star-Based Reachability Analysis of Deep Neural Networks (FM 2018)

ReluVal

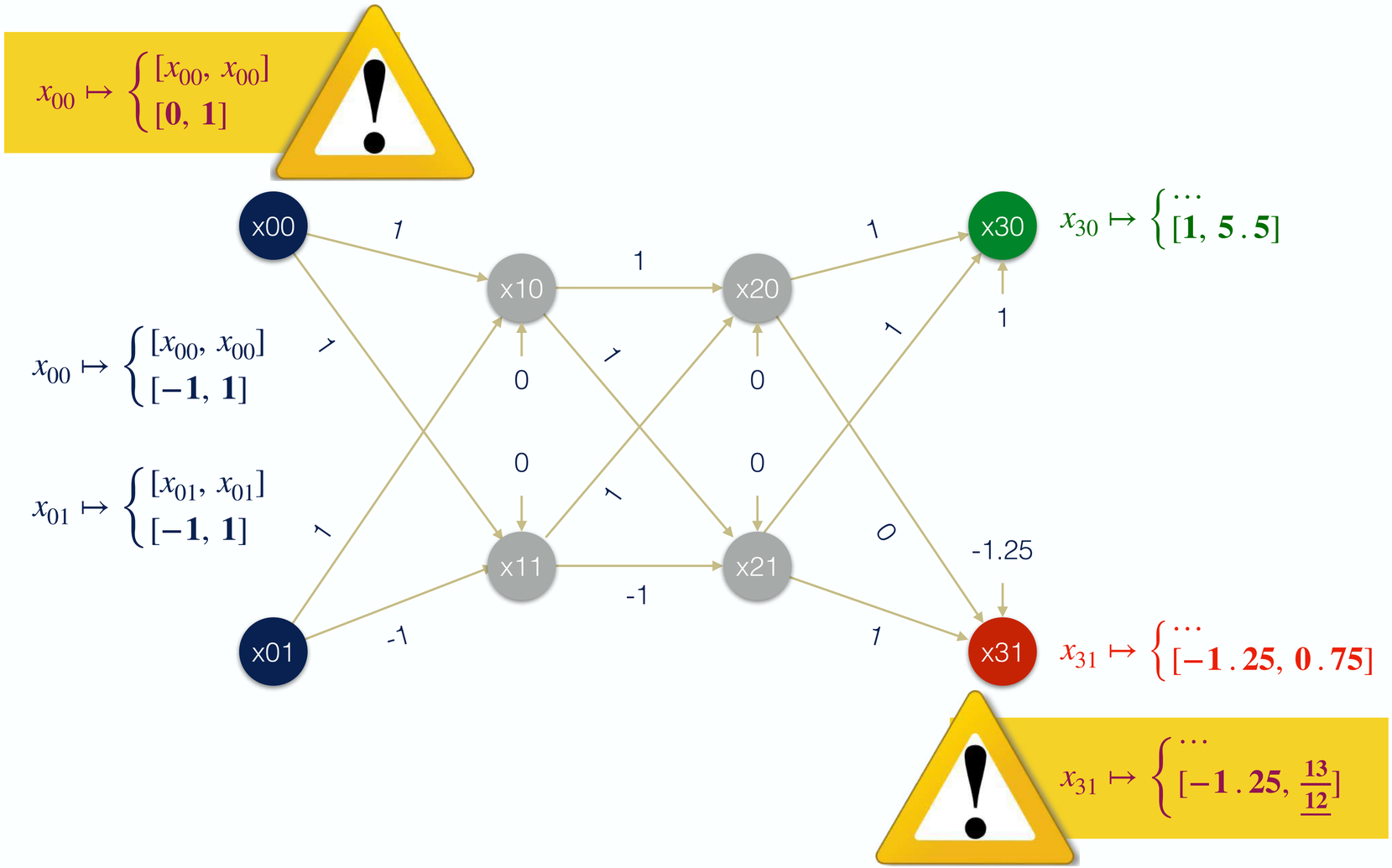
Asymptotically Complete Method

symbolic propagation
+ iterative input refinement



S. Wang et al. - Formal Security Analysis of Neural Networks Using Symbolic Intervals (USENIX Security 2018)

DeepPoly Abstraction [Singh19] + Input Refinement



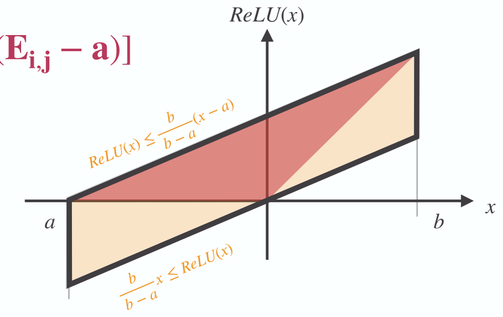
Neurify

Asymptotically Complete Method

symbolic propagation
+ convex ReLU approximation +
iterative input/ReLU refinement

$$x_{i,j} \mapsto \begin{cases} [\sum_k c_{0,k} \cdot x_{0,k} + c,\ \sum_k d_{0,k} \cdot x_{0,k} + d] & c_{0,k}, c, d_{0,k}, d \in \mathscr{R} \\ [a,\ b] & a, b \in \mathscr{R} \end{cases}$$

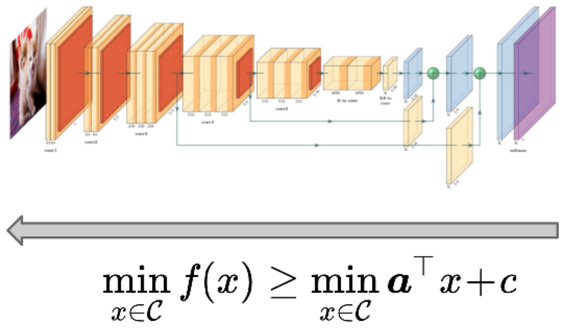
$$x_{i,j} \mapsto \begin{cases} [\mathbf{E_{i,j}}, \mathbf{E_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases} \xrightarrow{\text{ReLU}} \begin{cases} x_{i,j} \mapsto \begin{cases} [\mathbf{E_{i,j}}, \mathbf{E_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases} & 0 \le a \\ x_{i,j} \mapsto \begin{cases} [\frac{\mathbf{b}}{\mathbf{b}-\mathbf{a}}\mathbf{E_{i,j}}, \frac{\mathbf{b}}{\mathbf{b}-\mathbf{a}}(\mathbf{E_{i,j}} - \mathbf{a})] \\ [\mathbf{a}, \mathbf{b}] \end{cases} & a < 0 \wedge 0 < b \\ x_{i,j} \mapsto \begin{cases} [\mathbf{0}, \mathbf{0}] \\ [\mathbf{0}, \mathbf{0}] \end{cases} & b \le 0 \end{cases}$$



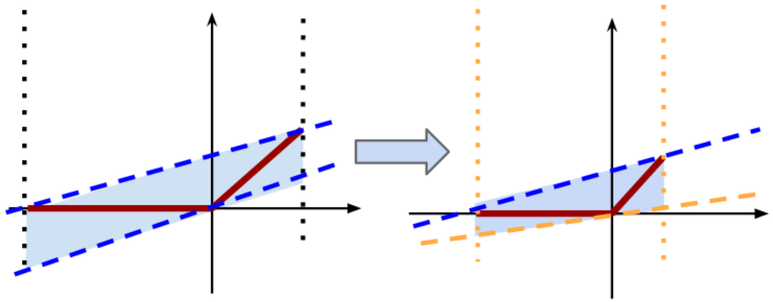
S. Wang et al. - Formal Security Analysis of Neural Networks Using Symbolic Intervals (USENIX Security 2018)

$\alpha\beta$-CROWN

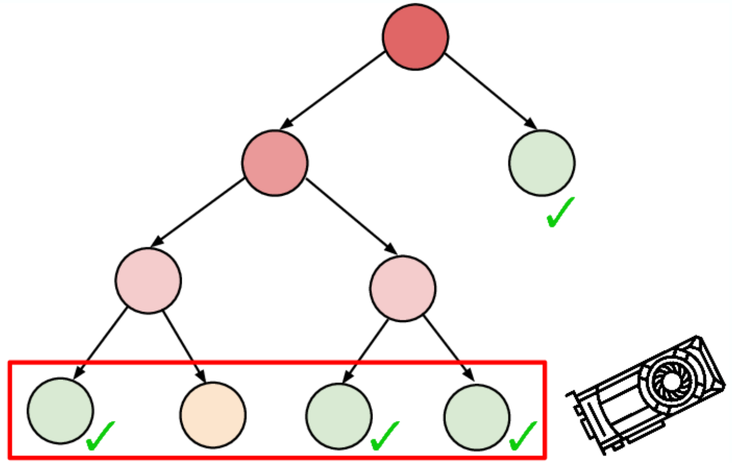
The State of the Art



Efficient bound propagation (**CROWN**)



GPU optimized relaxation (***α*-CROWN**)



Parallel branch and bound (***β*-CROWN**)

Winner of the International Verification of Neural Networks Competition since 2021

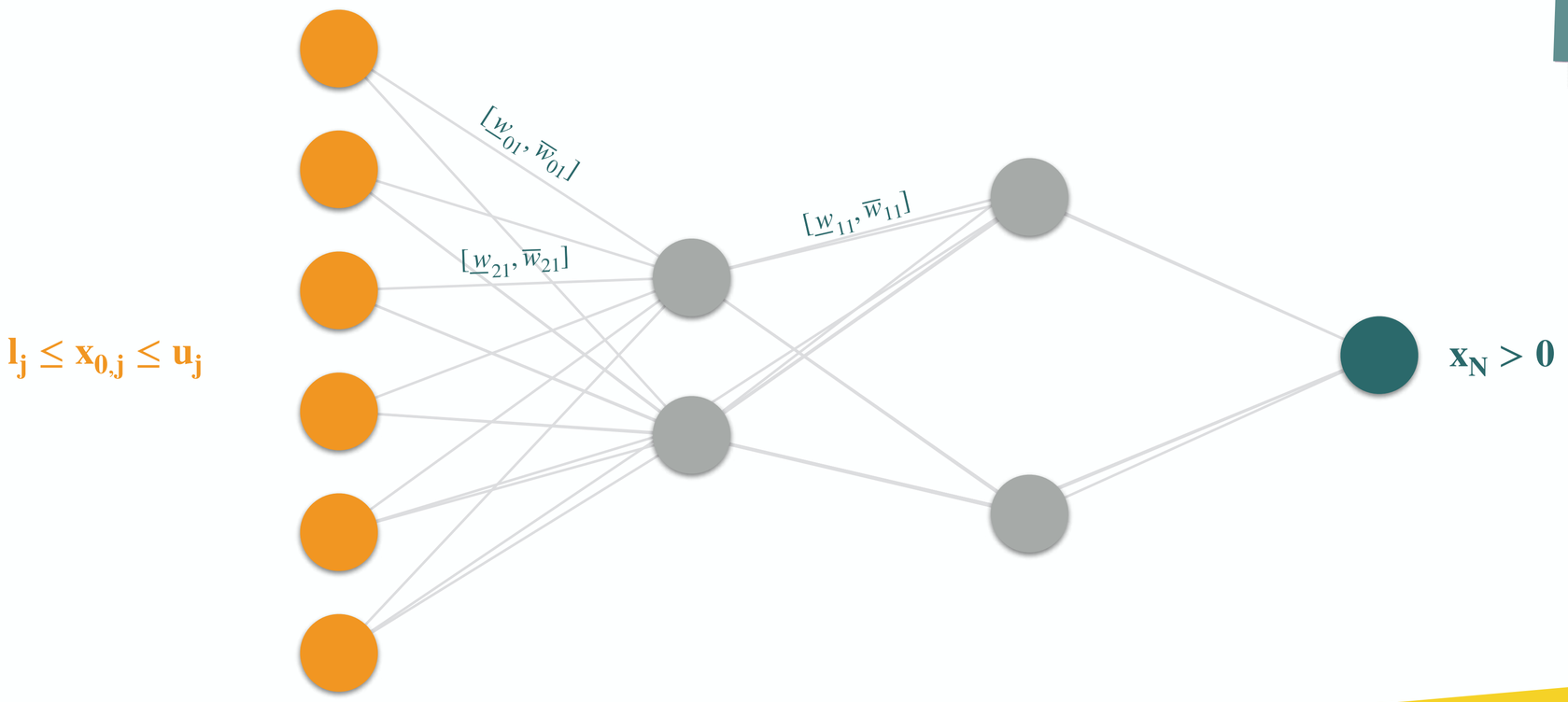
https://github.com/Verified-Intelligence/alpha-beta-CROWN

Other Abstractions

Interval Neural Networks

Related Work

Y. Y. Elboher et al. - An Abstraction-Based Framework for Neural Network Verification (CAV 2020)



merge neurons layer-wise based on partitioning strategy + replace weights with intervals

P. Prabhakar and Z. R. Afza - Abstraction based Output Range Analysis for Neural Networks (NeurIPS 2019)

Other Incomplete Methods

- **A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel.** CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks. In AAAI, 2019.
  approach focusing on convolutional neural networks

- **C.-Y. Ko, Z. Lyu, T.-W. Weng, L. Daniel, N. Wong, and D. Lin.** POPQORN: Quantifying Robustness of Recurrent Neural Networks. In ICML, 2019.
  H. Zhang, M. Shinn, A. Gupta, A. Gurfinkel, N. Le, and N. Narodytska. Verification of Recurrent Neural Networks for Cognitive Tasks via Reachability Analysis. In ECAI, 2020.
  approaches focusing on recurrent neural networks

- **G. Bonaert, D. I. Dimitrov, M. Baader, M. T. Vechev.** Fast and Precise Certification of Transformers. In PLDI, 2021.
  approach focusing on transformer models

- **D. Gopinath, H. Converse, C. S. Pasareanu, and A. Taly.** Property Inference for Deep Neural Networks. In ASE, 2019.
  an approach for inferring safety properties of neural networks

Complete Methods

Advantages

sound and **complete**

Disadvantages

soundness not typically guaranteed
with respect to **floating-point arithmetic**

do not scale to large models

often **limited** to certain
model **architectures**

Incomplete Methods

Advantages

suffer from **false positives**

Disadvantages

able to scale to large models

sound often also with respect to
floating-point arithmetic

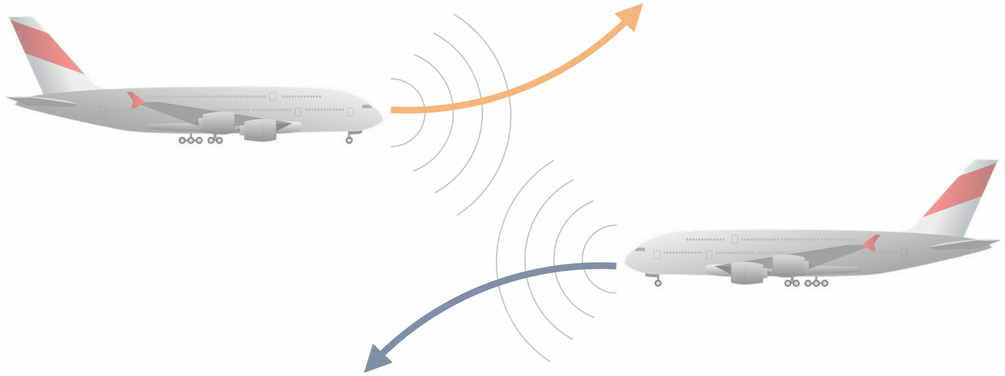
less limited to certain
model **architectures**

Stability

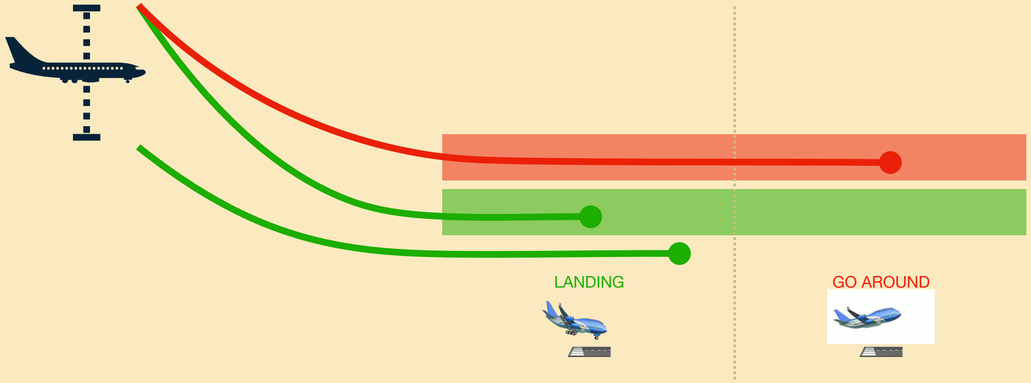
Goal G3 in [Kurd03]

Safety

Goal G4 in [Kurd03]



Hypersafety



Runway Excursions during Landing

~20% of Air Transportation Accidents*

Jeju Air Crash (December 29th, 2024)

https://www.newsweek.com/

WIKIPEDIA
The Free Encyclopedia

WIKIPEDIA

Jeju Air Flight 2216

Jeju Air Flight 2216 was a scheduled international passenger flight operated by Jeju Air from Suvarnabhumi Airport in Bangkok, Thailand, to Muan International Airport in Muan County, South Korea. On 29 December 2024, the Boeing 737-800 operating the flight was approaching Muan, when a bird strike occurred. The pilots issued a mayday alert, performed a go-around, and on the second landing attempt, the landing gear did not deploy and the airplane belly landed well beyond the normal touchdown zone. It overran the runway and crashed into a berm encasing a concrete structure that supported an antenna array for the instrument landing system.

Jeju Air Flight 2216

HL8088, the aircraft involved in the accident, pictured in 2023

Accident	
Date	29 December 2024

*https://www.airbus.com/en/newsroom/stories/2022-10-safety-innovation-5-runway-overrun-prevention-system-rops-and-runway

Regulation (EU) 2020/1159

August 5th, 2020

COMMISSION IMPLEMENTING REGULATION (EU) 2020/1159

of 5 August 2020

amending Regulations (EU) No 1321/2014 and (EU) No 2015/640 as regards the introduction of new additional airworthiness requirements

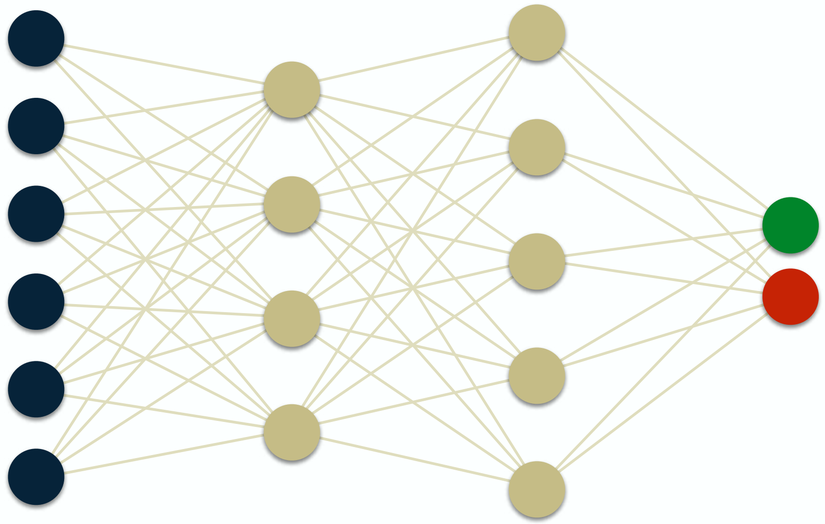
'26.205 Runway overrun awareness and alerting systems

(a) Operators of large aeroplanes used in commercial air transport shall ensure that every aeroplane for which the first individual certificate of airworthiness was issued on or after 1 January 2025, is equipped with a runway overrun awareness and alerting system.

Having regard to Regulation (EU) 2018/1139 of the European Parliament and of the Council of 4 July 2018 on common rules in the field of civil aviation and establishing a European Union Aviation Safety Agency, and amending Regulations (EC) No 2111/2005, (EC) No 1008/2008, (EU) No 996/2010, (EU) No 376/2014 and Directives 2014/30/EU

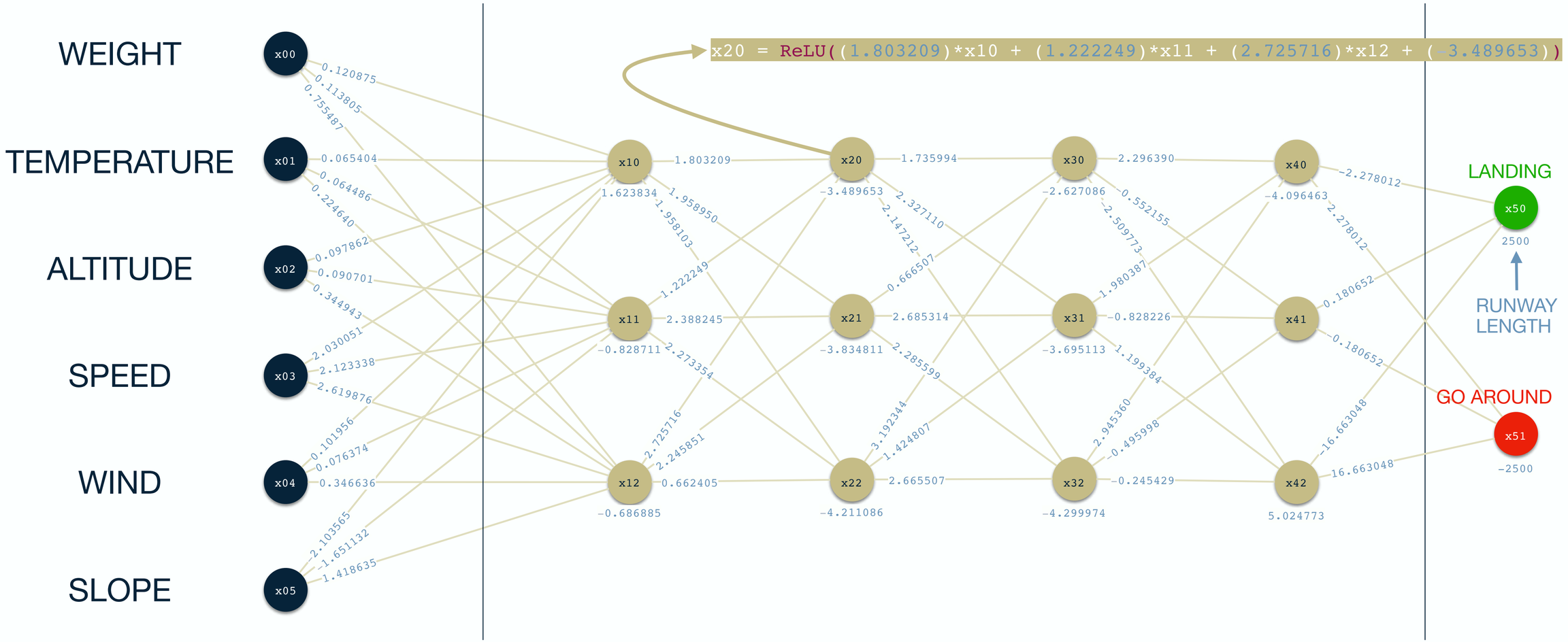
Neural Network Surrogates

Less Computing Power and Less Computing Time



Runway Overrun Warning

Toy Example

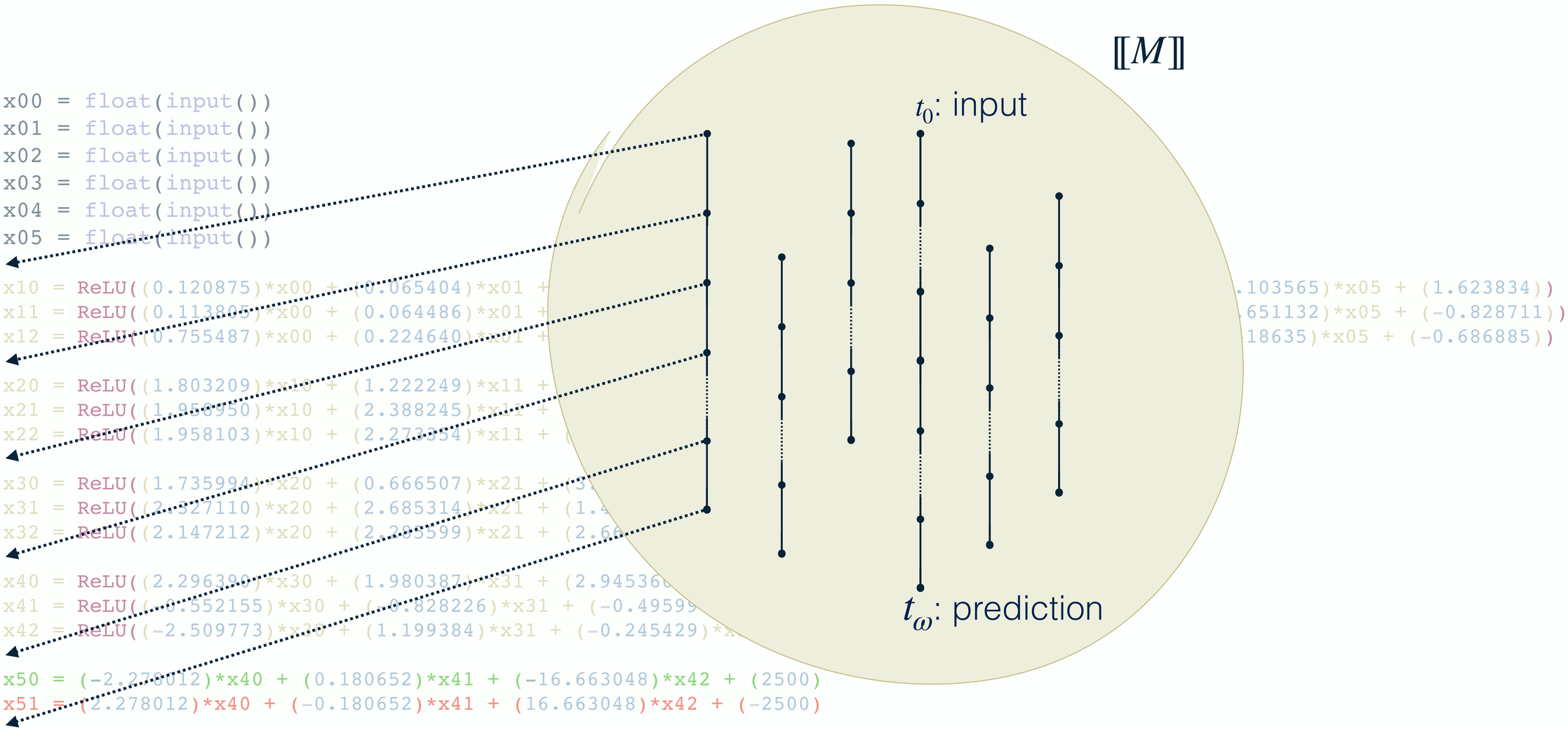


Runway Overrun Warning

Toy Example

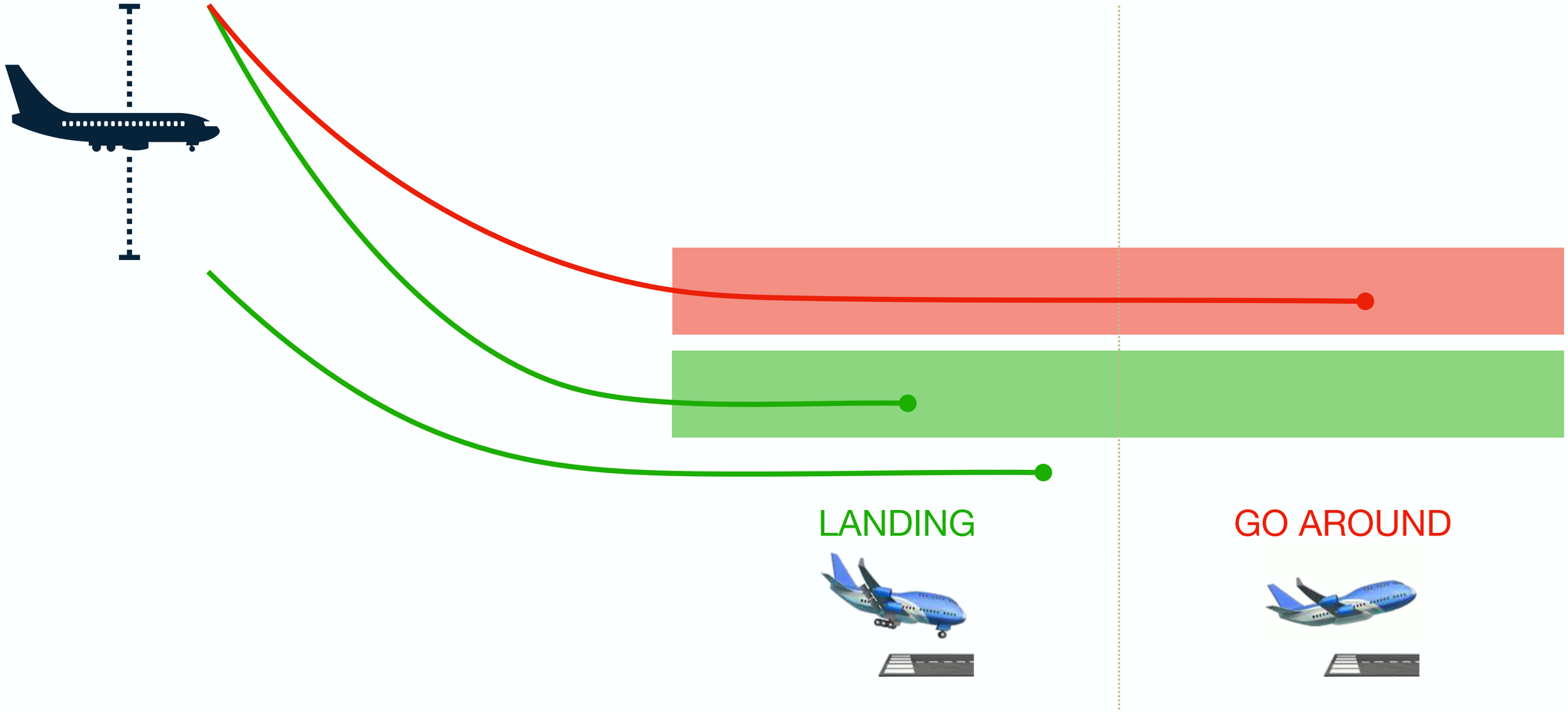
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (2500)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-2500)
```

Maximal Trace Semantics



Runway Overrun Warning

HyperSafety of Neural Network Surrogate



Hypersafety Verification

Abstract Non-Interference Properties

η: input abstraction

ρ: output abstraction

$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T \colon \eta(t_0) = \eta(t'_0) \Rightarrow \rho(t_\omega) = \rho(t'_\omega) \right\}$$

$\mathscr{H}^{\eta}_{\rho}$ is the set of all executions that **satisfy** abstract non-interference with respect to η and ρ

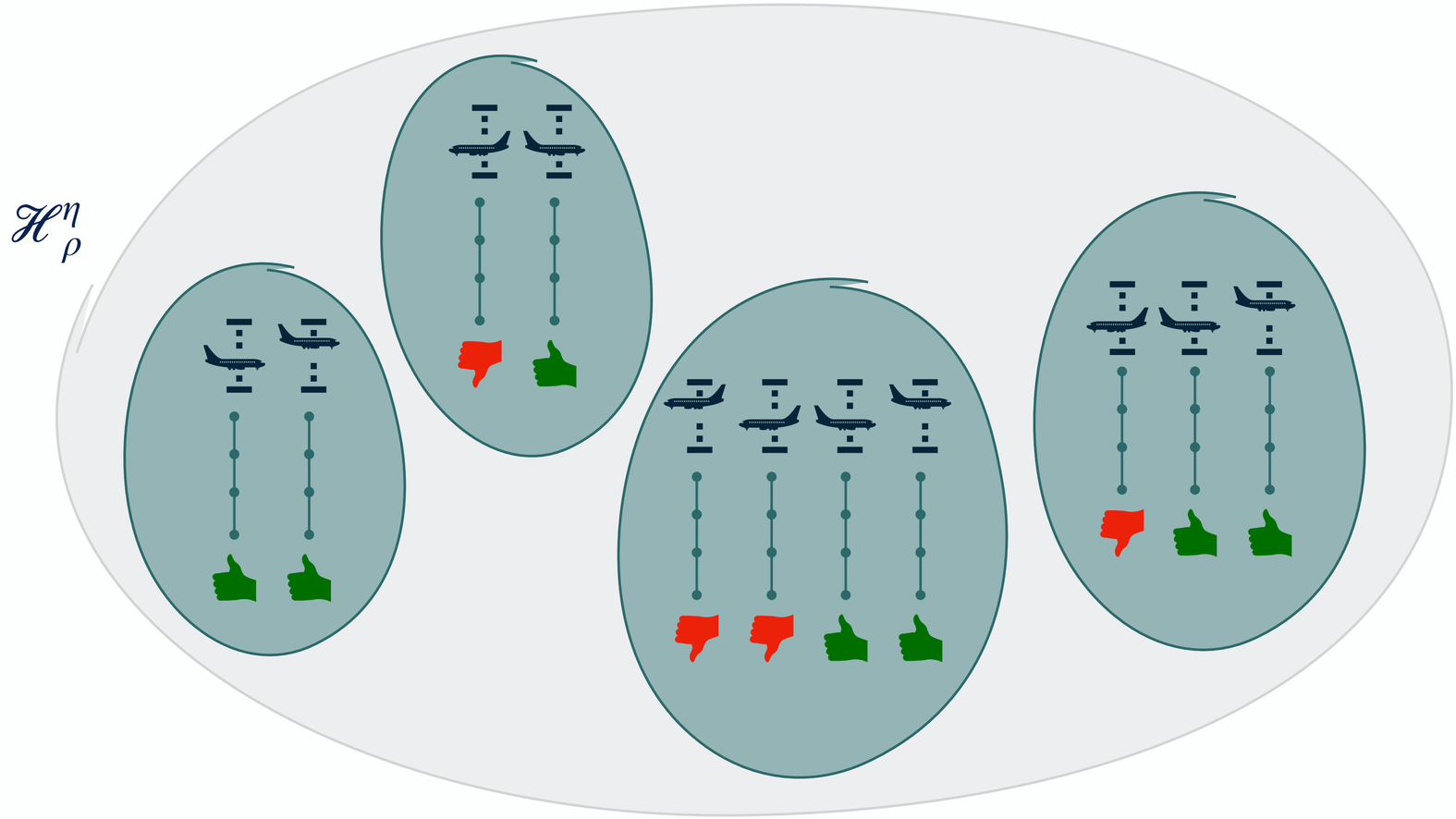
Theorem

$$M \vDash \mathscr{H}^{\eta}_{\rho} \Leftrightarrow [\![M]\!] \in \mathscr{H}^{\eta}_{\rho} \Leftrightarrow \{[\![M]\!]\} \subseteq \mathscr{H}^{\eta}_{\rho}$$

Giacobazzi and Mastroeni. Abstract Non-Interference: A Unifying Framework for Weakening Information-Flow. In TOPS, 2018.

Abstract Non-Interference

Subset-Closed Property (*)



(*) Machine Learning Models are Deterministic

Hypersafety Verification

Abstract Non-Interference Properties

η: input abstraction

ρ: output abstraction

$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T\colon \eta(t_0) = \eta(t'_0) \Rightarrow \rho(t_\omega) = \rho(t'_\omega) \right\}$$

$\mathscr{H}^{\eta}_{\rho}$ is the set of all executions that **satisfy** abstract non-interference with respect to η and ρ

Theorem

$$M \vDash \mathscr{H}^{\eta}_{\rho} \Leftrightarrow [\![M]\!] \in \mathscr{H}^{\eta}_{\rho} \Leftrightarrow \{[\![M]\!]\} \subseteq \mathscr{H}^{\eta}_{\rho}$$

Corollary

$$M \vDash \mathscr{H}^{\eta}_{\rho} \Leftarrow [\![M]\!] \subseteq [\![M]\!]^{\natural} \subseteq \mathscr{H}^{\eta}_{\rho}$$

Giacobazzi and Mastroeni. Abstract Non-Interference: A Unifying Framework for Weakening Information-Flow. In TOPS, 2018.

Abstract Non-Interference Verification

Example

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

ALTITUDE

η:

$\eta(x00) = x00$
$\eta(x01) = x01$
$\eta(x02) = \top$
$\eta(x03) = x03$
$\eta(x04) = x04$
$\eta(x05) = x05$

“**the risk of a runway overrun does not change when only varying the altitude** at which it is measured (in the expected range) and nothing else”

ρ:

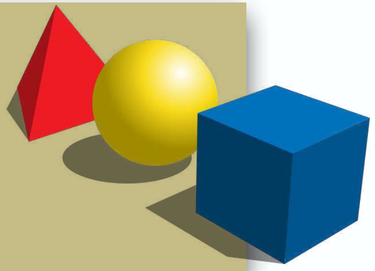
$\rho(x50) = 1$ if $x50 > x51$ else 0
$\rho(x51) = 1$ if $x51 > x50$ else 0

Abstract Interpretation

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Abstract Interpretation

3-Step Recipe

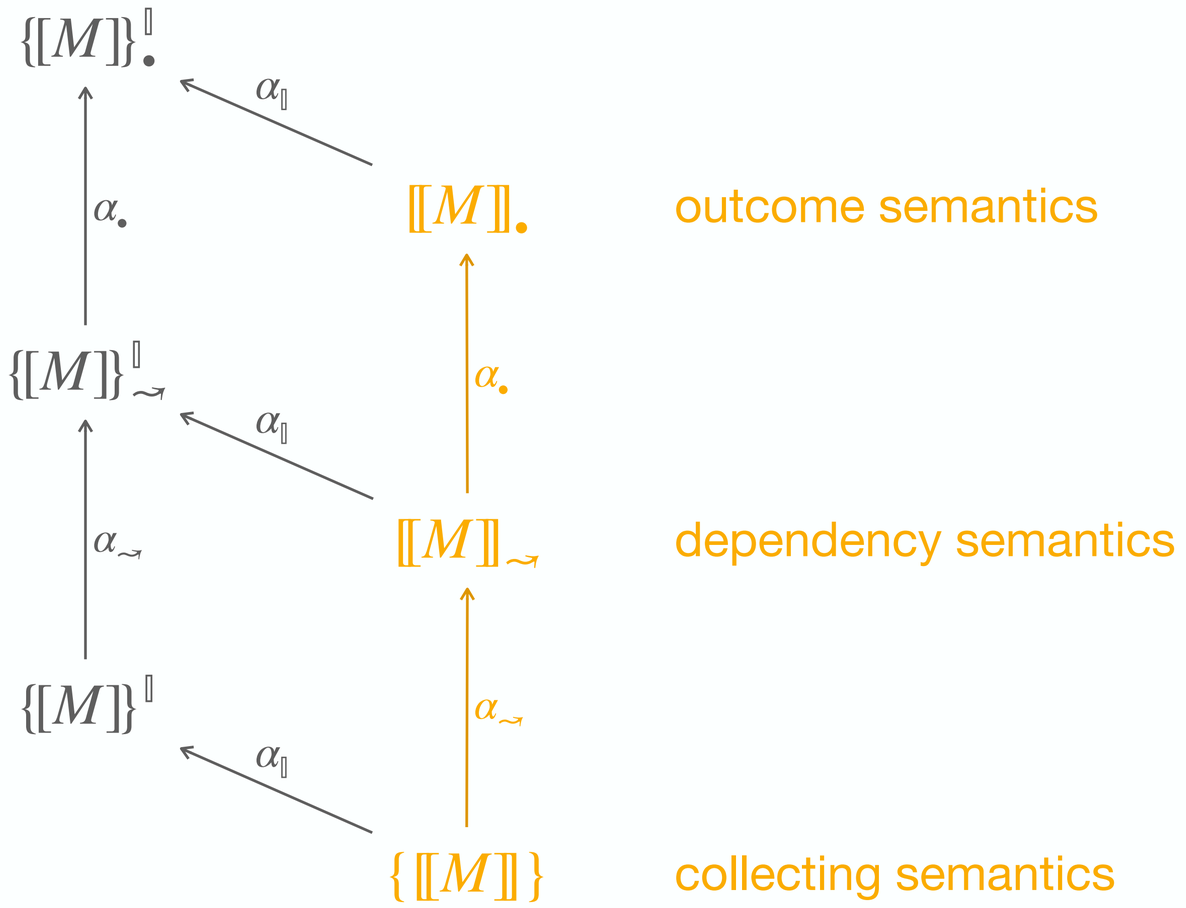
practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

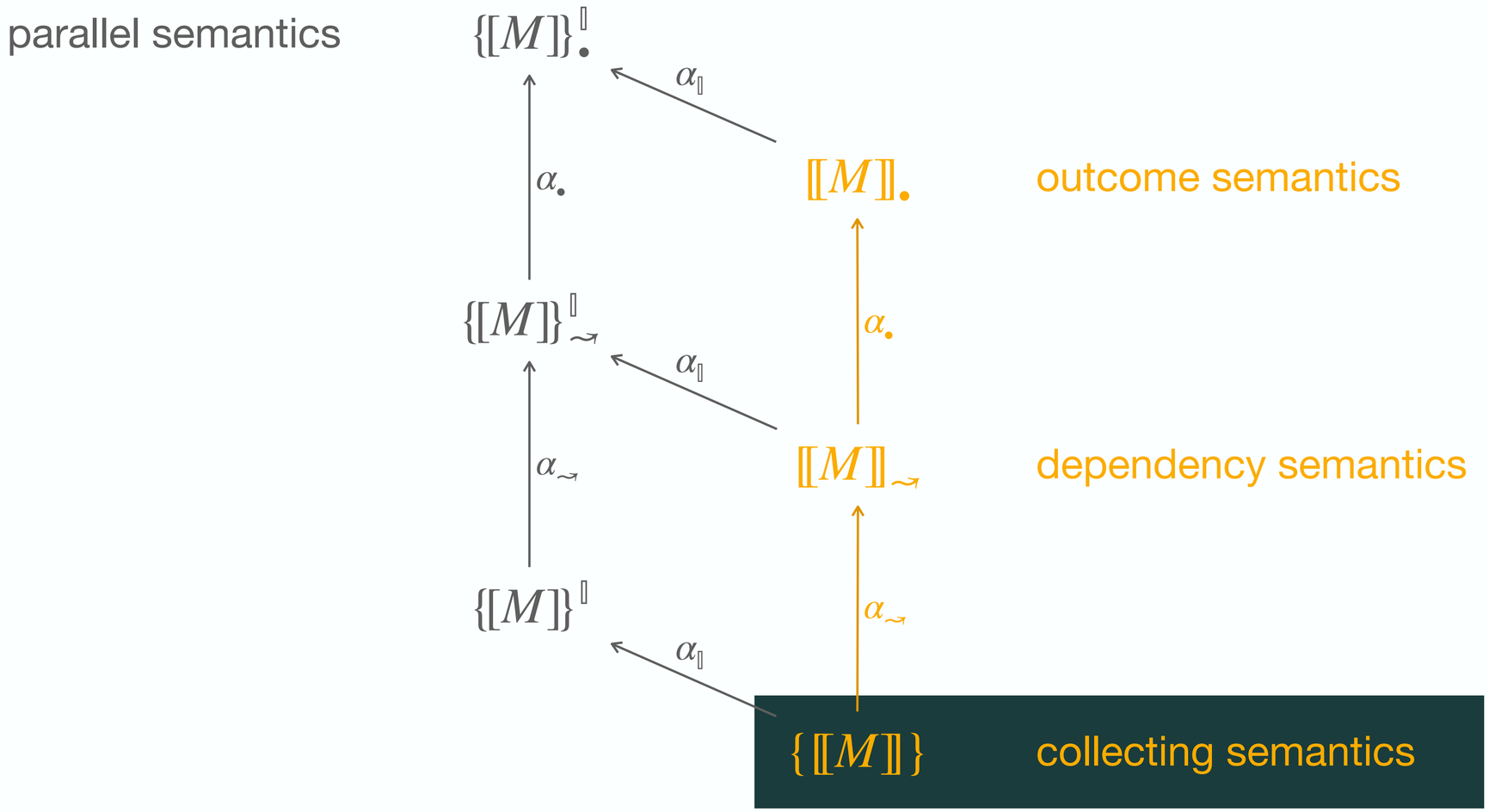
concrete semantics
mathematical models of the program behavior

Hierarchy of Semantics

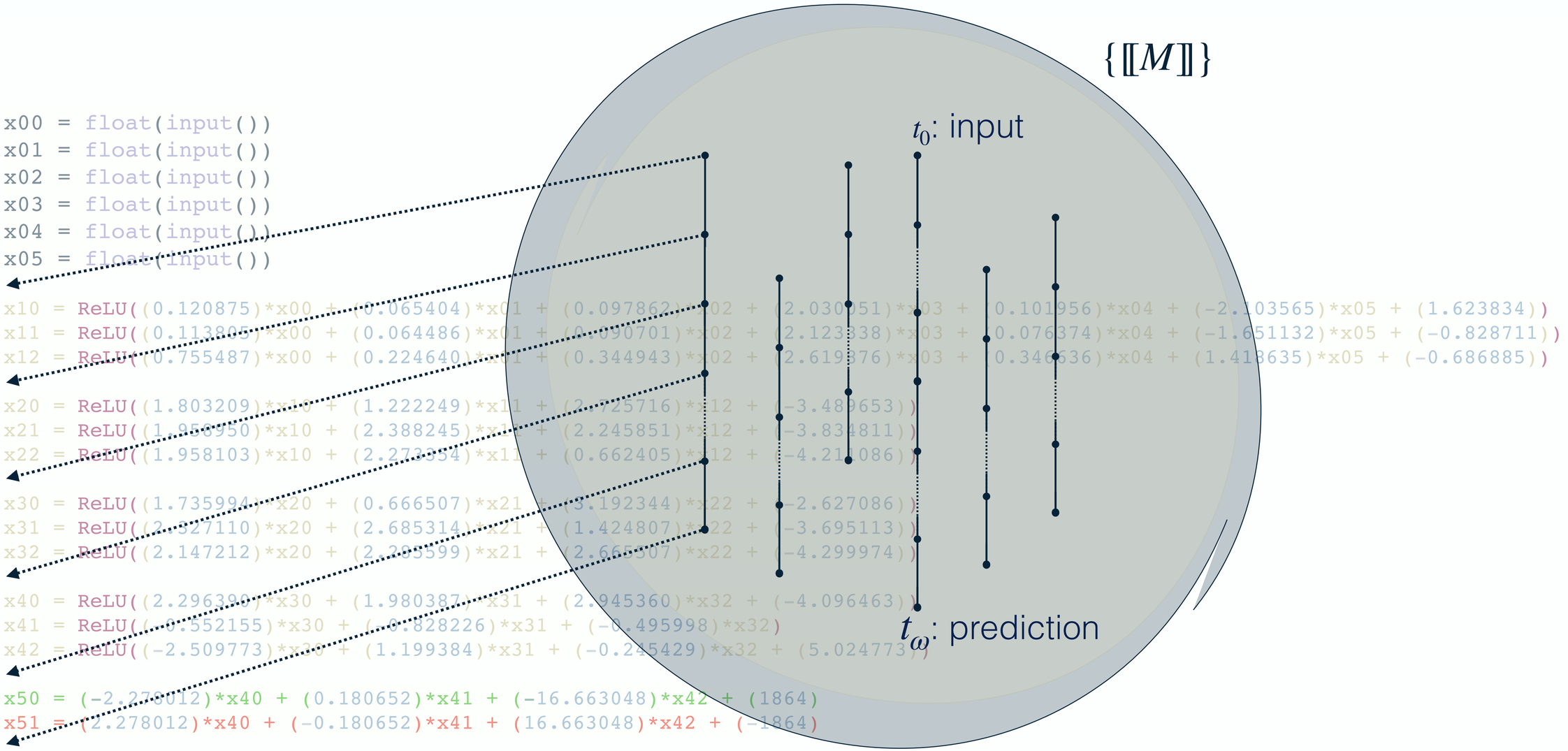
parallel semantics



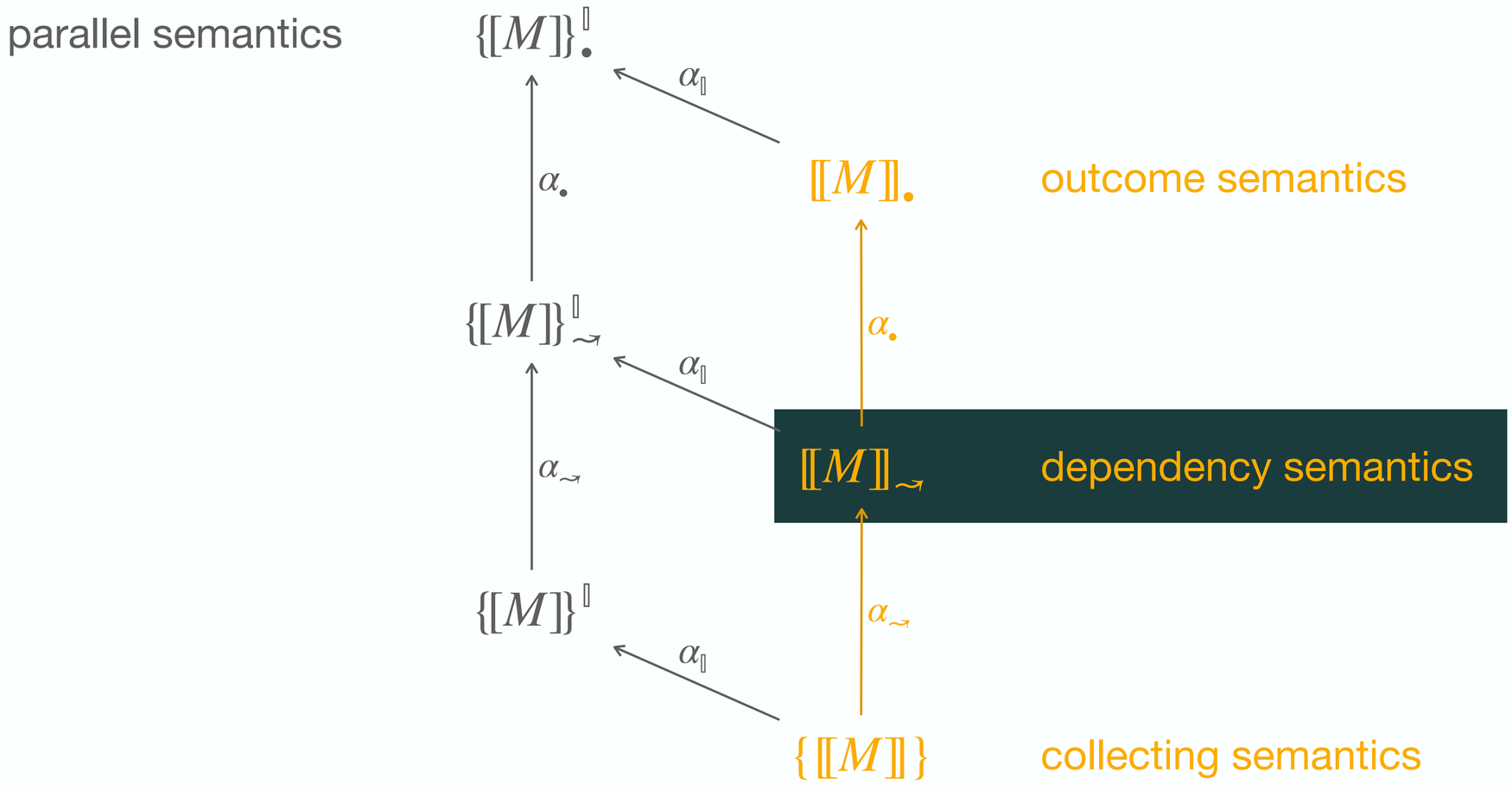
Hierarchy of Semantics



Collecting Semantics

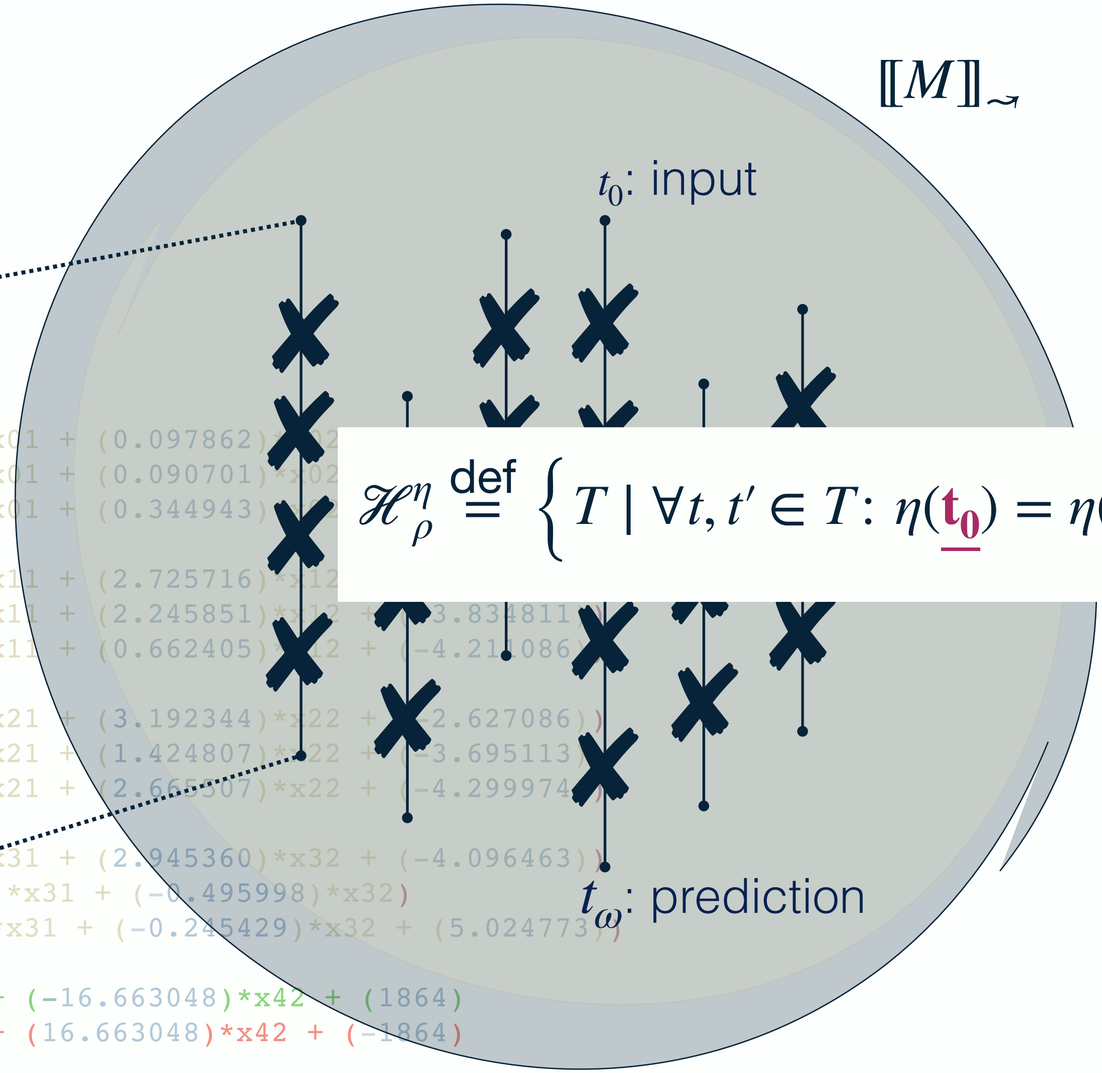


Hierarchy of Semantics



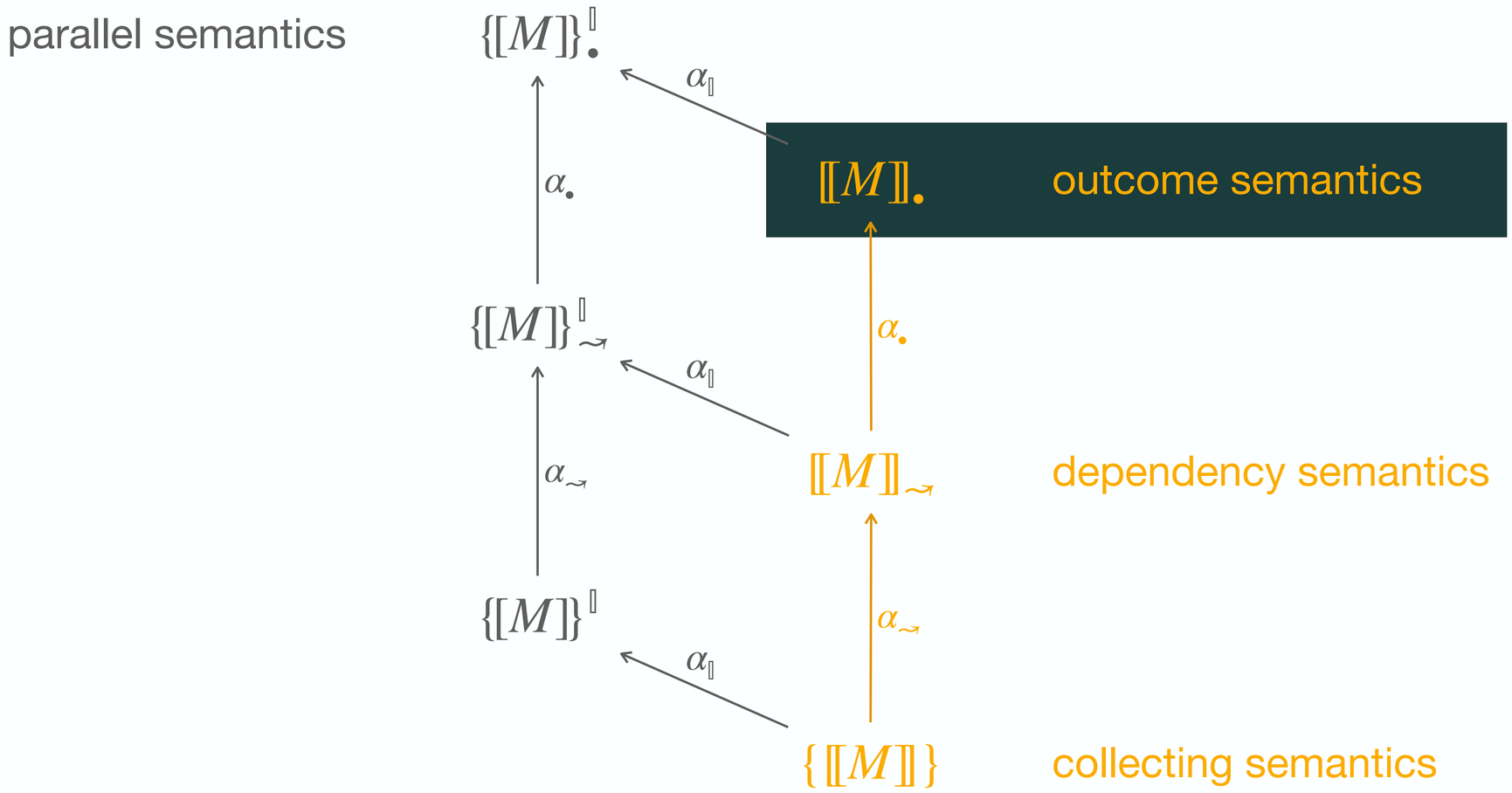
Dependency Semantics

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.214086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665307)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



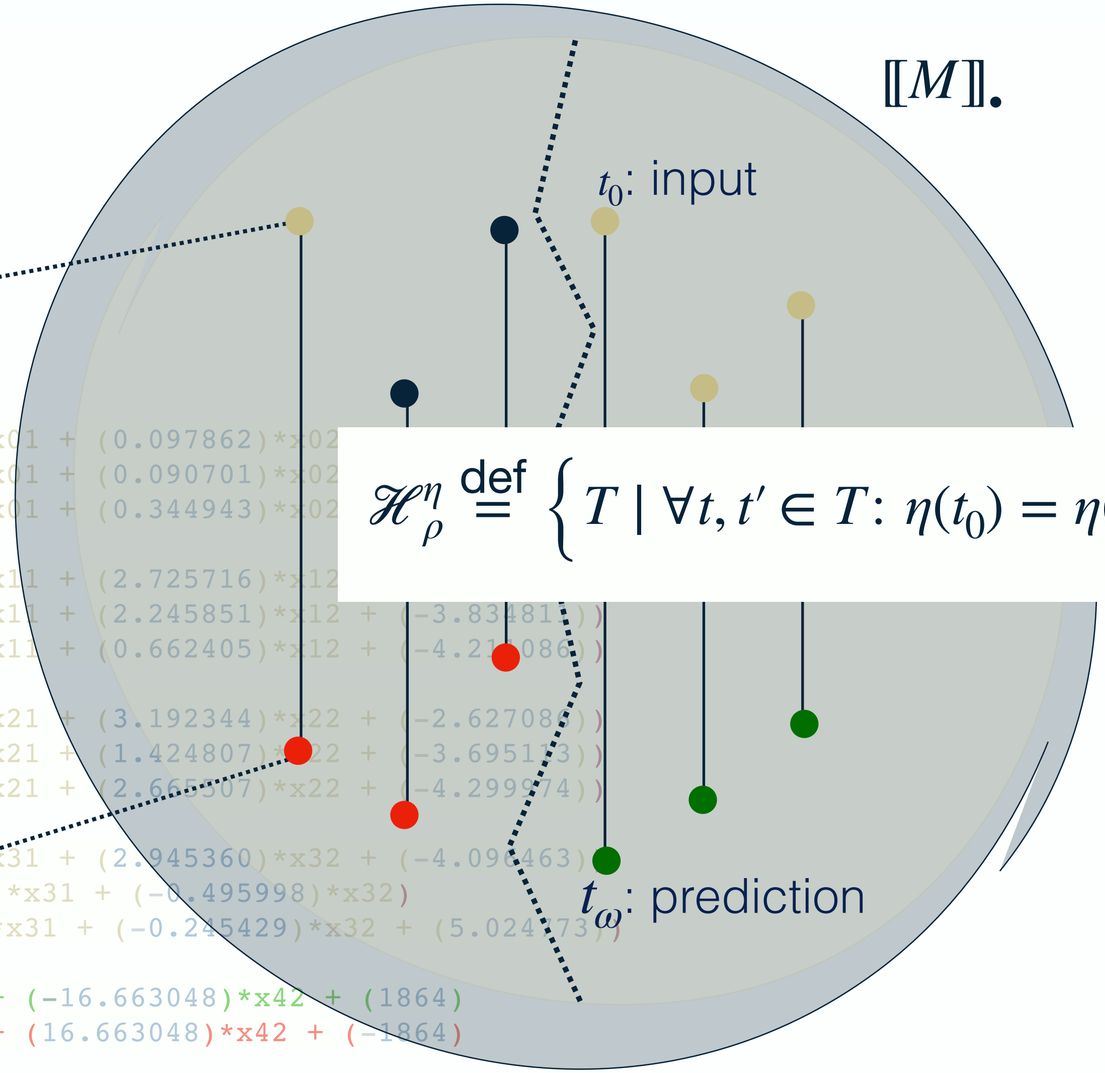
$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T\colon \eta(\mathbf{t_0}) = \eta(\mathbf{t'_0}) \Rightarrow \rho(\mathbf{t}_\omega) = \rho(\mathbf{t}'_\omega) \right\}$$

Hierarchy of Semantics



Outcome Semantics

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627080))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695173))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665307)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024473))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T\colon \eta(t_0) = \eta(t'_0) \Rightarrow \underline{\rho}(t_\omega) = \underline{\rho}(t'_\omega) \right\}$$

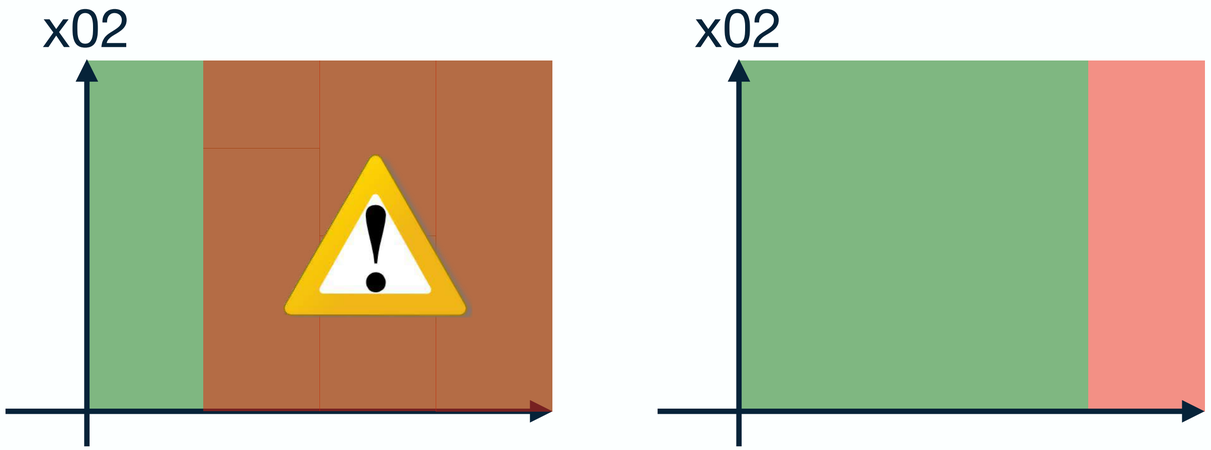
Hypersafety Verification

Abstract Non-Interference Properties

$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T\colon \eta(t_0) = \eta(t'_0) \Rightarrow \rho(t_\omega) = \rho(t'_\omega) \right\}$$

Lemma

$$M \vDash \mathscr{H}^{\eta}_{\rho} \Leftrightarrow \forall A, B \in [\![M]\!]_{\bullet}\colon \rho(A_\omega) \sqcap \rho(B_\omega) = \bot \Rightarrow \eta(A_0) \sqcap \eta(B_0) = \bot$$



Giacobazzi and Mastroeni. Abstract Non-Interference: A Unifying Framework for Weakening Information-Flow. In TOPS, 2018.

Abstract Interpretation

3-Step Recipe

practical tools
targeting specific programs

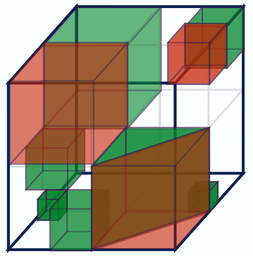
abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

Hypersafety Verification

Naïve Backward Analysis

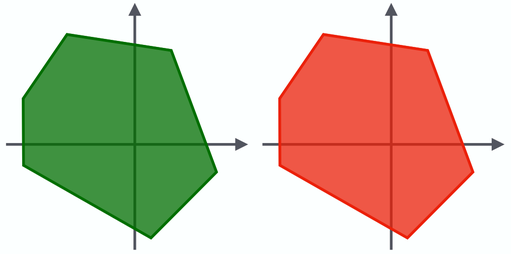
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① check for **disjunction** in corresponding **input partitions**:
disjoint → ✓ **safe**
otherwise → 🚨 **alarm**

② proceed **backwards** through the model layers

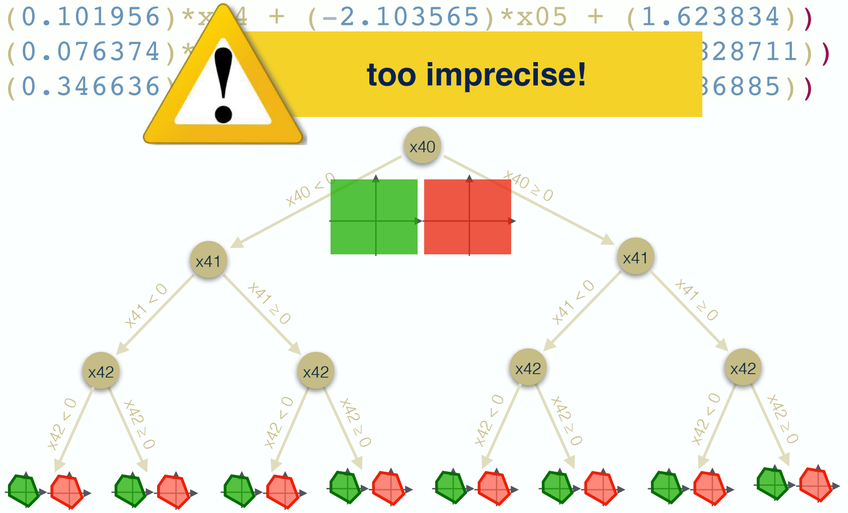
① start from an **abstraction** for each possible classification outcome



Hypersafety Verification

Naïve Backward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-2.328711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (-2.836885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



$(-4.556024) * x40 + (0.361304) * x41 + (-33.326096) * x42 + (3728) > 0$

$(4.556024) * x40 + (33.326096) * x42 - 3728 > 0$

Hypersafety Verification

Naïve Backward Analysis with Disjunctive Completion

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

too many disjunctions!

4096

4096 `x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))`

`x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))`

512 `x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))`

512 `x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))`

`x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))`

64 `x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))`

64 `x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))`

`x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))`

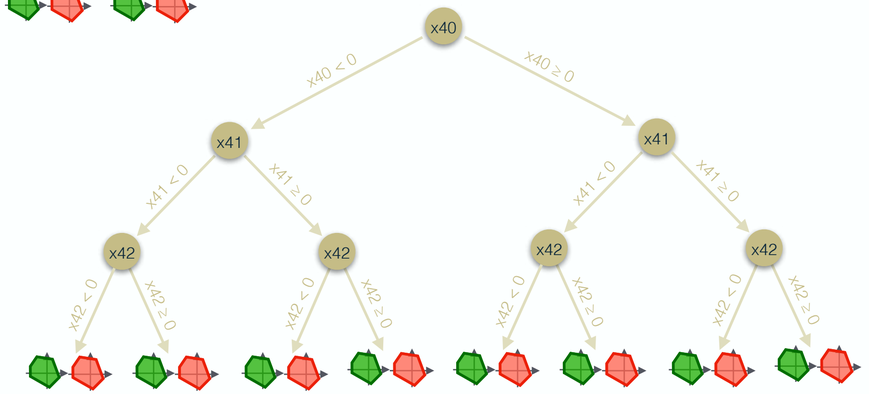
8 `x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))`

8 `x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))`

`x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)`

`x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))`

```
x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



$(-4.556024) * x40 + (0.361304) * x41 + (-33.326096) * x42 + (3728) > 0$

$(4.556024) * x40 + (33.326096) * x42 - 3728 > 0$

Abstract Interpretation

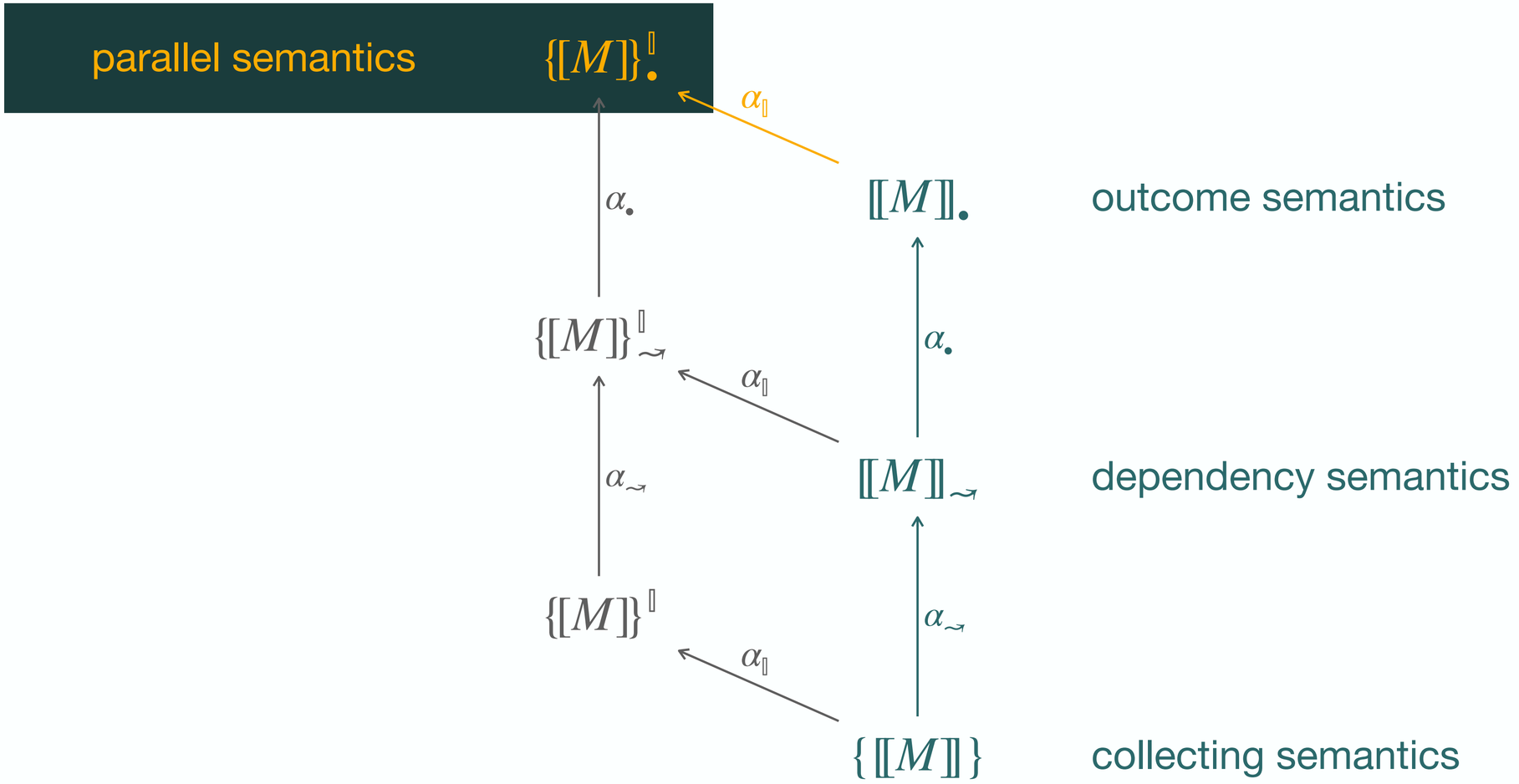
3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

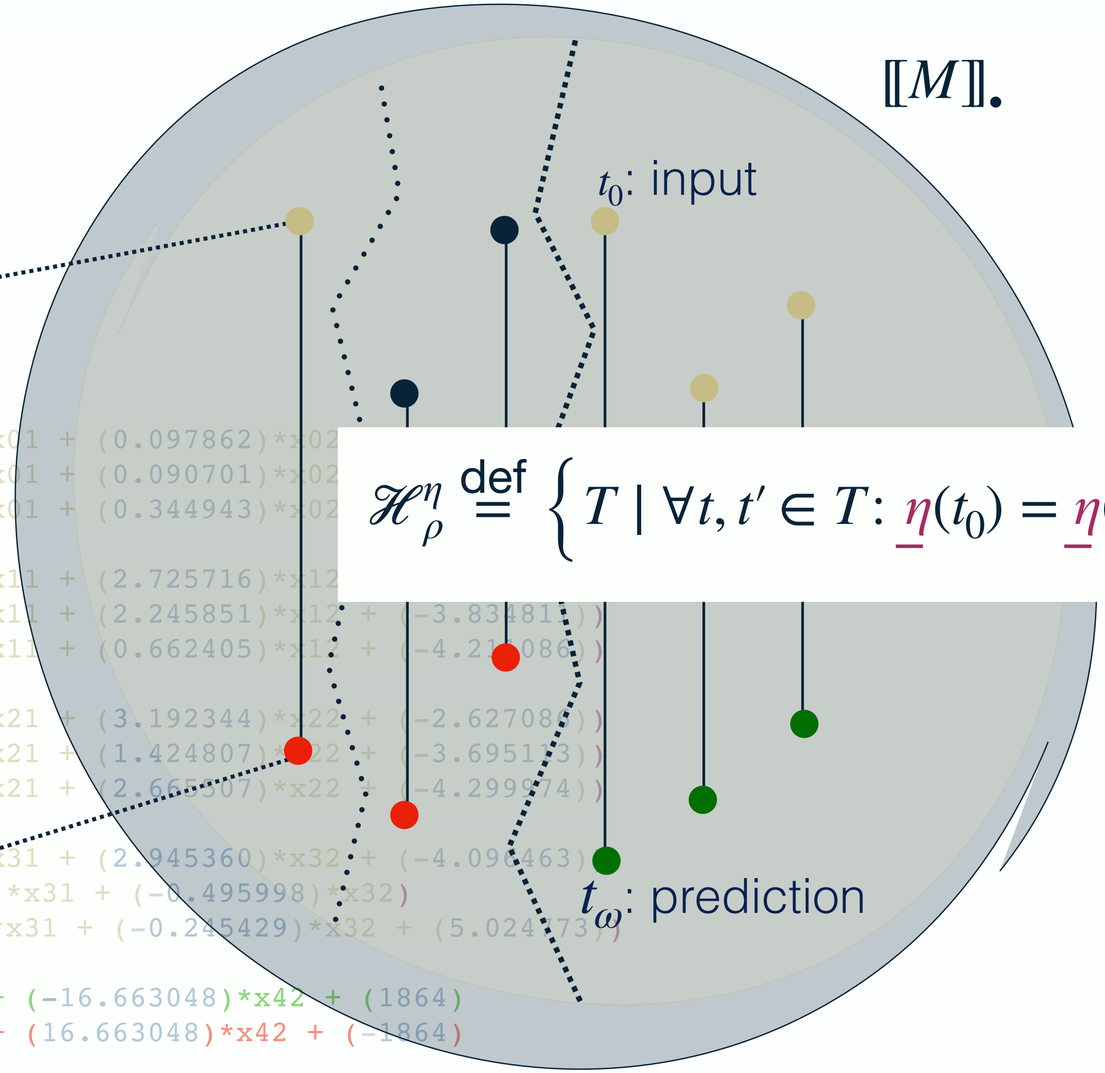
concrete semantics
mathematical models of the program behavior

Hierarchy of Semantics



Parallel Semantics

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627080))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695173))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665307)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024473))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T\colon \underline{\eta}(t_0) = \underline{\eta}(t'_0) \Rightarrow \rho(t_\omega) = \rho(t'_\omega) \right\}$$

Hypersafety Verification

Abstract Non-Interference Properties

$$\mathscr{H}^{\eta}_{\rho} \stackrel{\text{def}}{=} \left\{ T \mid \forall t, t' \in T \colon \eta(t_0) = \eta(t'_0) \Rightarrow \rho(t_\omega) = \rho(t'_\omega) \right\}$$

Lemma

$$M \vDash \mathscr{H}^{\eta}_{\rho} \Leftrightarrow \forall I \in \mathbb{I} \colon \forall A, B \in [\![M]\!]^{\mathbb{I}}_{\bullet} \colon \rho(A^I_\omega) \sqcap \rho(B^I_\omega) = \bot \Rightarrow \eta(A^I_0) \sqcap \eta(B^I_0) = \bot$$

Giacobazzi and Mastroeni. Abstract Non-Interference: A Unifying Framework for Weakening Information-Flow. In TOPS, 2018.

Abstract Interpretation

3-Step Recipe

practical tools
targeting specific programs

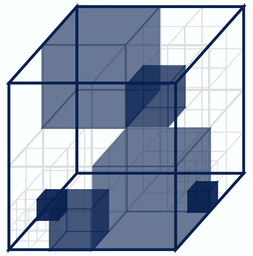
abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

Hypersafety Verification [Urban20]

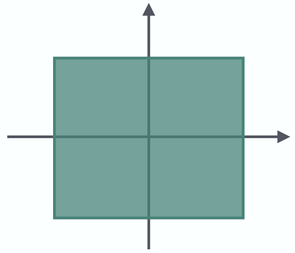
Static Forward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① start from a **partition** of the input space

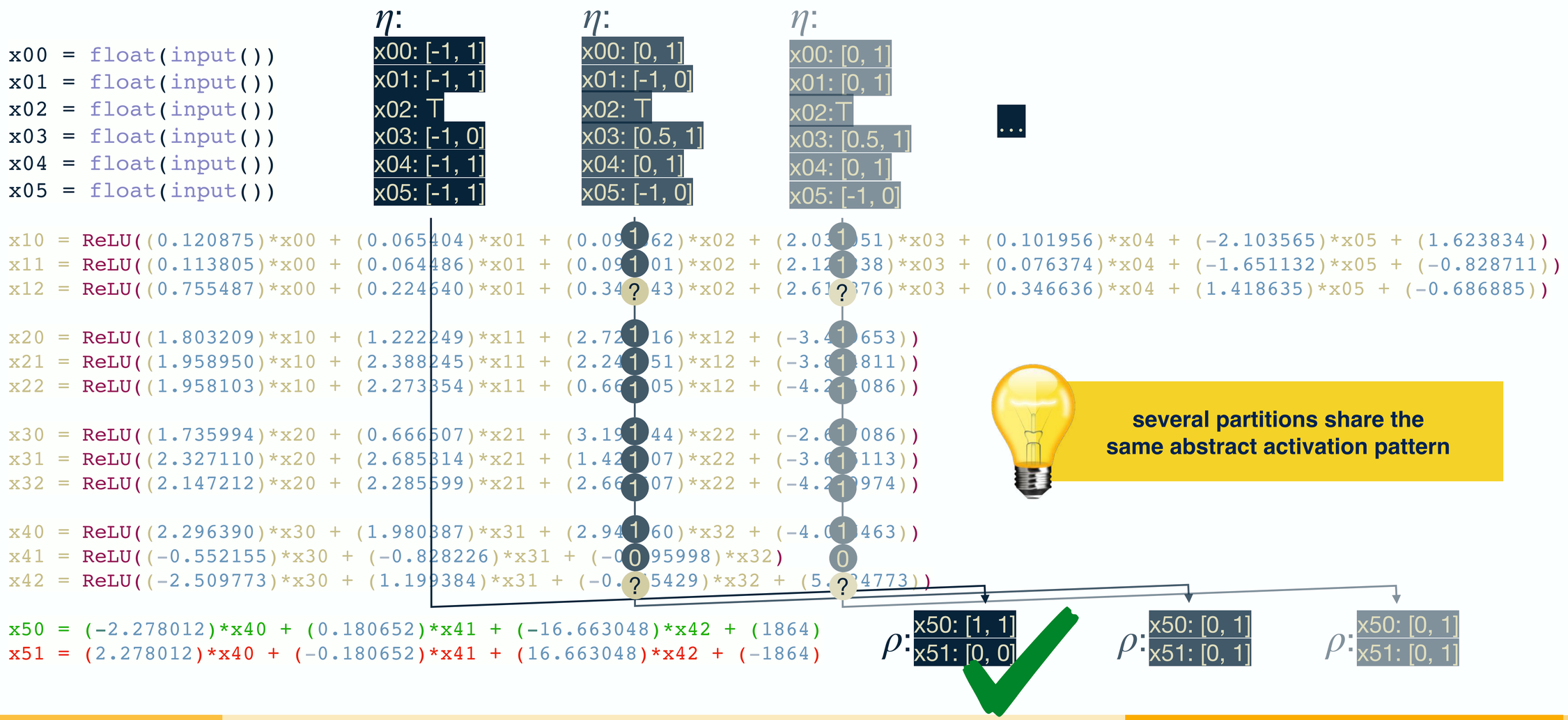
② proceed **forwards** **in parallel** from all partitions



③ check output for:
- **unique classification outcome** → ✓ **safe**
- **abstract activation pattern**

Static Forward Analysis

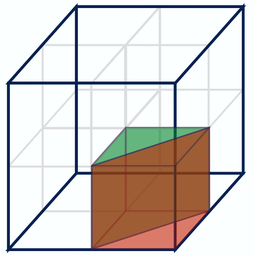
Symbolic & DeepPoly Product Abstraction



Hypersafety Verification [Urban20]

Static Backward Analysis

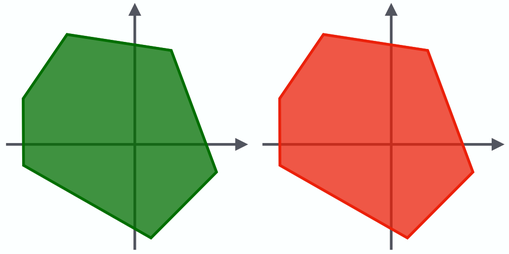
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① check for **disjunction** in corresponding **input partitions**:
disjoint → ✓ **safe**
otherwise → 🚨 **alarm**

② proceed **backwards** in parallel **for each abstract activation pattern**

① start from an **abstraction** for each possible classification outcome



Static Backward Analysis

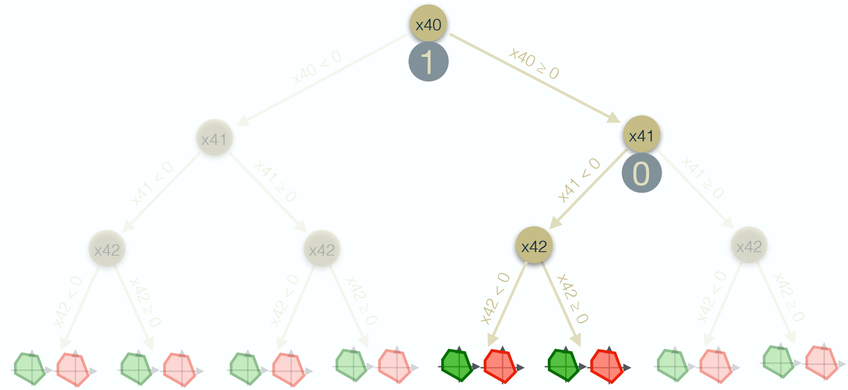
Symbolic & DeepPoly Product Abstraction

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

η:
x00: [0, 1]
x01: [-1, 0]
x02: T
x03: [0.5, 1]
x04: [0, 1]
x05: [-1, 0]

η:
x00: [0, 1]
x01: [0, 1]
x02:T
x03: [0.5, 1]
x04: [0, 1]
x05: [-1, 0]

```
1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
1 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
? x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

1 x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
1 x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
1 x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

1 x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
1 x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
1 x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

1 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
? x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



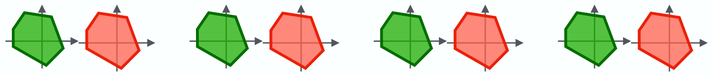
$(-4.556024) * x40 + (0.361304) * x41 + (-33.326096) * x42 + (3728) > 0$

$(4.556024) * x40 + (33.326096) * x42 - 3728 > 0$

Static Backward Analysis

Symbolic & DeepPoly Product Abstraction

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```



η:

x00: [0, 1]
x01: [-1, 0]
x02: T
x03: [0.5, 1]
x04: [0, 1]
x05: [-1, 0]

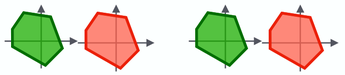
η:

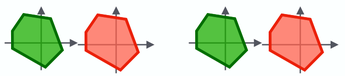
x00: [0, 1]
x01: [0, 1]
x02:T
x03: [0.5, 1]
x04: [0, 1]
x05: [-1, 0]

counterexample

x00: 1	x00: 1
x01: 1	x01: 1
x02: -1	x02: 1
x03: 1	x03: 1
x04: 1	x04: 1
x05: -1	x05: -1

```
1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
1 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
? x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

1 x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
1 x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
1 x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

⋮

1 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
? x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```





$(-4.556024) * x40 + (0.361304) * x41 + (-33.326096) * x42 + (3728) > 0$

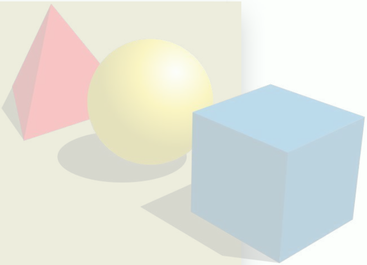
$(4.556024) * x40 + (33.326096) * x42 - 3728 > 0$

Abstract Interpretation

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



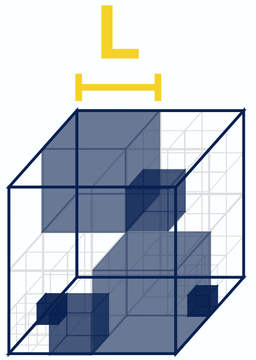
concrete semantics
mathematical models of the program behavior

Hypersafety Verification [Urban20]

Static Forward Analysis

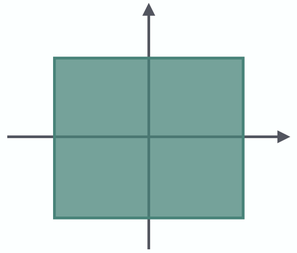
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

```
1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
1 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
? x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

? x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
? x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
? x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

? x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
1 x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
0 x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

1 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
0 x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① **iteratively** partition the input space

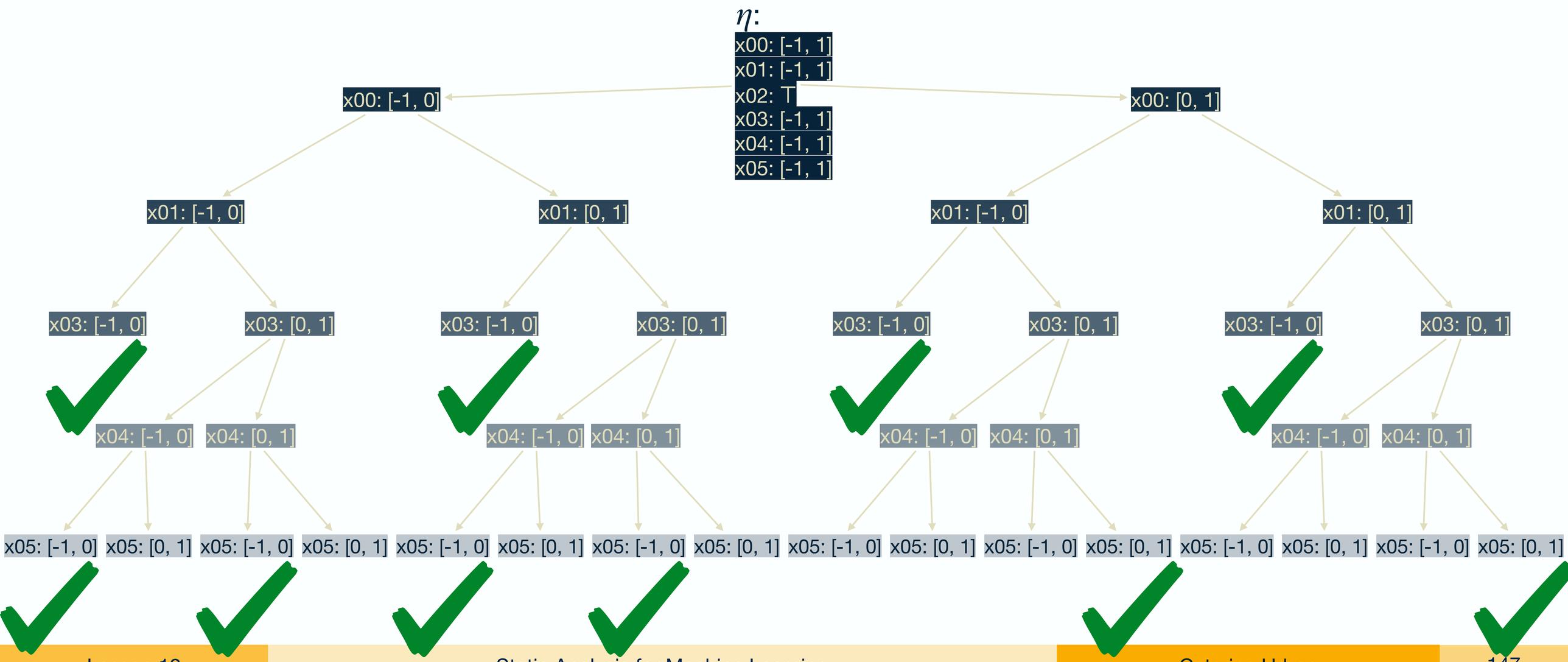
② proceed **forwards** **in parallel** from all partitions



③ check output for:
- **unique classification outcome** → ✓ **safe**
- **abstract activation pattern** U

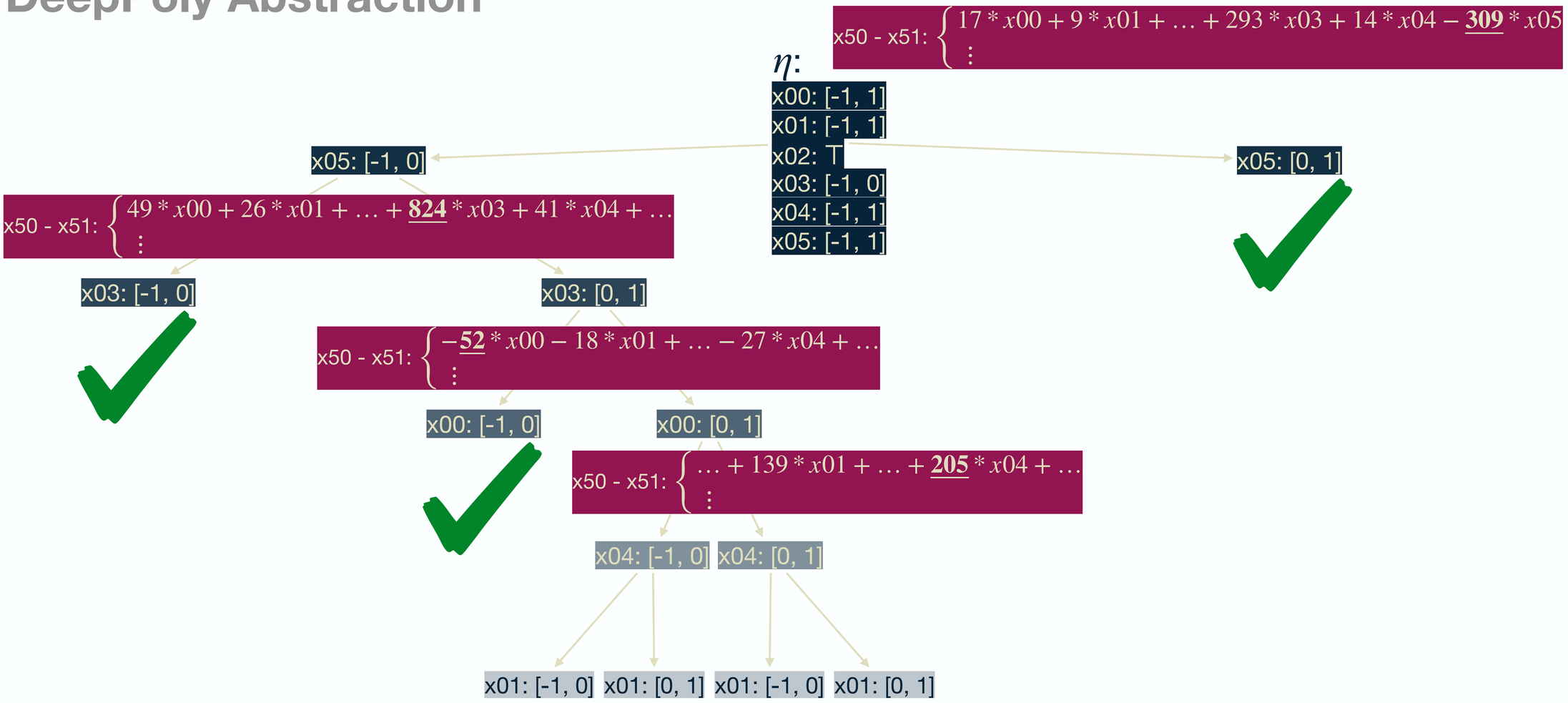
Partitioning Strategies: Interval Range

DeepPoly Abstraction



Partitioning Strategies: ReCIPH

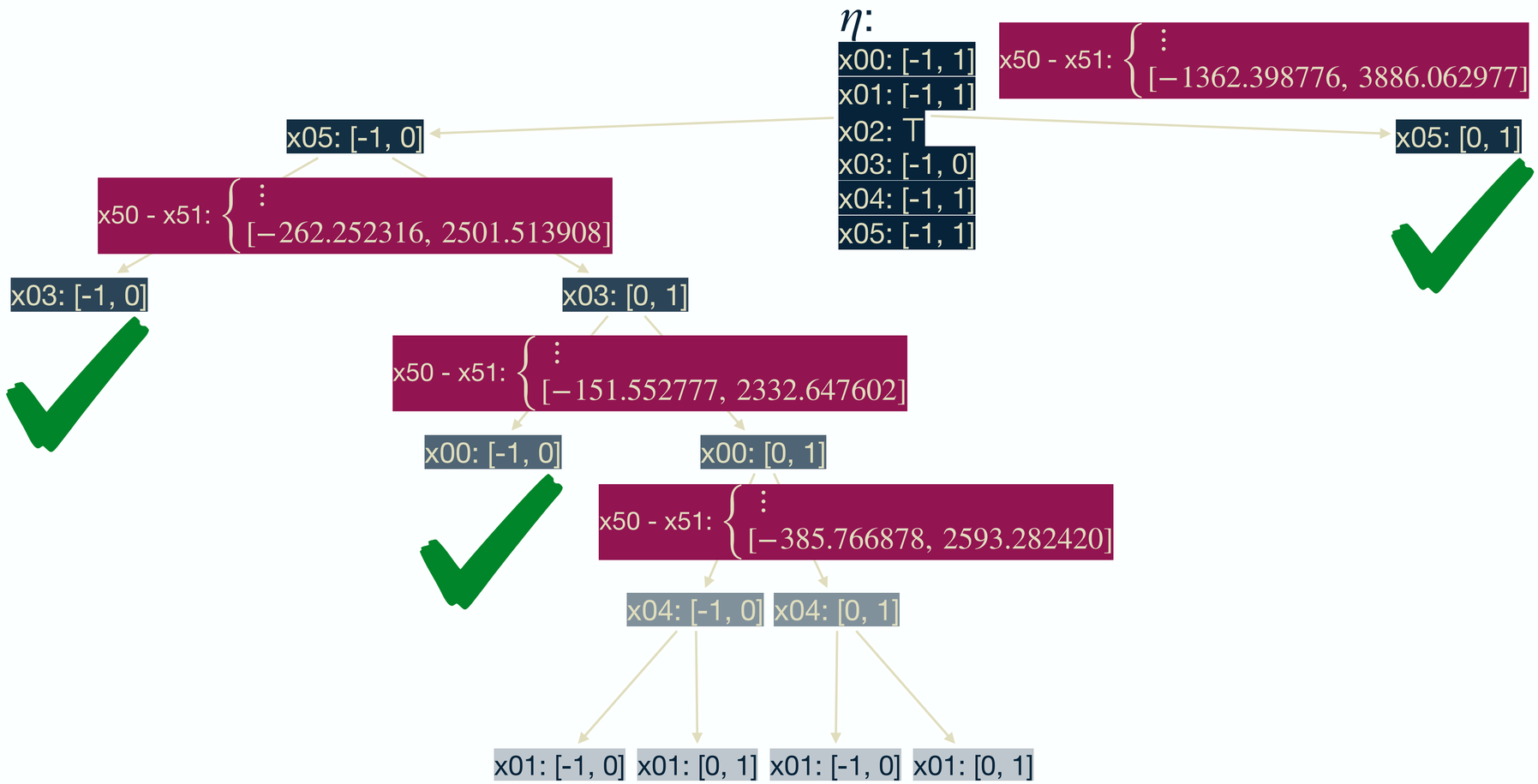
DeepPoly Abstraction



Durand, Lemesle, Chihani, CU, and Terrier. ReCIPH: Relational Coefficients for Input Partitioning Heuristic. In WFVML, 2022

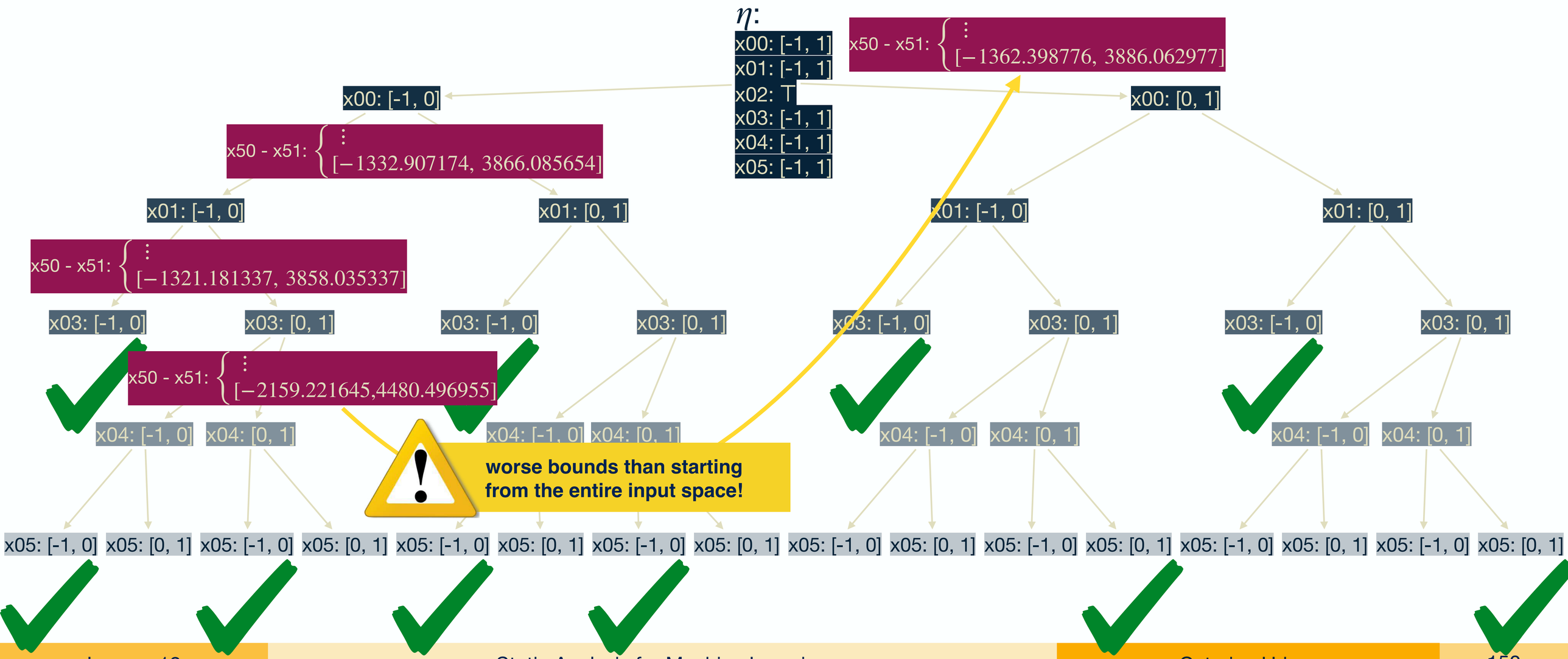
Input Refinement ⇏ Output Refinement

DeepPoly Abstraction with ReCIPH Partitioning



Input Refinement ⇏ Output Refinement

DeepPoly Abstraction with Input Range Partitioning



Scalability-vs-Precision Tradeoff

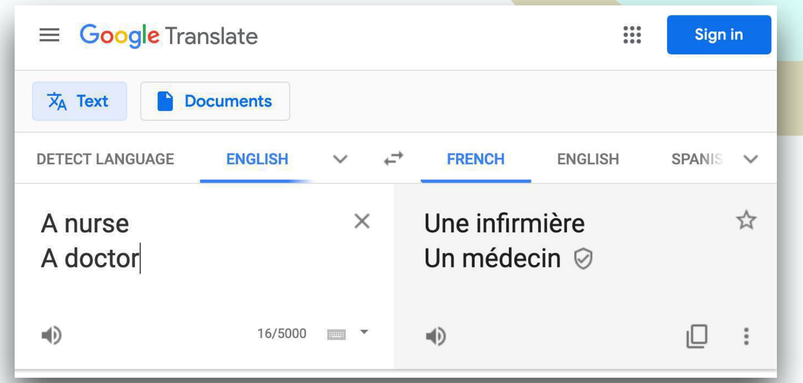
Analyzed Input Space Percentage

L	U	Boxes	Symbolic	DeepPoly		Product	
				Input Range Partitioning	ReCIPH	Input Range Partitioning	ReCIPH
1	2	46,9 %	46,9 %	68,8 %	87,5 %	90,6 %	90,6 %
	6	46,9 %	46,9 %	68,8 %	87,5 %	90,6 %	90,6 %
0.5	2	76,9 %	89,2 %	100,0 %	100,0 %	100,0 %	100,0 %
	6	84,4 %	89,9 %	100,0 %	100,0 %	100,0 %	100,0 %

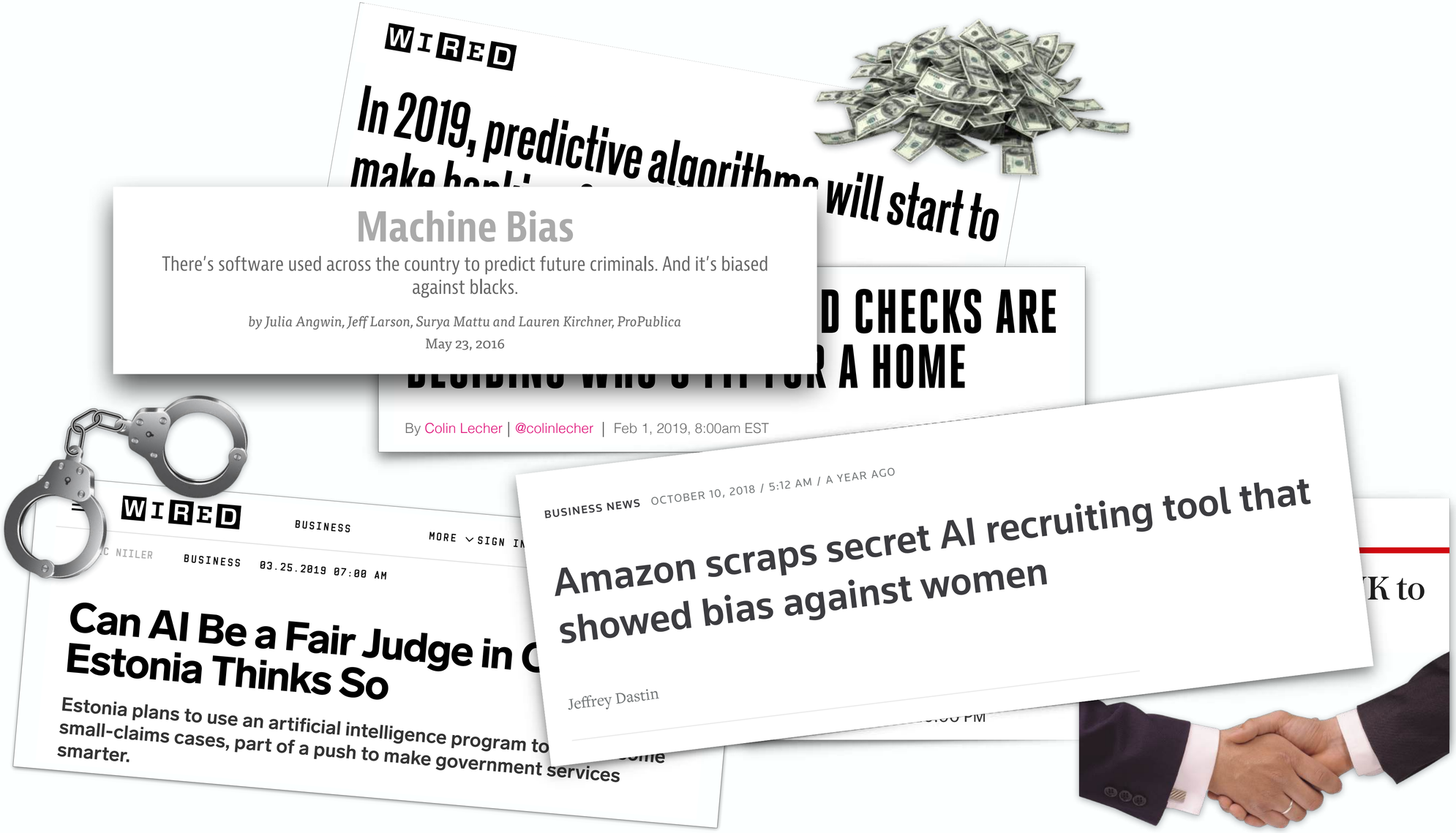
Execution Time

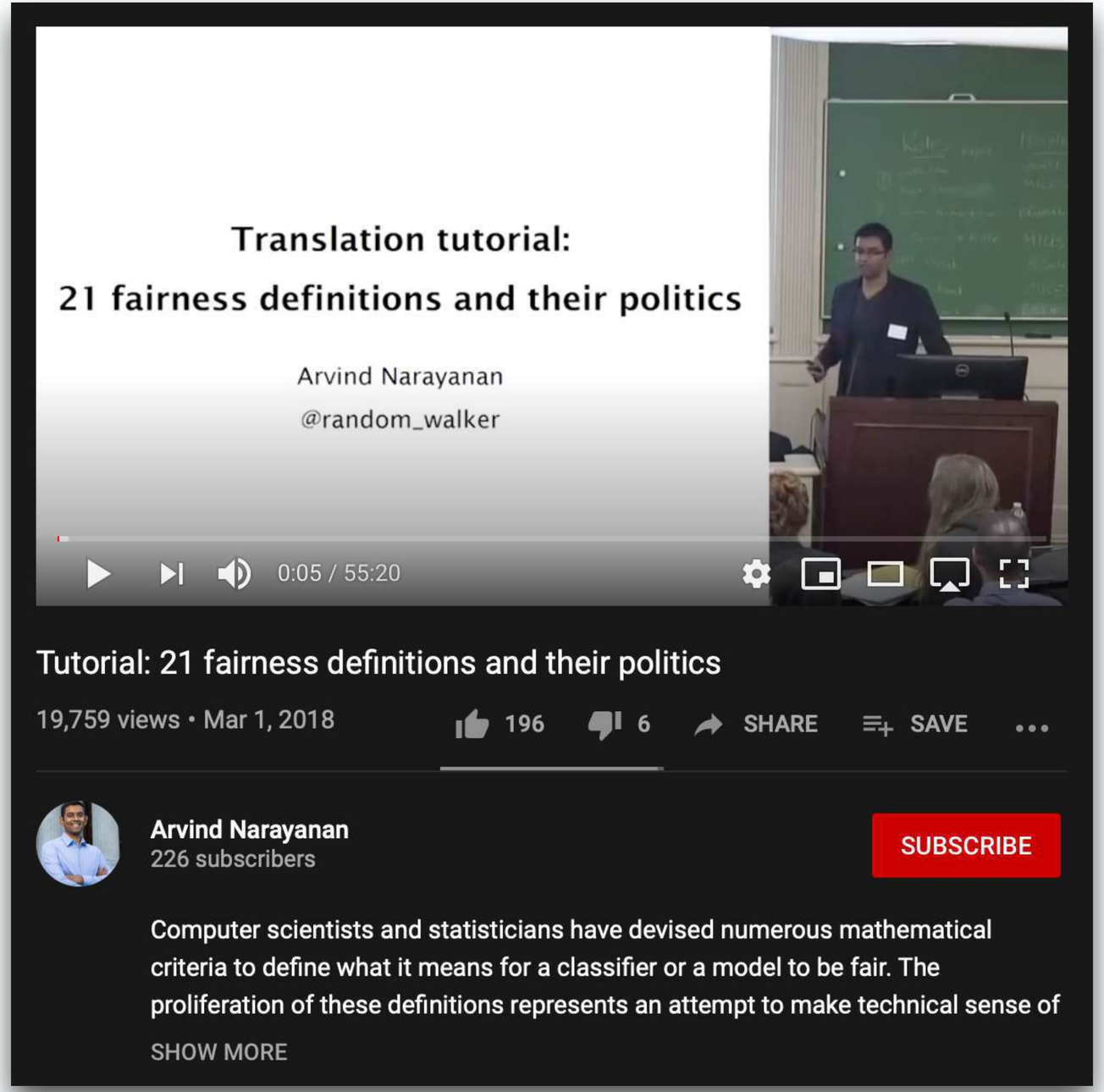
L	U	Boxes	Symbolic	DeepPoly		Product	
				Input Range Partitioning	ReCIPH	Input Range Partitioning	ReCIPH
1	2	0,08s	0,14s	0,26s	0,11s	0,26s	0,12s
	6	0,16s	0,31s	0,51s	0,20s	0,35s	0,20s
0.5	2	8,88s	5,76s	2,60s	1,61s	2,10s	1,61s
	6	64,67s	40,90s	2,65s	1,63s	2,10s	1,62s

Fairness


Google Translate
Sign in
Text
Documents
DETECT LANGUAGE
ENGLISH
FRENCH
ENGLISH
SPANIS
A nurse
A doctor
16/5000
Une infirmière
Un médecin

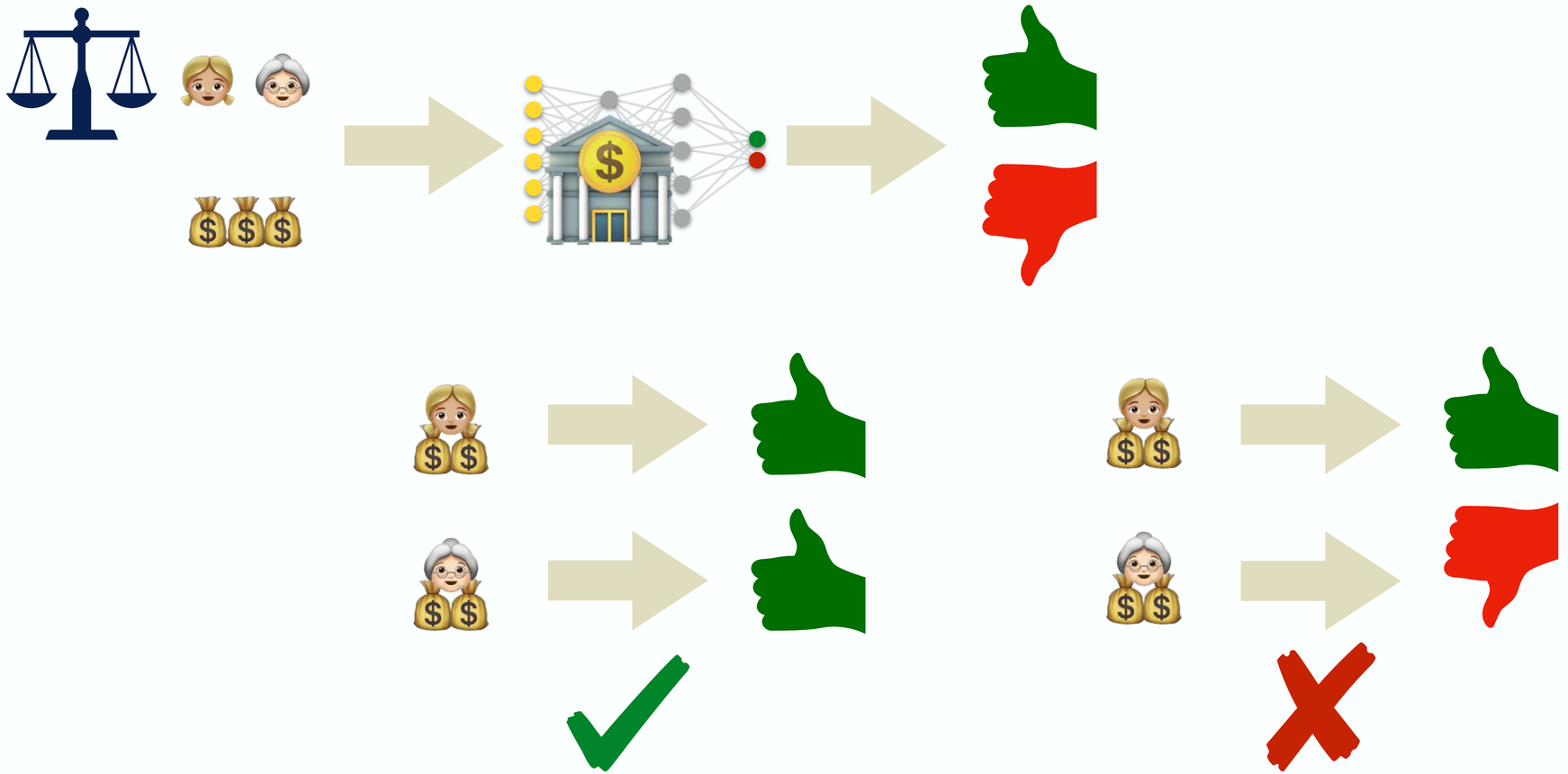
Machine Learning Impacts Our Society




Translation tutorial:
21 fairness definitions and their politics
Arvind Narayanan
@random_walker
0:05 / 55:20
Tutorial: 21 fairness definitions and their politics
19,759 views • Mar 1, 2018
196
6
SHARE
SAVE
Arvind Narayanan
226 subscribers
SUBSCRIBE
Computer scientists and statisticians have devised numerous mathematical criteria to define what it means for a classifier or a model to be fair. The proliferation of these definitions represents an attempt to make technical sense of
SHOW MORE

Dependency Fairness [Galhotra17]

Prediction is Independent of Sensitive Input Values



Dependency Fairness

$$\mathscr{F}_i \overset{\text{def}}{=} \{[\![M]\!] \mid \text{UNUSED}_i([\![M]\!])\}$$

$\mathscr{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\text{UNUSED}_i(T) \overset{\text{def}}{=} \forall t, t' \in T : t_0(x_{0,i}) \neq t'_0(x_{0,i}) \wedge (\forall 0 \leq j \leq |L_0| : j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j})) \Rightarrow t_\omega = t'_\omega$$

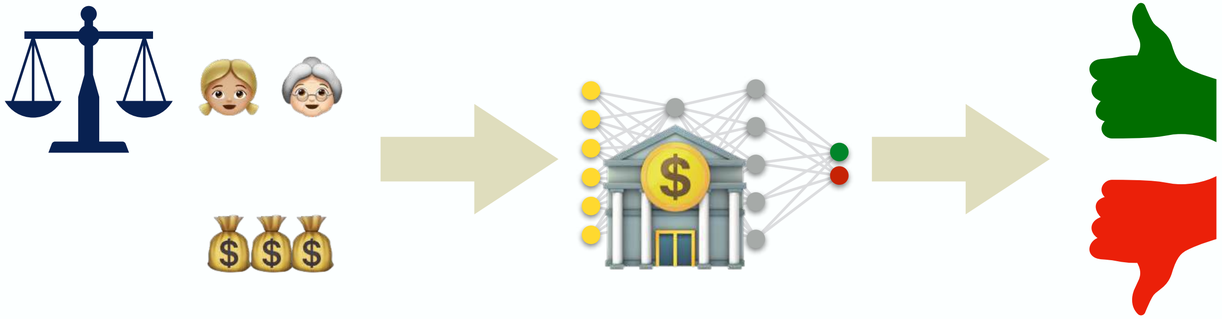
$\eta(t_0) = \eta(t'_0)$ $\rho(t_0) = \rho(t'_0)$

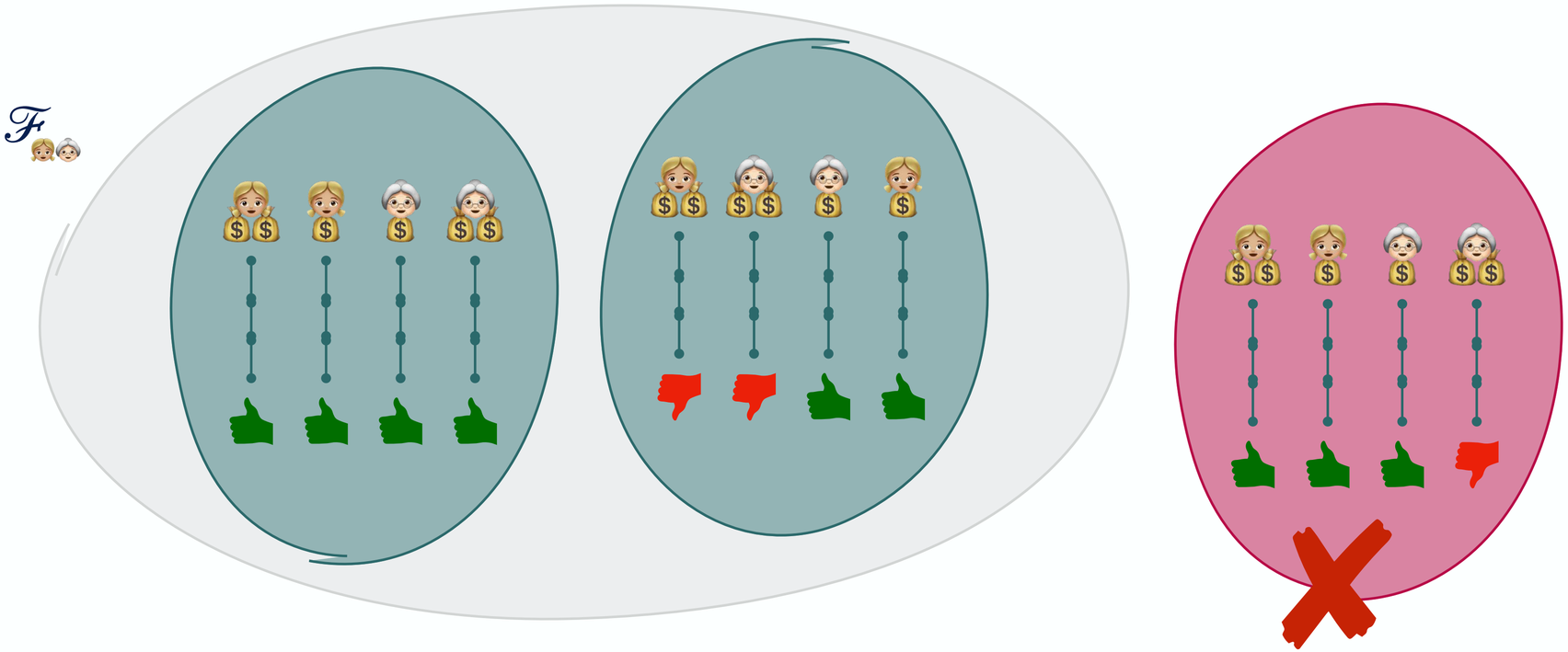
η:

$$\eta(x0j) = \begin{cases} \top & j = i \\ x0j & \text{otherwise} \end{cases}$$

Intuitively: inputs differing only on the value of the sensitive input node $x_{0,i}$ should lead to the same **classification outcome**

Dependency Fairness





Dependency Fairness

$$\mathcal{F}_i \overset{\text{def}}{=} \{[\![M]\!] \mid \text{UNUSED}_i([\![M]\!])\}$$

$\mathcal{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\text{UNUSED}_i(T) \overset{\text{def}}{=} \forall t, t' \in T: t_0(x_{0,i}) \neq t'_0(x_{0,i}) \wedge \eta(t_0) = \eta(t'_0)$$
$$(\forall 0 \leq j \leq |L_0| : j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j}))$$
$$\Rightarrow t_\omega = t'_\omega \quad \rho(t_0) = \rho(t'_0)$$

η:

$$\eta(x0j) = \begin{cases} \top & j = i \\ x0j & \text{otherwise} \end{cases}$$

Intuitively: inputs differing only on the value of the sensitive input node $x_{0,i}$ should lead to the same **classification outcome**

Theorem

$$M \vDash \mathcal{F}_i \Leftrightarrow \{[\![M]\!]\} \subseteq \mathcal{F}_i$$

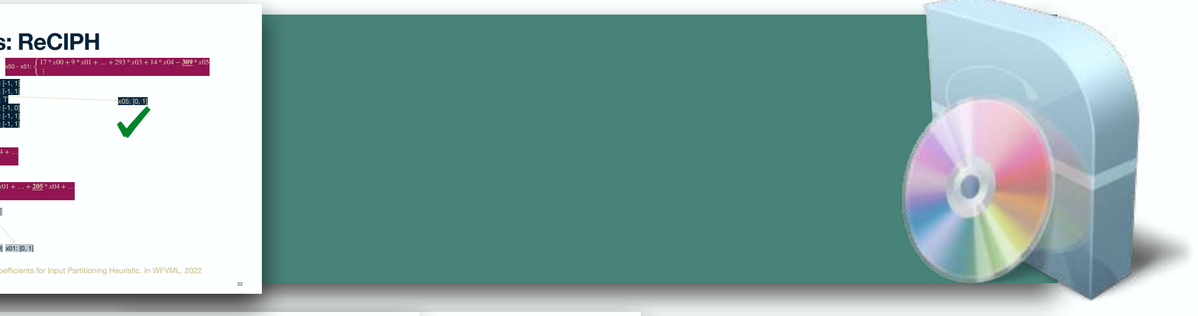
Corollary

$$M \vDash \mathcal{F}_i \Leftarrow [\![M]\!] \subseteq [\![M]\!]^\natural \subseteq \mathcal{F}_i$$

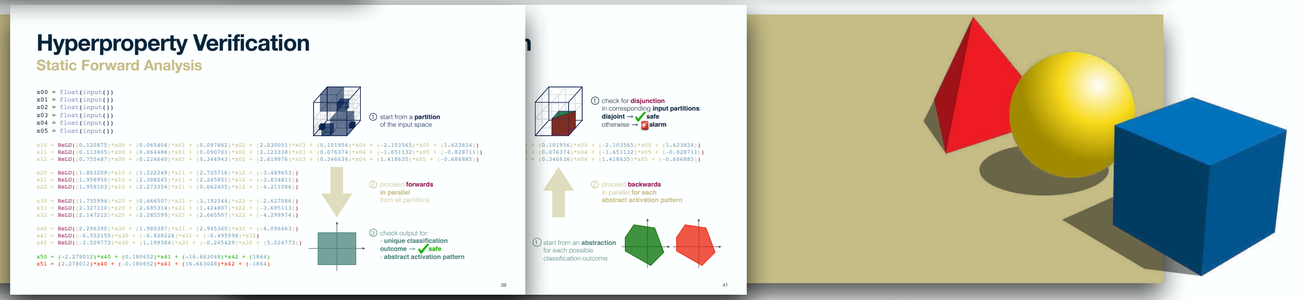
Hypersafety Verification [Urban20]

3-Step Recipe

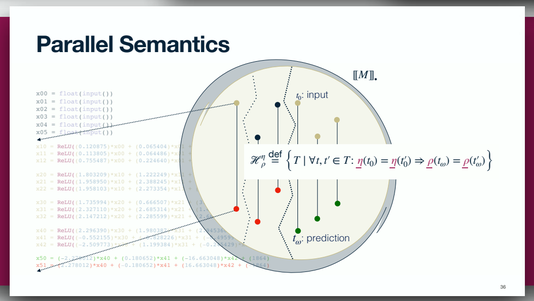
practical tools

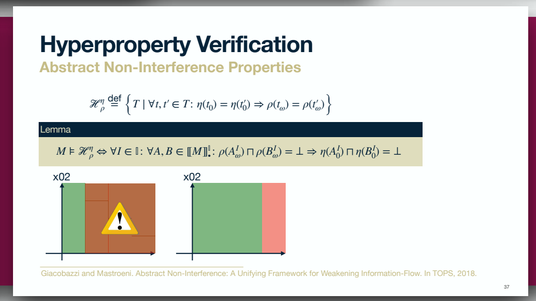


abstract semantics, abstract domains

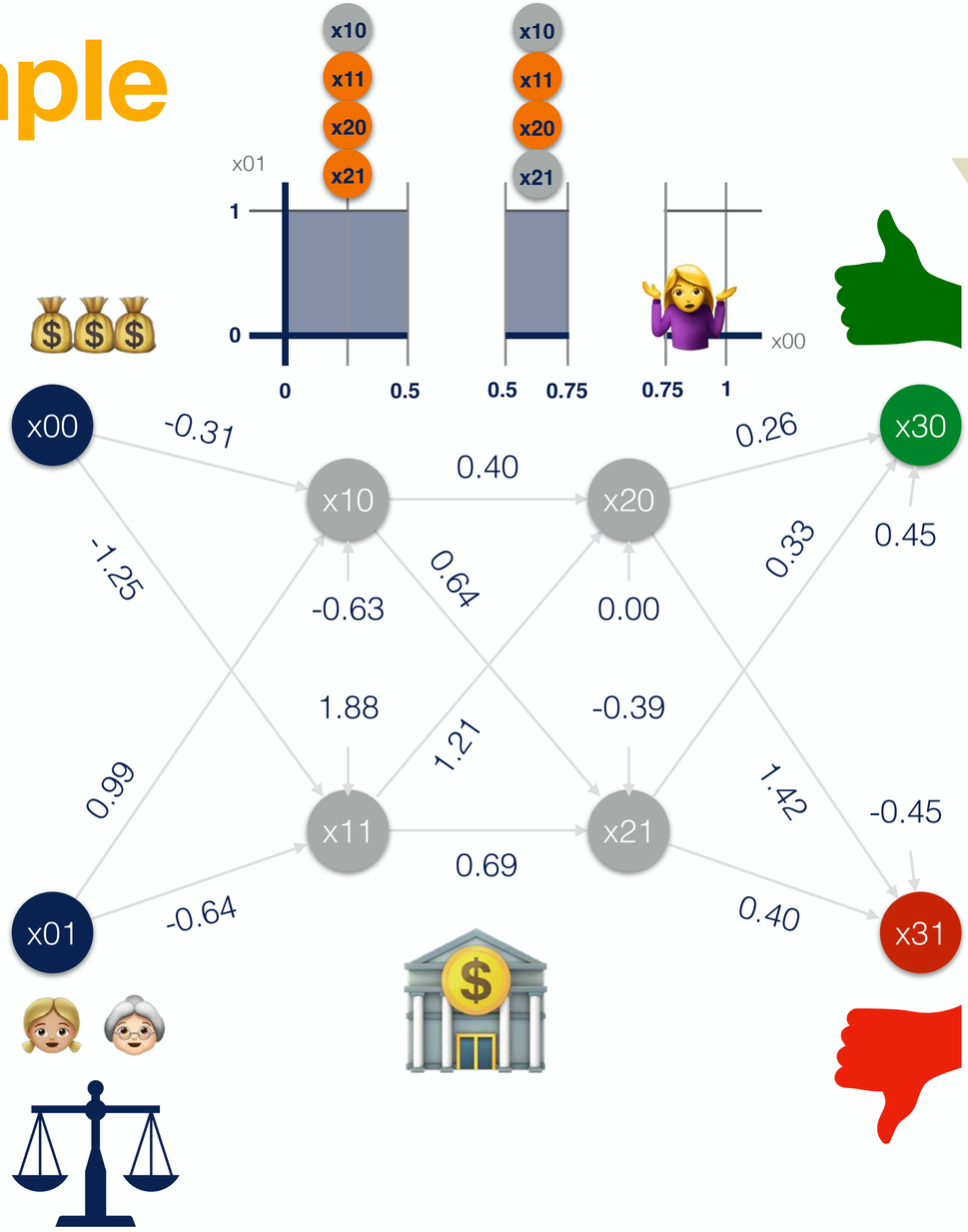


concrete semantics





Example

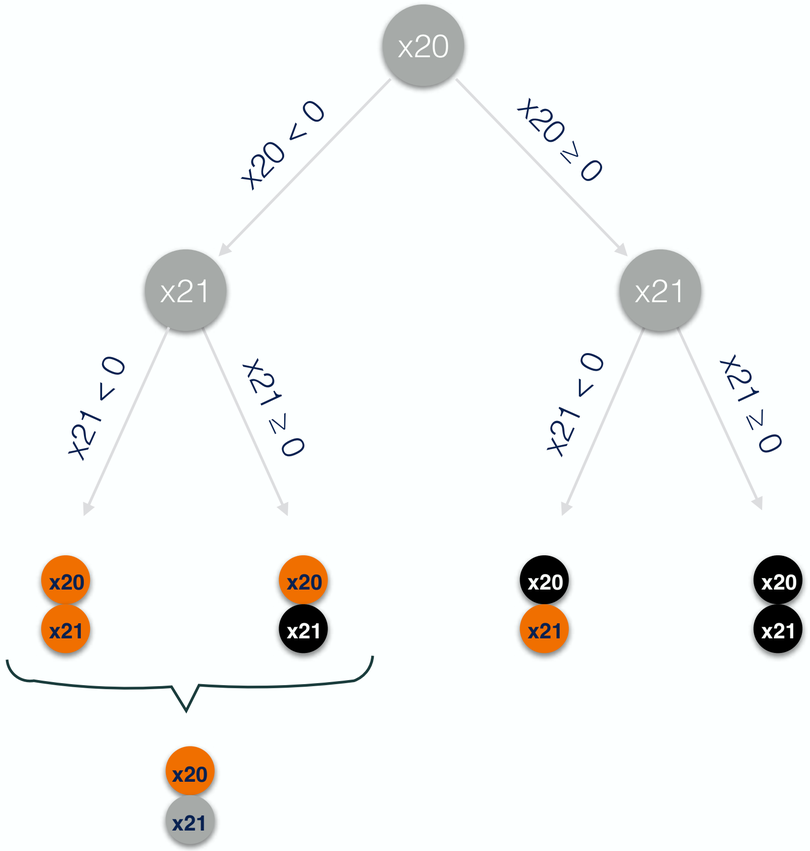


L = 0.25

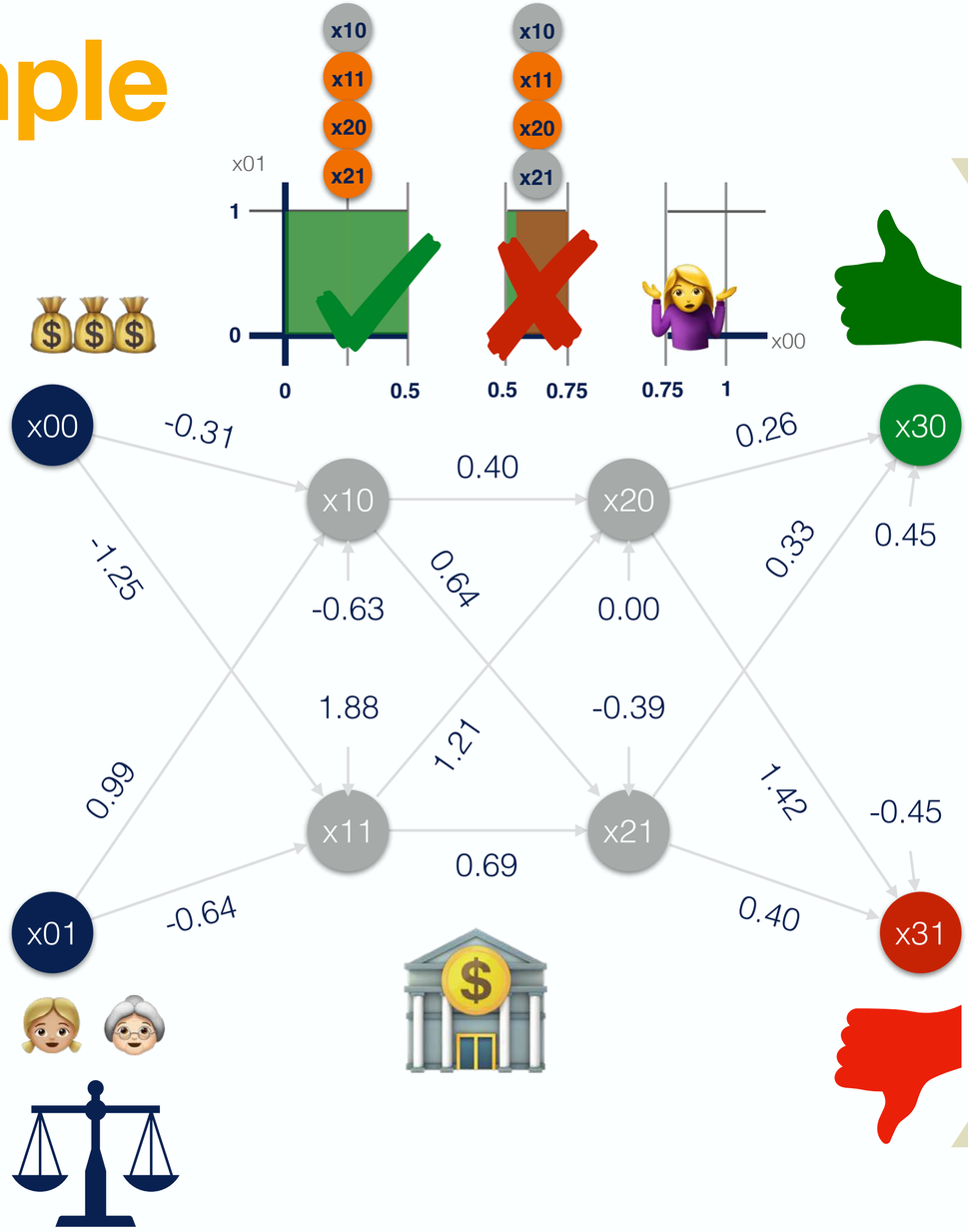
U = 2

```
x00 = input()
x01 = input()

x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88

x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11

x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)

x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21

x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)

return '👍' if x31 < 30 else '👎'
```

(Abstract) Activation Patterns



Example

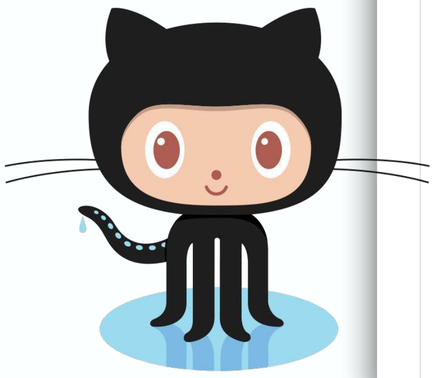


L = 0.25

U = 2

```
x00 = input()
x01 = input()
x10 = -0.31 * x00 + 0.99 * x01 + (-0.63)
x11 = -1.25 * x00 + (-0.64) * x01 + 1.88
x10 = 0 if x10 < 0 else x10
x11 = 0 if x11 < 0 else x11
x20 = 0.40 * x10 + 1.21 * x11 + 0.00
x21 = 0.64 * x10 + 0.69 * x11 + (-0.39)
x20 = 0 if x20 < 0 else x20
x21 = 0 if x21 < 0 else x21
1.16 * x20 + 0.07 * x21 ≤ 0.90   1.16 * x20 + 0.07 * x21 ≥ 0.90
x30 = 0.26 * x20 + 0.33 * x21 + 0.45
X31 = 1.42 * x20 + 0.40 * x21 + (-0.45)
x30 ≥ x31   x31 ≥ x30
return '👍' if x31 < 30 else '👎'
```

Libra



caterinaurban / **Libra**

Code | Issues | Pull requests | Actions | Projects | Security | Insights

master | 2 branches | 0 tags | Go to file | Code

caterinaurban README — 9f830db on Aug 8 — 53 commits

src	RQ5 and RQ6 reproducibility	4 months ago
.gitignore	RQ1 reproducibility	4 months ago
LICENSE	Initial prototype	2 years ago
README.md	RQ5 and RQ6 reproducibility	4 months ago
README.pdf	README	4 months ago
icon.png	icon	4 months ago
libra.png	icon	4 months ago
requirements.txt	some documentation	4 months ago
setup.py	some documentation	4 months ago

README.md

Libra

Nowadays, machine-learned software plays an increasingly important role in critical decision-making in our social, economic, and civic lives.

About

No description or website provided.

#abstract-interpretation
#static-analysis
#machine-learning
#neural-networks #fairness

Readme

MPL-2.0 License

Releases

No releases published

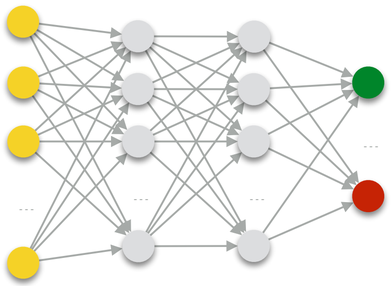
Packages

No packages published

Languages

- **Python** 98.7%
- **Shell** 1.3%

Scalability-vs-Precision Tradeoff

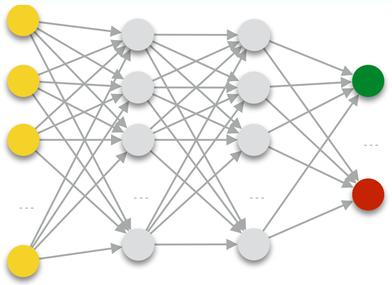


17 inputs
4 HL * 5 N
2 classes
86% accuracy

- a larger U or a smaller L improves **precision**
- a more precise forward analysis improves **scalability**

L	U	⬟ BOXES					▲ SYMBOLIC					✶ DEEPPOLY				
		INPUT	|C|	|F|		TIME	INPUT	|C|	|F|		TIME	INPUT	|C|	|F|		TIME
0.5	4	15.28%	37	0	0	8s	58.33%	79	8	20	1m 26s	69.79%	115	10	39	3m 18s
	6	17.01%	39	6	6	51s	69.10%	129	22	61	5m 41s	80.56%	104	23	51	7m 53s
	8	51.39%	90	28	85	12m 2s	82.64%	88	31	67	12m 35s	91.32%	84	27	56	19m 33s
	10	79.86%	89	34	89	34m 15s	93.06%	98	40	83	42m 32s	96.88%	83	29	58	43m 39s
0.25	4	59.09%	1115	20	415	54m 32s	95.94%	884	39	484	54m 31s	98.26%	540	65	293	14m 29s
	6	83.77%	1404	79	944	37m 19s	98.68%	634	66	376	23m 31s	99.70%	322	79	205	13m 25s
	8	96.07%	869	140	761	1h 7m 29s	99.72%	310	67	247	1h 3m 33s	99.98%	247	69	177	22m 52s
	10	99.54%	409	93	403	1h 35m 20s	99.98%	195	52	176	1h 2m 13s	100.00%	111	47	87	34m 56s
0.125	4	97.13%	12449	200	9519	3h 33m 48s	99.99%	1101	60	685	47m 46s	99.99%	768	81	415	19m 1s
	6	99.83%	5919	276	4460	3h 23m	100.00%	988	77	606	26m 47s	100.00%	489	80	298	16m 54s
	8	99.98%	1926	203	1568	2h 14m 25s	100.00%	404	73	309	46m 31s	100.00%	175	57	129	20m 11s
	10	100.00%	428	95	427	1h 39m 31s	100.00%	151	53	141	57m 32s	100.00%	80	39	62	28m 33s
0	4	100.00%	19299	295	15446	6h 13m 24s	100.00%	1397	60	885	40m 5s	100.00%	766	87	425	**16m 41s**
	6	100.00%	4843	280	3679	2h 24m 7s	100.00%	763	66	446	35m 24s	100.00%	401	81	242	32m 29s
	8	100.00%	1919	208	1567	2h 9m 59s	100.00%	404	73	309	45m 48s	100.00%	193	68	144	24m 16s
	10	100.00%	486	102	475	1h 41m 3s	100.00%	217	55	192	1h 2m 11s	100.00%	121	50	91	30m 53s

Scalability wrt Neural Network Size



23 inputs
2 HL * 5 N
2 classes

23 inputs
4 HL * 3 N
2 classes

23 inputs
4 HL * 5 N
2 classes

23 inputs
4 HL * 10 N
2 classes

23 inputs
9 HL * 5 N
2 classes

- **scalability degrades for larger neural networks (less for models with fewer nodes per layer)**
- **a larger U sometimes improves scalability**

|M|	U	BOXES					SYMBOLIC					DEEPPOLY				
		INPUT	|C|	|F|		TIME	INPUT	|C|	|F|		TIME	INPUT	|C|	|F|		TIME
10 ○ ● ⊕	4	88.26%	1482	77	1136	33m 55s	95.14%	1132	65	686	19m 5s	93.99%	1894	77	992	29m 55s
	6	99.51%	769	51	723	1h 10m 25s	99.93%	578	47	447	39m 8s	99.83%	1620	54	1042	1h 24m 24s
	8	100.00%	152	19	143	3h 47m 23s	100.00%	174	18	146	1h 51m 2s	100.00%	1170	26	824	8h 2m 27s
	10	100.00%	1	1	1	**55m 58s**	100.00%	1	1	1	56m 8s	100.00%	1	1	1	56m 43s
12 △ ▲ ⅄	4	49.83%	719	9	329	13m 43s	72.29%	1177	11	559	24m 9s	60.52%	1498	14	423	10m 32s
	6	72.74%	1197	15	929	2h 6m 49s	98.54%	333	7	195	20m 46s	66.46%	1653	17	594	15m 44s
	8	98.68%	342	9	284	1h 46m 43s	98.78%	323	9	190	1h 27m 18s	70.87%	1764	18	724	2h 19m 11s
	10	99.06%	313	7	260	1h 21m 47s	99.06%	307	5	182	**1h 13m 55s**	80.76%	1639	18	1007	3h 22m 11s
20 ◊ ◆ ⬙	4	38.92%	1044	18	39	2m 6s	51.01%	933	31	92	15m 28s	49.62%	1081	34	79	3m 2s
	6	46.22%	1123	62	255	20m 51s	61.60%	916	67	405	44m 40s	59.20%	1335	90	356	22m 13s
	8	64.24%	1111	96	792	2h 24m 51s	74.27%	1125	78	780	3h 26m 20s	69.69%	1574	127	652	5h 6m 7s
	10	85.90%	1390	71	1339	>13h	89.27%	1435	60	1157	>13h	76.25%	1711	148	839	**4h 36m 23s**
40 □ ■ ⬙	4	0.35%	10	0	0	1m 39s	34.62%	768	1	1	6m 56s	26.39%	648	2	3	10m 11s
	6	0.35%	10	0	0	1m 38s	34.76%	817	4	5	43m 53s	26.74%	592	8	10	1h 23m 11s
	8	0.42%	12	1	2	14m 37s	35.56%	840	21	28	2h 48m 15s	27.74%	686	32	42	2h 43m 2s
	10	0.80%	23	10	13	1h 48m 43s	37.19%	880	50	75	**11h 32m 21s**	30.56%	699	83	121	>13h
45 ⬠ ⬟ ✳	4	1.74%	50	0	0	1m 38s	41.98%	891	14	49	10m 14s	36.60%	805	6	8	2m 47s
	6	2.50%	72	3	22	4m 35s	45.00%	822	32	143	45m 42s	38.06%	847	25	50	5m 7s
	8	9.83%	282	25	234	25m 30s	47.78%	651	46	229	1h 14m 5s	42.53%	975	74	180	25m 1s
	10	18.68%	522	33	488	1h 51m 24s	49.62%	714	51	294	**3h 23m 20s**	48.68%	1087	110	373	1h 58m 34s

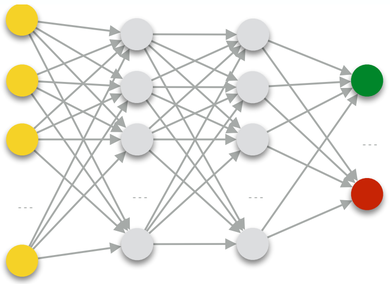
Scalability wrt Input Space Size

|M|	QUERY	BOXES				SYMBOLIC				DEEPPOLY			
		INPUT	|C|	|F|	TIME	INPUT	|C|	|F|	TIME	INPUT	|C|	|F|	TIME
80	F 0.009%	99.931% 0.009%	11	0 0	3m 5s	99.961% 0.009%	17	0 0	**3m 2s**	99.957% 0.009%	10	0 0	2m 36s
	E 0.104%	99.583% 0.104%	61	0 0	3m 6s	99.783% 0.104%	89	0 0	**3m 10s**	99.753% 0.104%	74	0 0	2m 44s
	D 1.042%	97.917% 1.020%	151	0 0	2m 56s	99.258% 1.034%	297	0 0	**3m 41s**	98.984% 1.031%	477	0 0	2m 58s
	C 8.333%	83.503% 6.958%	506	2 3	2h 1m	95.482% 7.956%	885	25 34	>13h	93.225% 7.768%	1145	23 33	**12h 57m 37s**
	B 50%	25.634% 12.817%	5516	7 11	**1h 28m 6s**	76.563% 38.281%	4917	123 182	>13h	63.906% 31.953%	7139	117 152	>13h
	A 100%	0.052% 0.052%	12	0 0	25m 51s	61.385% 61.385%	5156	73 102	**10h 25m 2s**	43.698% 43.698%	4757	68 88	>13h
320	F 0.009%	99.931% 0.009%	6	0 0	3m 15s	99.944% 0.009%	9	0 0	**3m 35s**	99.931% 0.009%	6	0 0	3m 30s
	E 0.104%	99.583% 0.104%	121	0 0	3m 39s	99.627% 0.104%	120	0 0	**6m 34s**	99.583% 0.104%	31	0 0	4m 22s
	D 1.042%	97.917% 1.020%	151	0 0	6m 18s	98.247% 1.024%	597	0 0	**21m 9s**	97.917% 1.020%	301	0 0	9m 35s
	C 8.333%	83.333% 6.944%	120	0 0	30m 37s	88.294% 7.358%	755	0 0	**1h 36m 35s**	83.342% 6.945%	483	0 0	52m 29s
	B 50%	25.000% 12.500%	5744	0 0	2h 24m 36s	46.063% 23.032%	4676	0 0	**7h 25m 57s**	25.074% 12.537%	5762	4 4	>13h
	A 100%	0.000% 0.000%	0	0 0	2h 54m 25s	24.258% 24.258%	2436	0 0	**9h 41m 36s**	0.017% 0.017%	4	0 0	5h 3m 33s
1280	F 0.009%	99.931% 0.009%	11	0 0	7m 35s	99.948% 0.009%	10	0 0	**24m 42s**	99.931% 0.009%	6	0 0	7m 6s
	E 0.104%	99.583% 0.104%	31	0 0	15m 49s	99.674% 0.104%	71	0 0	**51m 52s**	99.583% 0.104%	31	0 0	15m 14s
	D 1.042%	97.917% 1.020%	151	0 0	1h 49s	98.668% 1.028%	557	0 0	**3h 31m 45s**	97.917% 1.020%	301	0 0	1h 3m 33s
	C 8.333%	83.333% 6.944%	481	0 0	**7h 11m 39s**	–	–	– –	>13h	83.333% 6.944%	481	0 0	7h 12m 57s
	B 50%	–	–	– –	>13h	–	–	– –	>13h	–	–	– –	>13h
	A 100%	–	–	– –	>13h	–	–	– –	>13h	–	–	– –	>13h

the **size of the queried input space** (rather than the size of the neural network) is the **most important factor** for scalability!

Scalability-vs-Precision Tradeoff

Product Domain



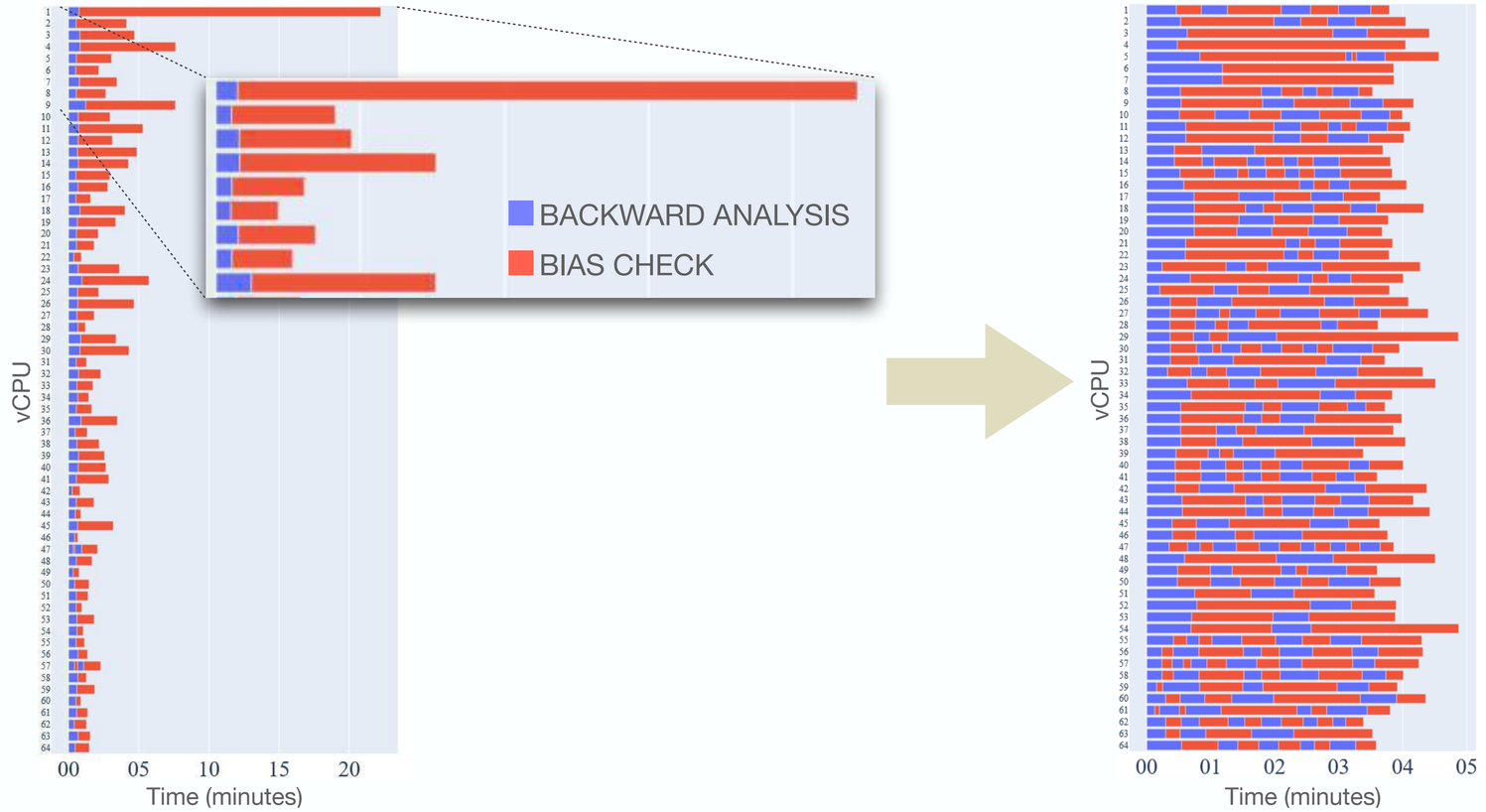
23 inputs
4 HL * 5 N
2 classes

L	U	Intervals	Symbolic	DeepPoly	Neurify	**Product**	
0.5	3	37,9 %	48,8 %	48,9 %	46,5 %	**59,2 %**	**+ 10,3%**
	5	41,0 %	56,1 %	56,3 %	53,1 %	**68,2 %**	**+ 11,9%**
0.25	3	70,6 %	83,6 %	81,8 %	81,4 %	**87,0 %**	**+ 3,4%**
	5	83,1 %	91,7 %	91,6 %	92,3 %	**95,5 %**	**+ 3,2%**

L	U	Intervals	Symbolic	DeepPoly	Neurify	**Product**	
0.5	3	47s	60s	96s	37s	**119s**	**+ 23-59s**
	5	246s	736s	557s	362s	**835s**	**+ 99-278s**
0.25	3	498s	554s	396s	420s	**534s**	**- 20s / + 36-138s**
	5	3369s	2674s	2840s	2920s	**3716s**	**+ 796-1042s**

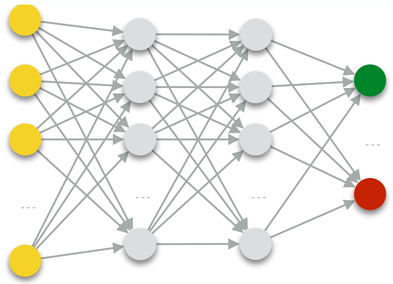
Forward and Backward Analysis

Perfect Parallelization



Scalability-vs-Precision Tradeoff

Perfect Parallelization



23 inputs
4 HL * 5 N
2 classes

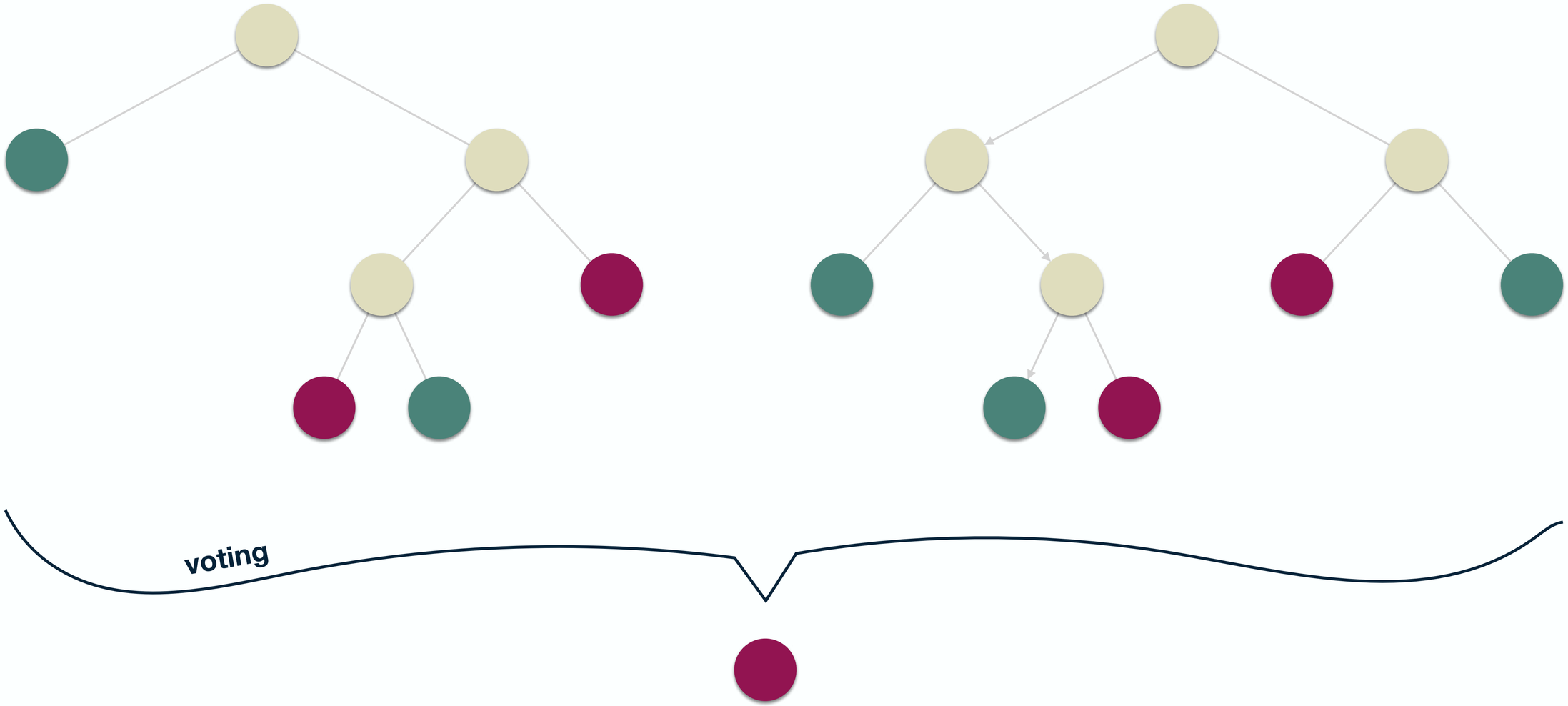
L	U	Intervals	Symbolic	DeepPoly	Neurify	Product
0.5	3	37,9 %	48,8 %	48,9 %	46,5 %	**59,2 %**
	5	41,0 %	56,1 %	56,3 %	53,1 %	**68,2 %**
0.25	3	70,6 %	83,6 %	81,8 %	81,4 %	**87,0 %**
	5	83,1 %	91,7 %	91,6 %	92,3 %	**95,5 %**

L	U	Intervals		Symbolic		DeepPoly		Neurify		Product	
0.5	3	~~47s~~	36s	~~60s~~	42s	~~96s~~	95s	~~37s~~	32s	~~119s~~	118s
	5	~~246s~~	248s	~~736s~~	550s	~~557s~~	227s	~~362s~~	237s	~~835s~~	496s
0.25	3	~~498s~~	349s	~~554s~~	355s	~~396s~~	320s	~~420s~~	320s	~~534s~~	432s
	5	~~3369s~~	1603s	~~2674s~~	1268s	~~2840s~~	1328s	~~2920s~~	1554s	~~3716s~~	1318s

1.9x - 2.8x FASTER

Other Machine Learning Models

Decision Tree Ensembles



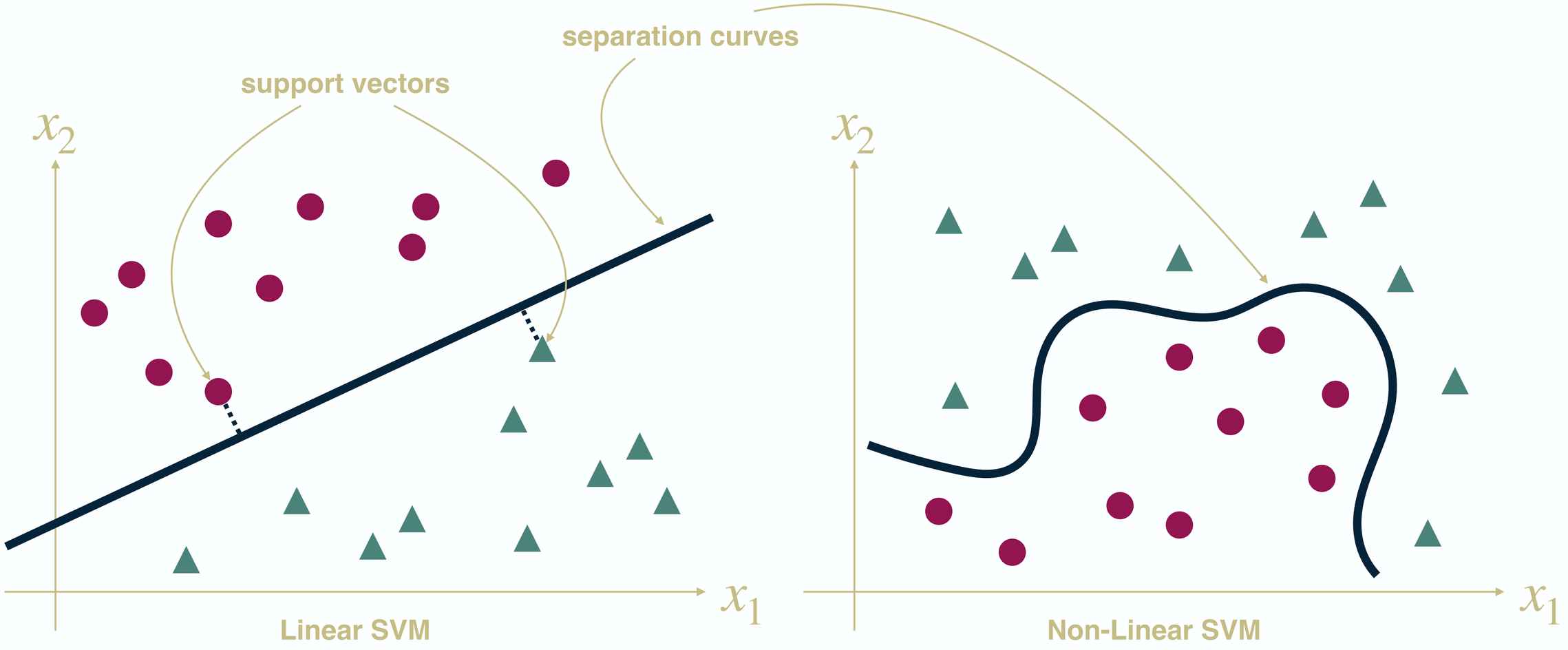
Decision Tree Ensembles

- **G. Einziger, M. Goldstein, Y. Sa'ar, and I. Segall.** Verifying Robustness of Gradient Boosted Models. In AAAI 2019.
  SMT-based approach for local stability
- **N. Sato, H. Kuruma, Y. Nakagawa, and H. Ogawa.** Formal Verification of Decision-Tree Ensemble Model and Detection of its Violating-Input-Value Ranges. 2020.
  approach for safety verification

- **J. Törnblom and S. Nadjm-Tehrani.** Formal Verification of Input-Output Mappings of Tree Ensembles. 2020.
  F. Ranzato and M. Zanella. Abstract Interpretation of Decision Tree Ensemble Classifiers. In AAAI 2020.
  S. Calzavara, P. Ferrara, and C. Lucchese. Certifying Decision Trees Against Evasion Attacks by Program Analysis. In ESORICS 2020.
  abstract interpretation-based approaches for local stability

Support Vector Machines (SVMs)

Support Vector Machines (SVMs)

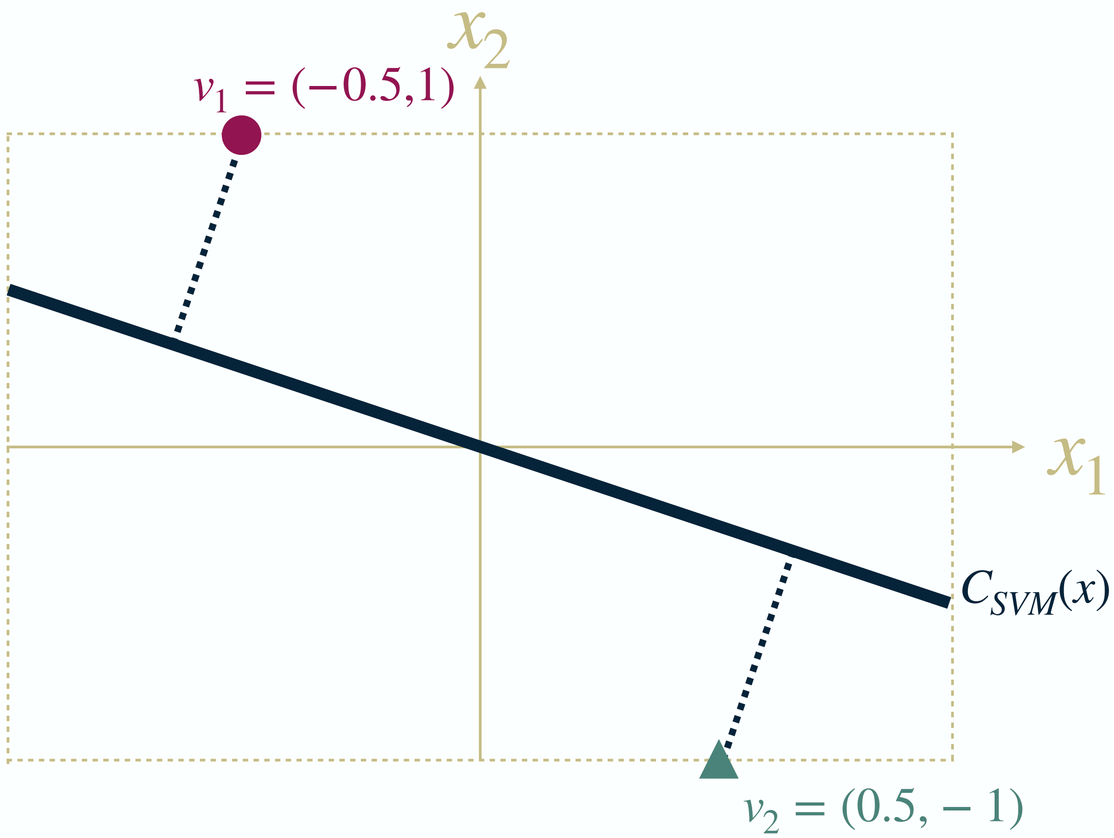


Support Vector Machines (SVMs)

Example

● $\mapsto -1$

▲ $\mapsto 1$



weights

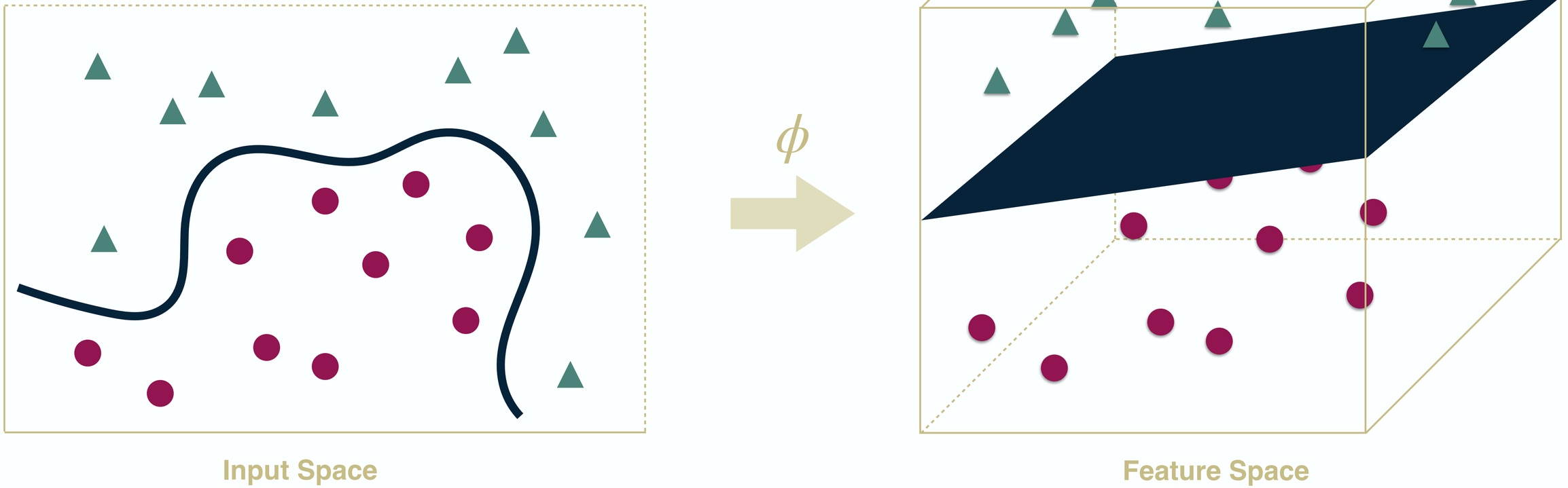
$$C_{SVM}(x) = sgn\left(SVM(x)\right) = sgn\left(-1 * 0.5(v_1 \cdot x) + 1 * 0.5(v_2 \cdot x)\right)$$

$$= sgn\left(0.5x_1 - x_2\right)$$

Non-Linear SVMs

Kernel Functions

- Polynomial
- Radial Basis Function (RBF)

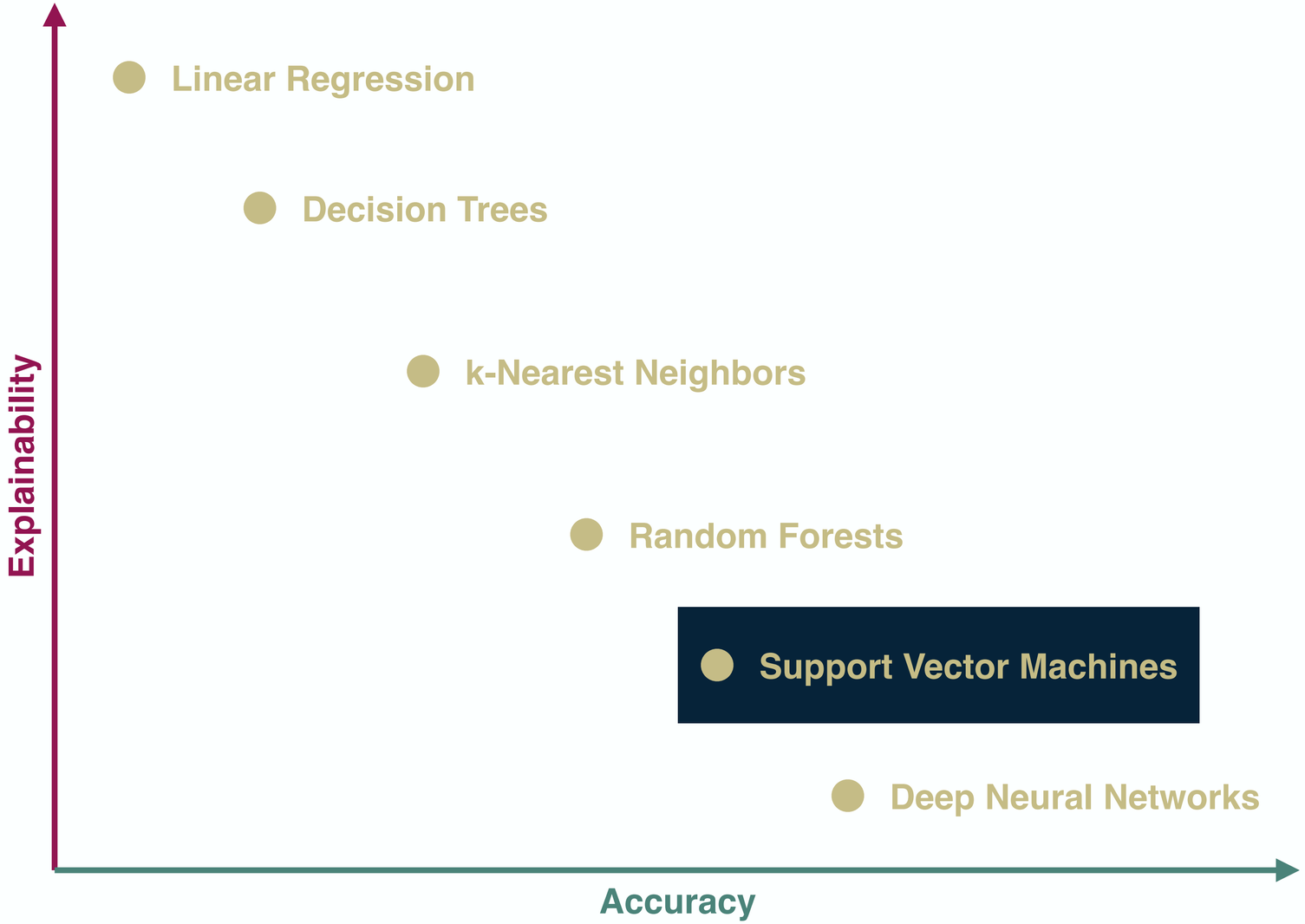


Support Vector Machines

- **Francesco Ranzato and Marco Zanella**. Robustness Verification of Support Vector Machines. In SAS, pages 271-295, 2019.
  abstract interpretation-based approach for local stability

SVM Explainability

Explainability vs Performance



Feature Importance

Measuring Contribution of Input Features to Prediction

	Local	Global	Model-		Performance	Effect
			Specific	Agnostic	-Based	
Permutation Feature Importance (PFI)		X		X	X	
Partial Dependence (PD) Plots		X		X		X
Individual Conditional Expectation (ICE) Plots		X		X		X
Accumulated Local Effects (ALE) Plots		X		X		X
Local Interpretable Model-Agnostic Explanations (LIME)	X			X		X
SHapley Additive exPlanations (SHAP)	X			X		X
Individual Conditional Importance (ICI) Curves	X			X	X	
Partial Importance (PI) Curves	X			X	X	
Shapley Feature Importance (SFIMP)		X		X	X	
Input Gradients	X			X	X	X
Abstract Feature Importance (AFI)	X	X	X			X

Abstract Feature Importance (AFI) [Pal24]

Why Another Feature Importance Measure?

Permutation Feature Importance (PFI)	• **result may greatly vary depending on the dataset** • **resource intensive** when the number of feat • **misleading result when feature** • **quality of the result he** **model accuracy**
Local Interpretable Model-Agnostic Explanations (L	• **requi** **ningful optimal neighborhood**: **le** and easily manipulable explanations **es that the decision boundary is linear at the local level**, but there is no theoretically guarantee that this is the case
S anations (SHAP)	• Shapley values estimations **depend on the dataset** • **assumes that features are independent** • **has a very high computational cost**, even for small models
Abstract Feature Importance (AFI)	• yields **a formally correct by construction approximation** • **does not depend** from a dataset nor the accuracy of the model • **extremely fast to compute**, whatever the number of features • supports both linear and non-linear kernel functions

"Make Sense" but Give No Guarantees

Abstract Interpretation of SVMs

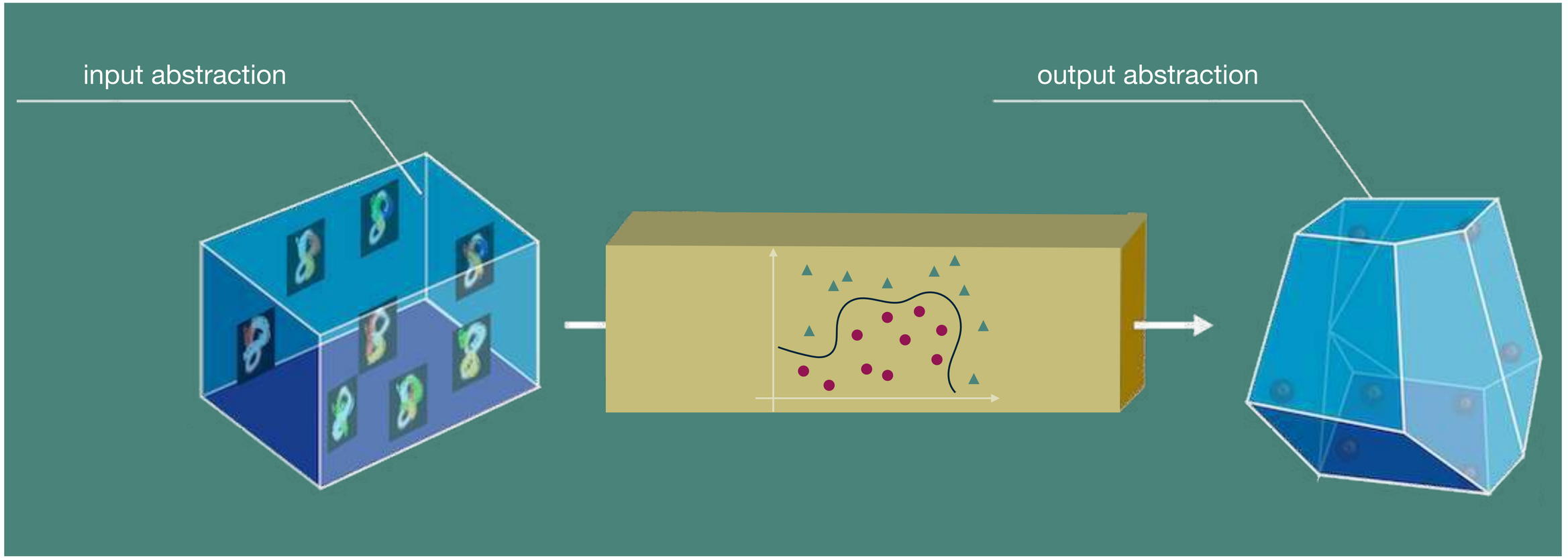
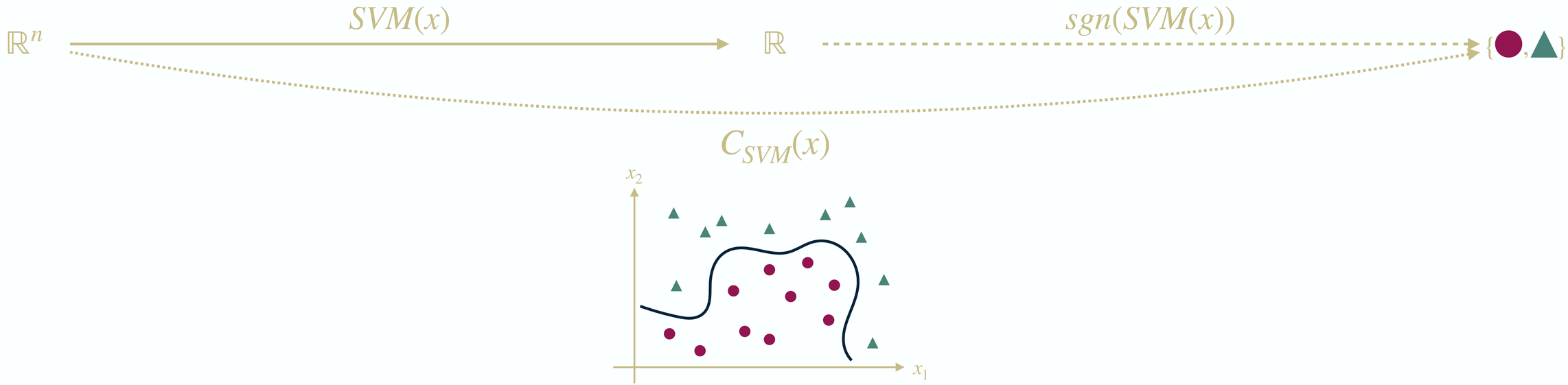


Image taken (and modified) from http://safeai.ethz.ch

Abstract Interpretation of SVMs

Reduced Affine Form (RAF) Abstraction

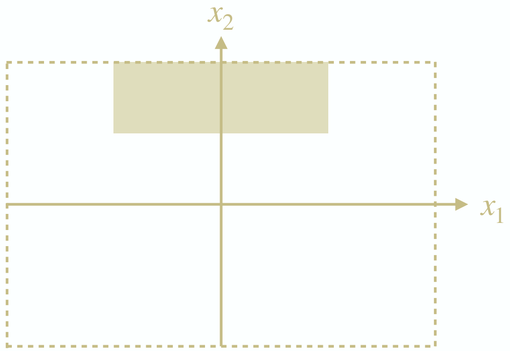


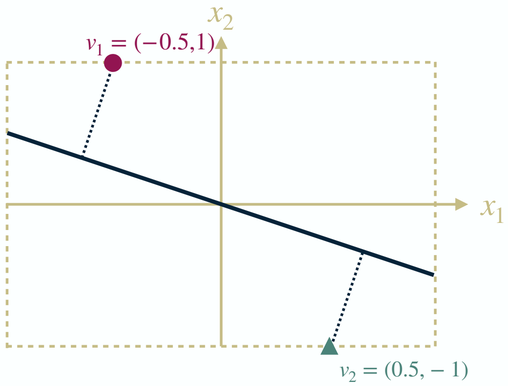
$$RAF_n \stackrel{\text{def}}{=} \left\{ a_0 + \sum_{i=1}^{n} a_i \epsilon_i + a_r \epsilon_r \mid a_0, a_1, \ldots a_n \in \mathbb{R}, a_r \in \mathbb{R}_{\geq 0} \right\} \cup \left\{ \top_{RAF} \right\}$$

Abstract Interpretation of SVMs

Example

$\mathbb{R}^n \xrightarrow{SVM(x) = -0.5(v_1 \cdot x) + 0.5(v_2 \cdot x)} \mathbb{R} \dashrightarrow sgn\,(0.5x_1 - 1x_2) \dashrightarrow \{\bullet, \blacktriangle\}$





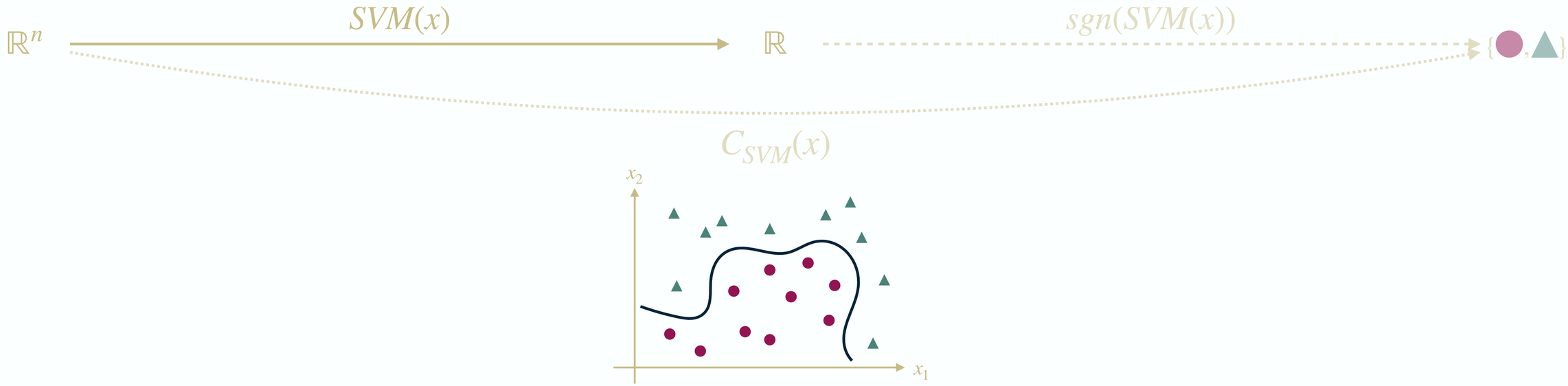
$(RAF_n)^n \xrightarrow{SVM^\sharp(x^\sharp)} RAF_n \dashrightarrow \{\bullet, \blacktriangle\}$

$(-0.5\epsilon_1,\ 0.75 - 0.25\epsilon_2)$

$-0.75 - 0.25\epsilon_1 + 0.25\epsilon_2$

$$
\begin{aligned}
&SVM^\sharp((-0.5\epsilon_1,\ 0.75 - 0.25\epsilon_2)) \\
&= -0.5(-0.5(-0.5\epsilon_1) + 1(0.75 - 0.25\epsilon_2)) + 0.5(0.5(-0.5\epsilon_1) - 1(0.75 - 0.25\epsilon_2)) \\
&= -0.5(0.75 + 0.25\epsilon_1 - 0.25\epsilon_2)) + 0.5(-0.75 - 0.25\epsilon_1 + 0.25\epsilon_2)) \\
&= -0.75 - 0.25\epsilon_1 + 0.25\epsilon_2
\end{aligned}
$$

Abstract Feature Importance (AFI) [Pal24]

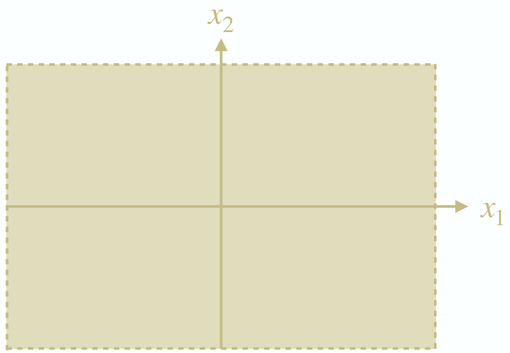


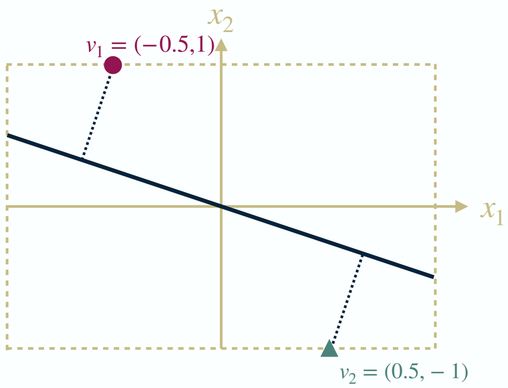
$$RAF_n \stackrel{\text{def}}{=} \left\{ a_0 + \sum_{i=1}^{n} a_i \epsilon_i + a_r \epsilon_r \mid a_0, a_1, \ldots a_n \in \mathbb{R}, a_r \in \mathbb{R}_{\geq 0} \right\} \cup \left\{ \top_{RAF} \right\}$$

Abstract Feature Importance (AFI) [Pal24]

Example

$\mathbb{R}^n$ $\xrightarrow{SVM(x) = -0.5(v_1 \cdot x) + 0.5(v_2 \cdot x)}$ $\mathbb{R}$ $\dashrightarrow$ $sgn\,(0.5x_1 - 1x_2)$ $\dashrightarrow$ {●, ▲}





$(RAF_n)^n$ $\xrightarrow{SVM^\sharp(x^\sharp)}$ RAF_n $\dashrightarrow$ {●, ▲}

(ϵ_1, ϵ_2) $\qquad$ $0.5\epsilon_1 - 1\epsilon_2$

$SVM^\sharp((\epsilon_1, \epsilon_2))$

$= -0.5(-0.5\epsilon_1 + 1\epsilon_2) + 0.5(0.5\epsilon_1 - 1\epsilon_2)$

$= 0.25\epsilon_1 - 0.5\epsilon_2 + 0.25\epsilon_1 - 0.5\epsilon_2$

$= 0.5\epsilon_1 - \epsilon_2$

AFI vs PFI

	Baseline	$N = 2k$ $\epsilon = 0.2$	$N = 10k$ $\epsilon = 0.2$	$N = 2k$ $\epsilon = 0.4$	$N = 10k$ $\epsilon = 0.4$	$N = 2k$ $\epsilon = 0.6$	$N = 5k$ $\epsilon = 0.6$	$N = 10k$ $\epsilon = 0.6$	$N = 2k$ $\epsilon = 0.8$	$N = 5k$ $\epsilon = 0.8$	$N = 10k$ $\epsilon = 0.8$
Adult	AFI (0.27s)	**0.0**	**0.0**	**1.0**	**0.0**	**1.0**	**1.41**	**1.0**	**1.0**	**1.41**	**1.0**
Linear	PFI (10009s)	2.45	2.45	2.24	2.45	2.24	**1.41**	2.24	2.24	**1.41**	2.24
Adult	AFI (0.48s)	**1.0**	**1.41**	**1.41**	**1.41**	**1.73**	**1.73**	**1.41**	**1.41**	**1.41**	**1.41**
RBF	PFI (25221s)	1.73	2.45	2.45	2.0	2.65	2.65	2.45	2.45	2.45	2.45
Adult	AFI (0.44s)	**1.0**	**1.0**	**0.0**	1.41	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Polynomial	PFI (9985s)	**1.0**	**1.0**	1.41	**1.0**	1.41	1.41	1.41	1.41	1.41	1.41
Compas	AFI (0.22s)	**1.41**	**1.41**	**1.73**	**1.73**	**1.41**	**1.73**	**1.41**	**1.41**	**1.41**	**1.73**
Linear	PFI (1953s)	1.73	1.73	2.0	2.0	2.24	2.0	2.24	2.24	2.24	2.83
Compas	AFI (0.27s)	**2.0**	**2.0**	**2.65**	**2.65**	**2.83**	**2.83**	**2.83**	**2.83**	**2.83**	**2.83**
RBF	PFI (6827s)	**2.0**	**2.0**	**2.65**	**2.65**	**2.83**	**2.83**	**2.83**	**2.83**	**2.83**	**2.83**
Compas	AFI (0.22s)	4.24	4.24	4.12	4.12	4.24	4.24	4.24	4.24	4.24	4.24
Polynomial	PFI (2069s)	**2.45**	**2.45**	**3.0**	**3.0**	**3.74**	**3.74**	**3.74**	**3.74**	**3.74**	**3.74**
German	AFI (0.01s)	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.41**	**1.73**	**1.41**
Linear	PFI (4.07s)	3.16	3.46	3.16	3.16	3.16	3.16	3.16	3.6	3.74	3.0
German	AFI (0.02s)	**1.73**	**1.0**	**1.73**	**1.73**	**2.0**	**1.41**	**1.73**	**1.73**	**2.0**	**2.24**
RBF	PFI (6.23s)	4.0	3.46	4.24	4.24	4.36	3.61	4.24	4.24	4.36	4.47
German	AFI (0.01s)	**4.90**	**4.12**	**4.47**	**3.87**	**3.87**	**4.24**	**3.46**	**3.46**	**3.46**	**3.46**
Polynomial	PFI (4.15s)	5.74	5.10	5.74	4.69	4.69	5.0	4.58	4.58	4.58	4.58

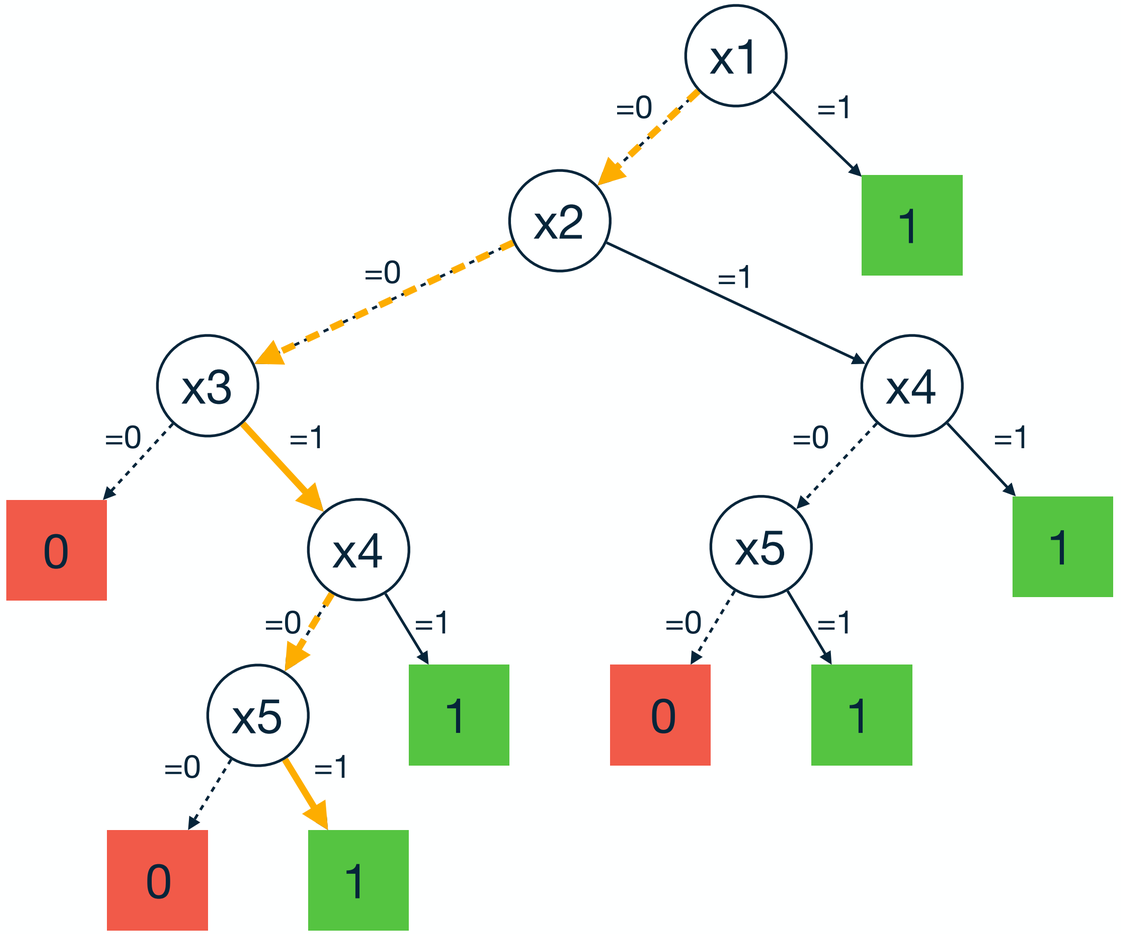
AFI vs LIME

Distance between LIME and ...	**Adult**			**Compas**			**German**		
	Lin.	RBF	Poly	Lin.	RBF	Poly	Lin.	RBF	Poly
AFI ($\epsilon = 0.1$)	2.42	2.04	2.98	1.67	1.06	3.05	2.62	2.03	**5.31**
AFI ($\epsilon = 0.2$)	1.68	1.32	2.67	1.63	0.17	2.73	2.21	2.00	5.41
AFI ($\epsilon = 0.3$)	1.39	0.51	2.58	**1.57**	0.14	**2.62**	1.92	2.05	5.45
AFI (Global)	**1.37**	**0.01**	**1.01**	**1.57**	**0.13**	3.16	**1.90**	**1.89**	5.53

Logic-Based Explainability

Abductive Explanations (AXp) [Marques-Silva21]

Subset-Minimal Set of Input Features Sufficient for Ensuring Prediction

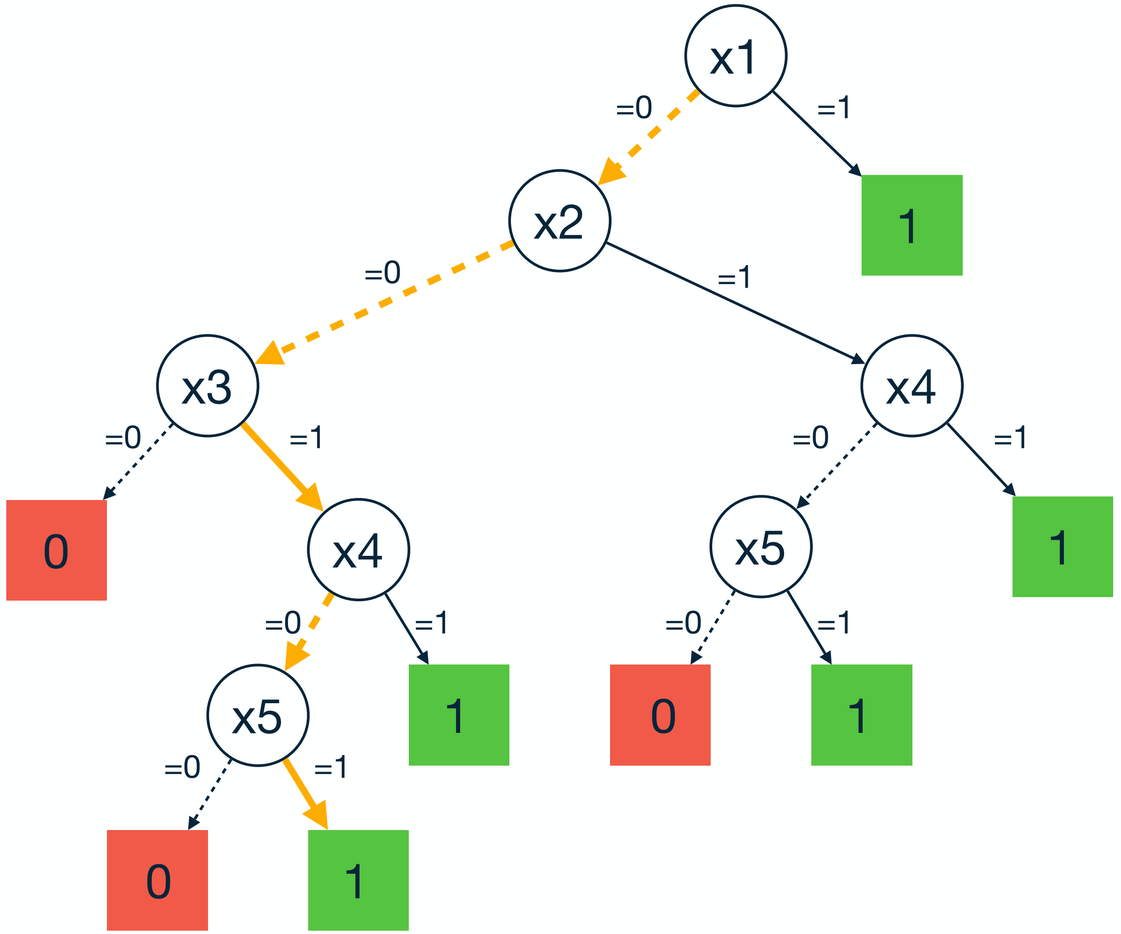


AXp = { 3, 5 }

x3	x5	x1	x2	x4		
1	1	0	0	0	→	1
1	1	0	0	1	→	1
1	1	0	1	0	→	1
1	1	0	1	1	→	1
1	1	1	0	0	→	1
1	1	1	0	1	→	1
1	1	1	1	0	→	1
1	1	1	1	1	→	1

Computing One AXp [Marques-Silva21]

Drop (i.e., Free) Input Features While AXp Condition Holds



{ 1, 2, 3, 4, 5 } → 1

Free 1: { 2, 3, 4, 5 } → 1

Free 2: { 3, 4, 5 } → 1

Free 3: { 4, 5 } →

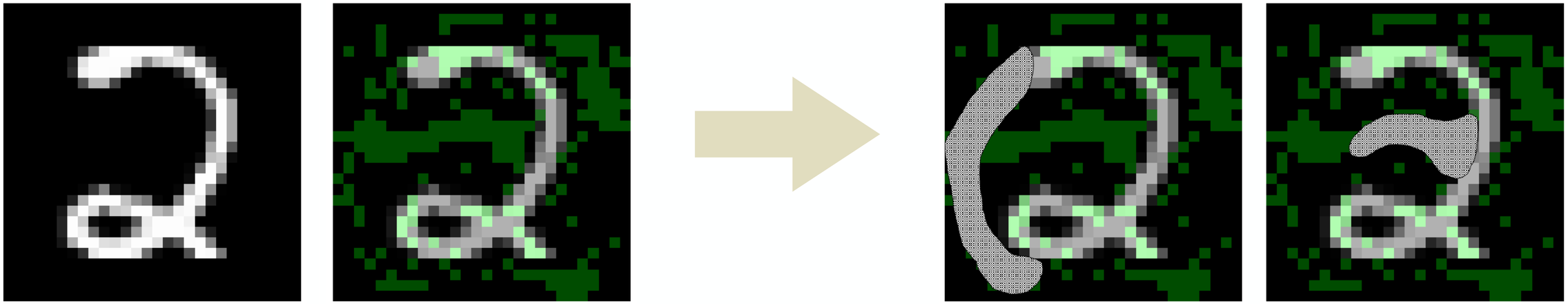
Free 4: { 3, 5 } → 1

Free 5: { 3 } →

AXp = { 3, 5 }

VeriX [Wu23]

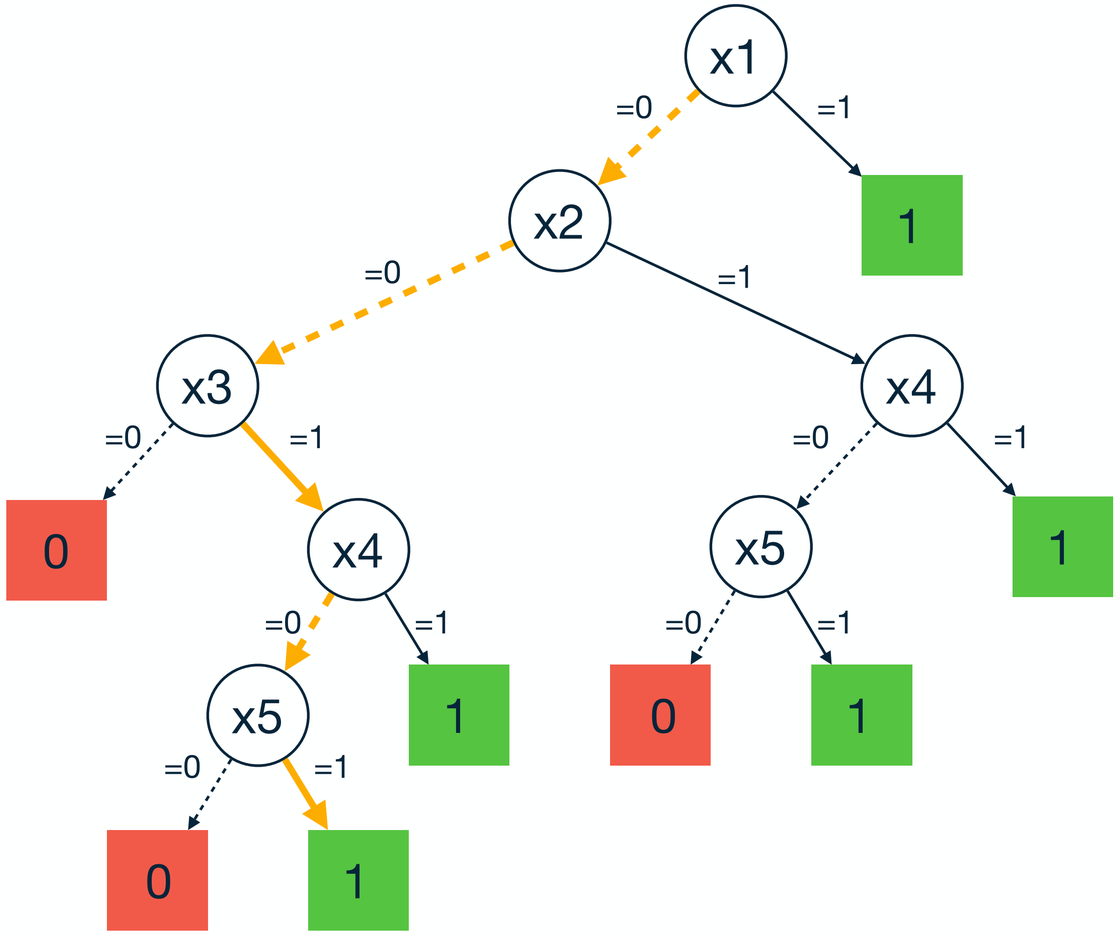
Distance-Restricted Abductive Explanations



(a) Original “2” (c) VERIX (e) “2” into “0” (f) “2” into “3”

Contrastive Explanations (CXp) [Marques-Silva21]

Subset-Minimal Set of Input Features Sufficient for Changing Prediction

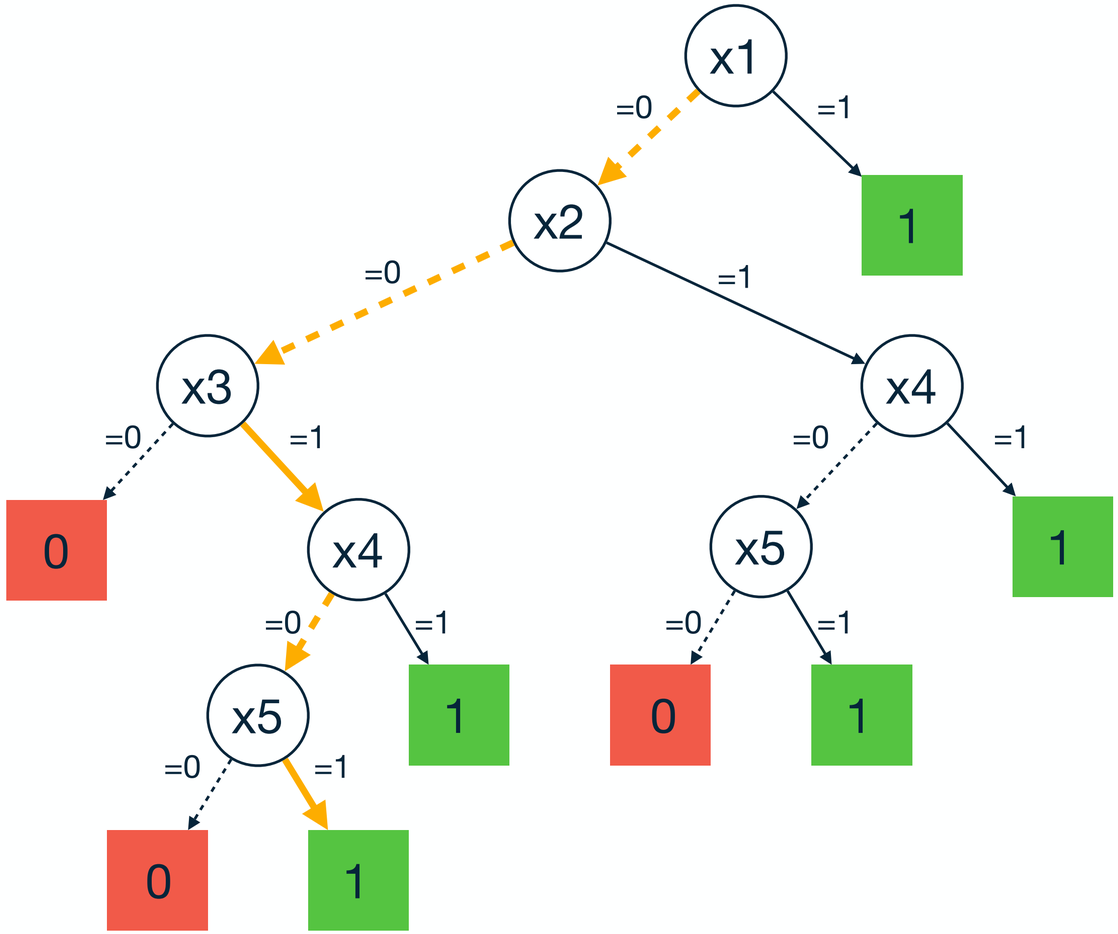


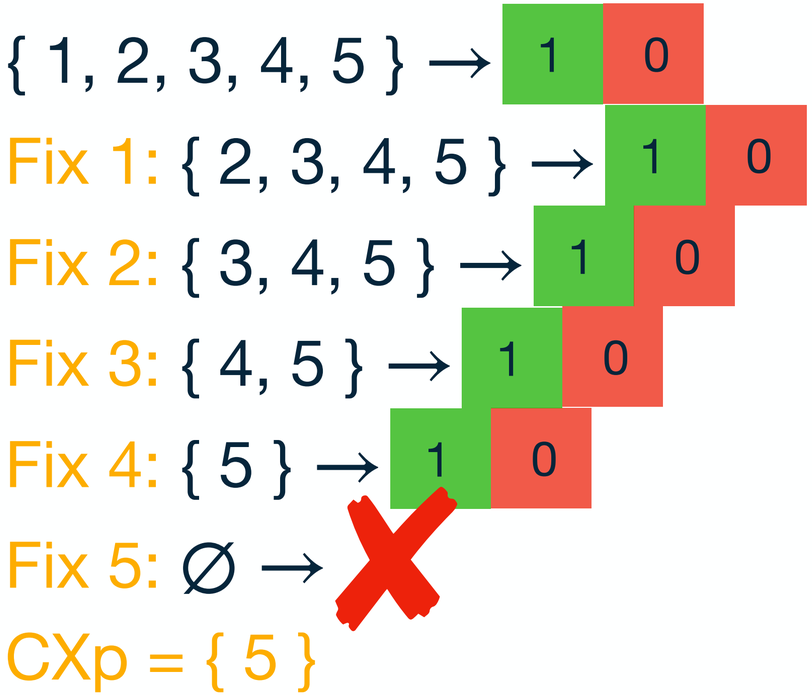
CXp = { 3 }

CXp = { 5 }

Computing One CXp [Marques-Silva21]

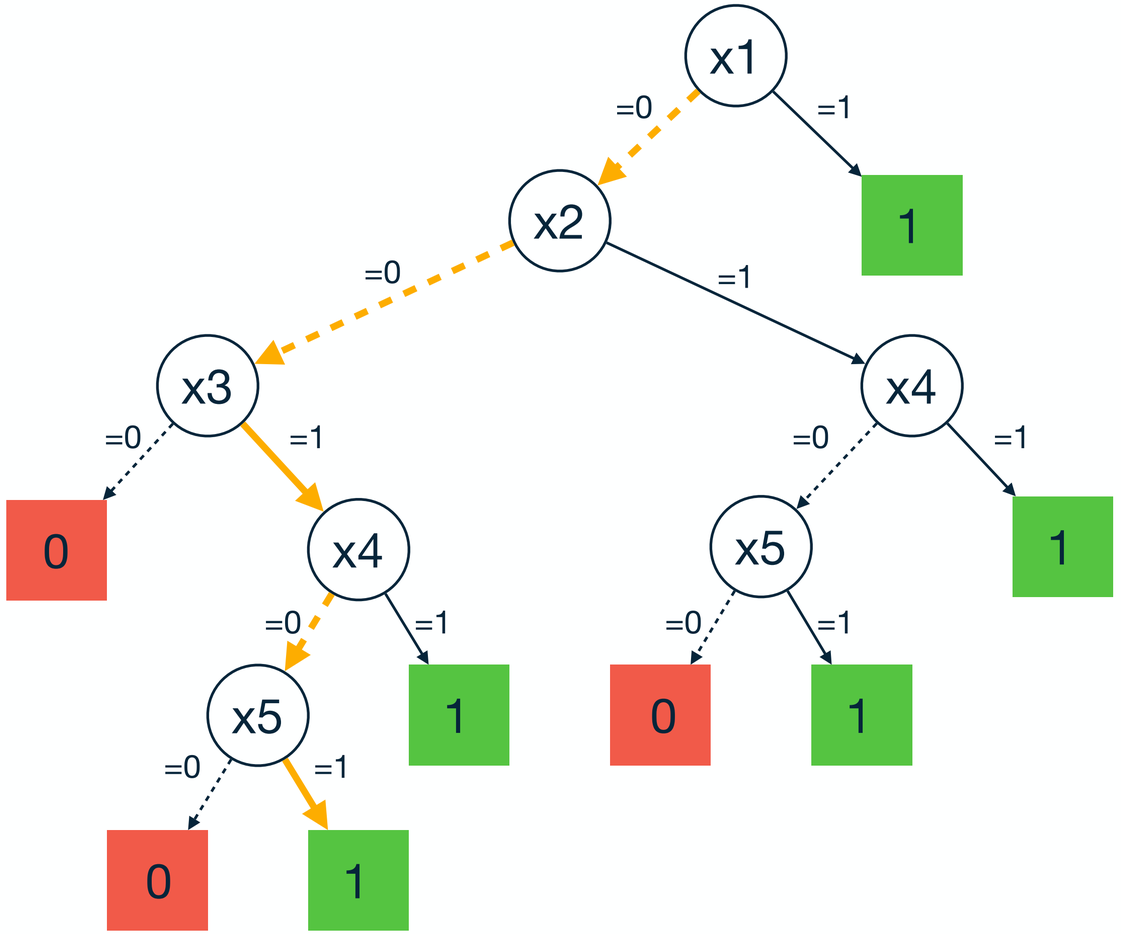
Drop (i.e., Fix) Input Features While CXp Condition Holds





Computing One CXp [Marques-Silva21]

Drop (i.e., Fix) Input Features While CXp Condition Holds



{ 1, 2, 3, 4, 5 } → 1 0

Fix 5: { 1, 2, 3, 4 } → 1 0

Fix 4: { 1, 2, 3 } → 1 0

Fix 3: { 1, 2 } → ✗

Fix 2: { 1, 3 } → 1 0

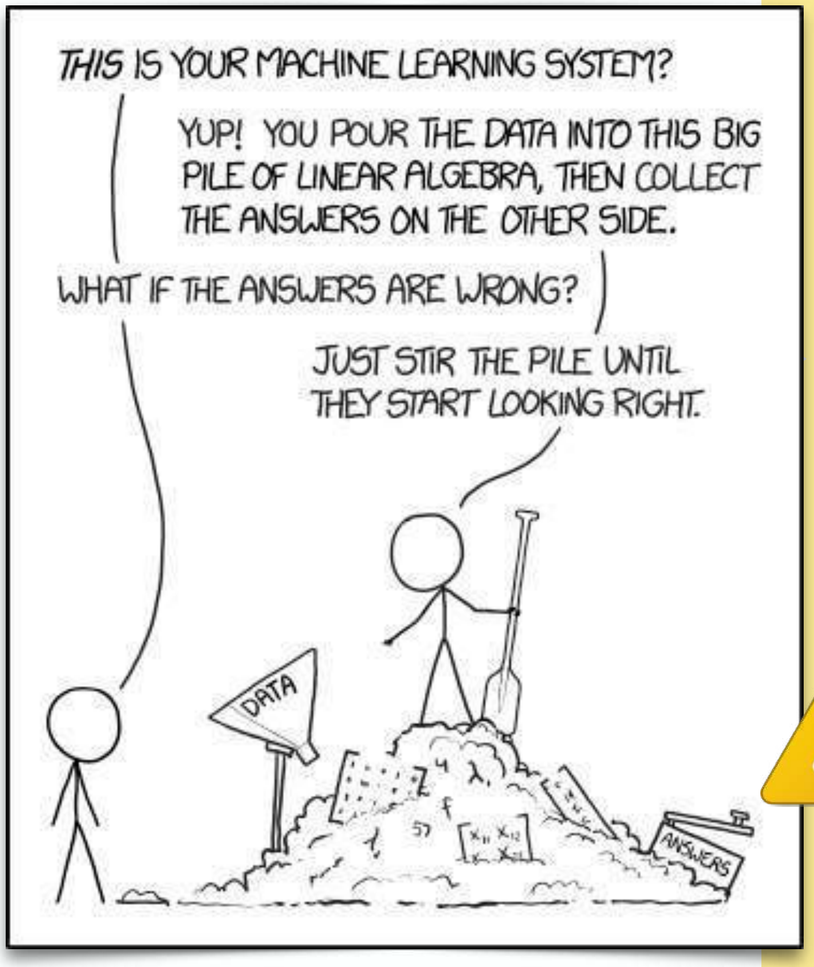
Fix 1: { 3 } → 1 0

CXp = { 3 }

Static Analysis for Model Training

Machine Learning Development Pipeline

Model Training is Highly Non-Deterministic



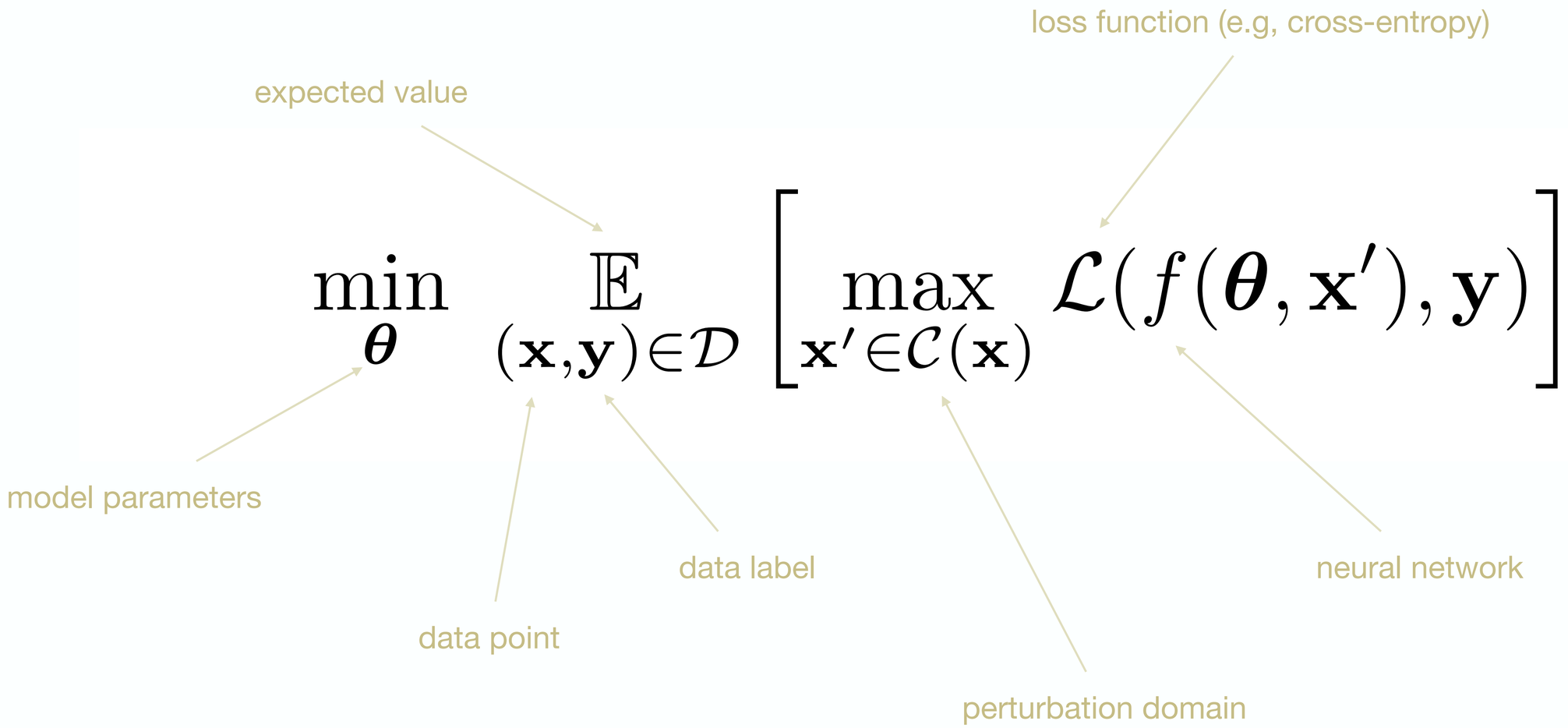
model training → model deployment → predictions

no predictability and traceability

Robust Training

Robust Training

Minimizing the Worst-Case Loss for Each Input

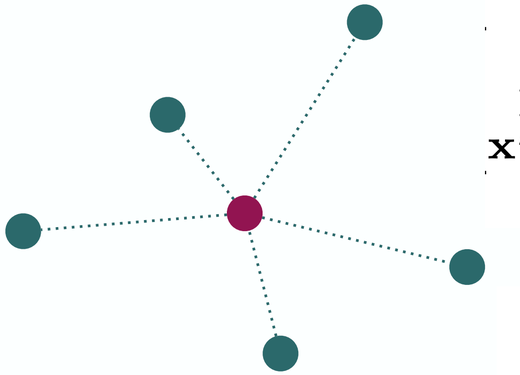


Robust Training

Minimizing the Worst-Case Loss for Each Input

Adversarial Training

Minimizing a Lower-Bound on the Worst-Case Loss



$$\max_{\mathbf{x}' \in \mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}'), \mathbf{y})$$

VI

$$\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y)$$

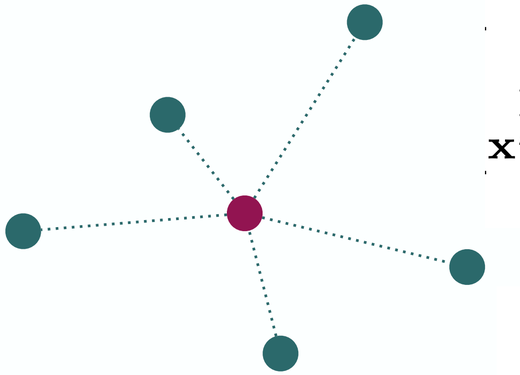
Machine Learning Community

Robust Training

Minimizing the Worst-Case Loss for Each Input

Adversarial Training

Minimizing a Lower-Bound on the Worst-Case Loss



$$\max_{\mathbf{x}' \in \mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}'), \mathbf{y})$$

VI

$$\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y)$$

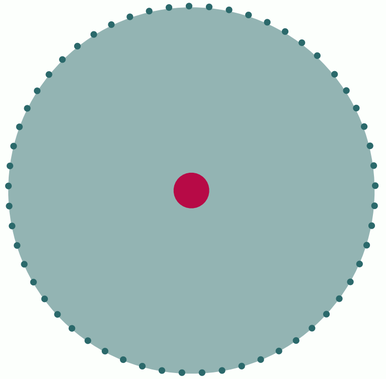
Certified Training

Minimizing an Upper-Bound on the Worst-Case Loss

$$\max_{\mathbf{x}' \in \mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}'), \mathbf{y})$$

AI

$$\mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$



Machine Learning Community

Formal Methods Community

Certified Training

- **M. Andriushchenko, and M. Hein**. *Provably Robust Boosted Decision Stumps and Trees Against Adversarial Attacks*. In NeurIPS 2019.
  approach targeting decision trees

- **M. Hein and M. Andriushchenko**. *Formal Guarantees on the Robustness of a Classifier Against Adversarial Manipulation*. In NeurIPS 2017.
  E. Wong and Z. Kolter. *Provable Defenses Against Adversarial Examples via the Convex Outer Adversarial Polytope*. In ICML, 2018.
  A. Raghunathan, J. Steinhardt, and P. Liang. *Certified Defenses against Adversarial Examples*. In ICML, 2018.
  approaches targeting neural networks

- **M. Mirman, T. Gehr, and M. Vechev**. *Differentiable Abstract Interpretation for Provably Robust Neural Networks* In ICML 2018.
  F. Ranzato and M. Zanella. *Genetic Adversarial Training of Decision Trees*. In GECCO 2021.
  abstract interpretation-based approaches

Robust Training

Empirical Robustness

Table 7: Comparison of the standard (Acc.), adversarial (Adv. Acc), and certified (Cert. Acc.) accuracy for different certified training methods on the full CIFAR-10 test set. We use MN-BAB (Ferrari et al., 2022) to compute all certified and adversarial accuracies.

ϵ_∞	Training Method	Source	Acc. [%]	Adv. Acc. [%]	Cert. Acc. [%]
2/255	COLT	Balunovic & Vechev (2020)	78.42	**66.17**	61.02
	CROWN-IBP	Zhang et al. (2020)†	71.27	59.58	58.19
	IBP	Shi et al. (2021)	-	-	-
	SABR	this work	**79.52**	65.76	**62.57**
8/255	COLT	Balunovic & Vechev (2020)	51.69	31.81	27.60
	CROWN-IBP	Zhang et al. (2020)†	45.41	33.33	33.18
	IBP	Shi et al. (2021)	48.94	35.43	**35.30**
	SABR	this work	**52.00**	**35.70**	35.25

RobustBench Leaderboards Paper FAQ Contribute Model Zoo

Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, untargeted attack

Show 15 entries Search: Papers, architectures, v

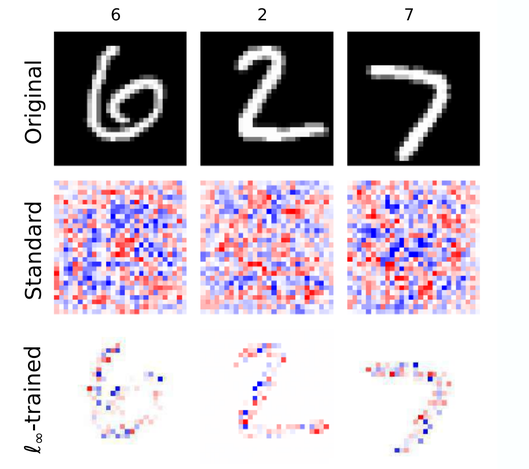
Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
	Robust Principles: Architectural Design Principles for Adversarially Robust CNNs *It uses additional 50M synthetic images in training.*	93.27%	71.07%	71.07%	×	×	RaWideResNet-70-16	BMVC 2023

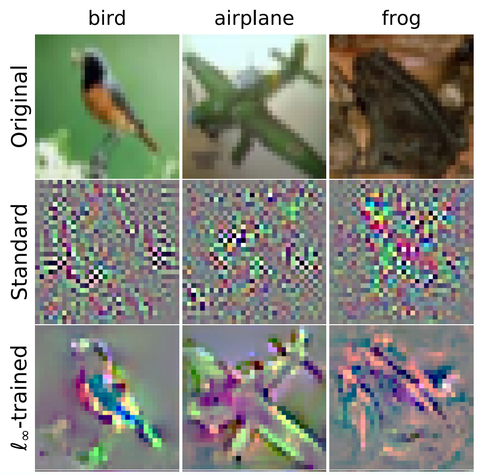
Machine Learning Community

Formal Methods Community

Robust Training

Perceptually Aligned Gradients





Adversarial Training

Fig. 6. Input Image

Fig. 7. Integrated Gradients

Certified Training

Robust Training

Minimizing the Worst-Case Loss for Each Input

Adversarial Training

Minimizing a Lower-Bound on the Worst-Case Loss

Certified Training

Minimizing an Upper-Bound on the Worst-Case Loss

Hybrid Training

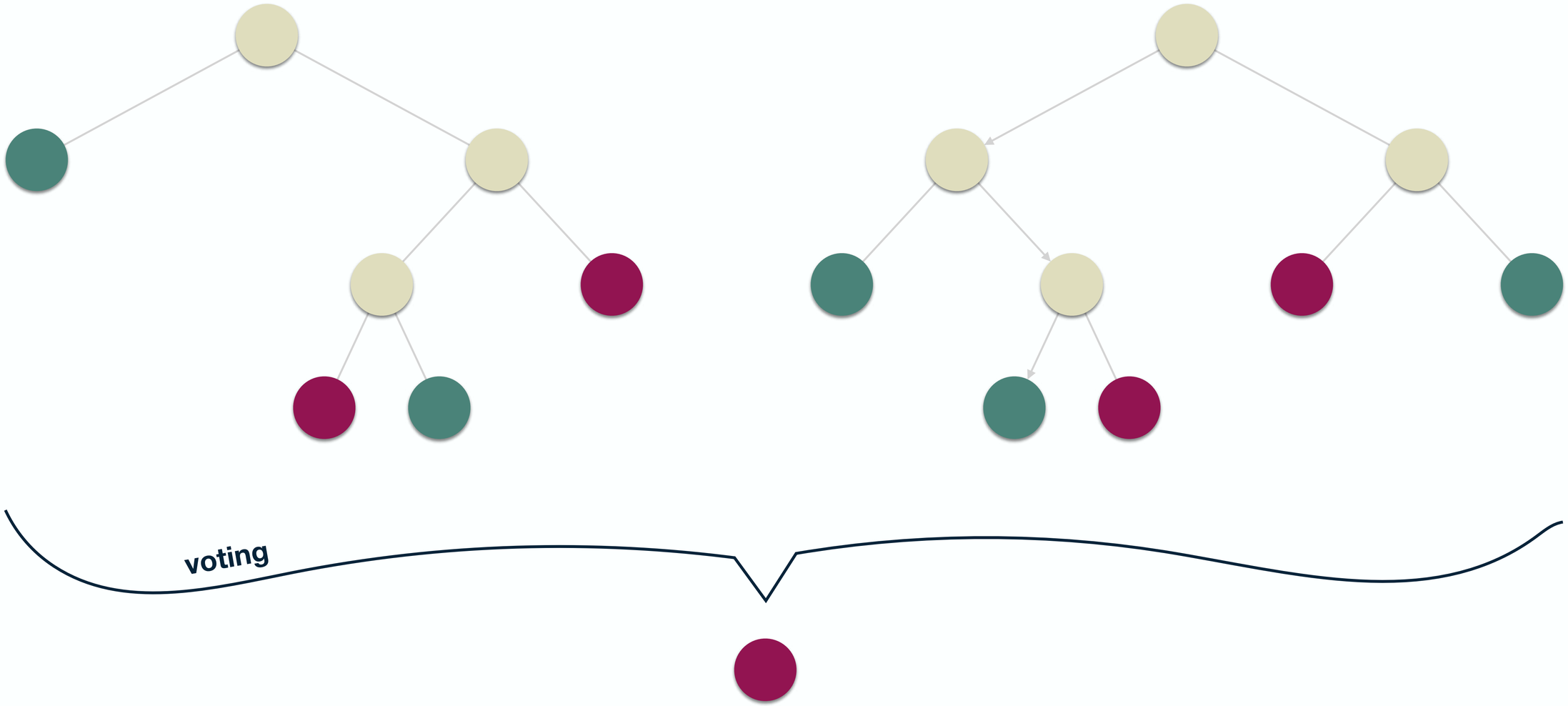
$$(1-\alpha)\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}_{\text{adv}}), y) + \alpha\ \mathcal{L}_{\text{ver}}(f(\boldsymbol{\theta}, \mathbf{x}), y)$$

Machine Learning Community

Formal Methods Community

Random Forests

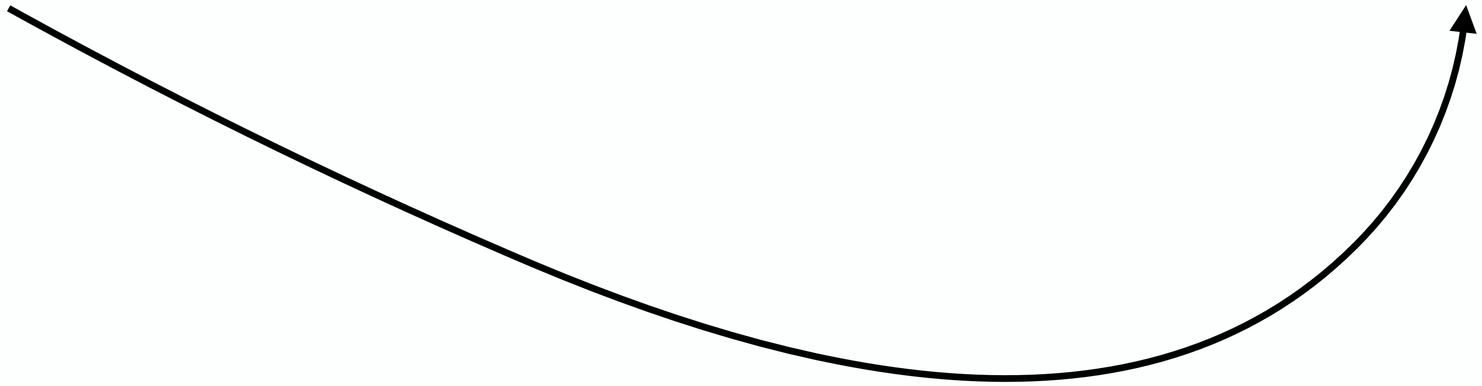
Random Forests



Hybrid Training [Ranzato21]

Random Forests

Dataset	FATT			Natural CART			CART with Hints		
	Accuracy %	Fairness %	Size	Accuracy %	Fairness %	Size	Accuracy %	Fairness %	Size
Adult	80.84	95.21	43	85.32	77.56	270	84.77	87.46	47
Compas	64.11	85.98	75	65.91	22.25	56	65.91	22.25	56
Crime	79.45	75.19	11	77.69	24.31	48	77.44	60.65	8
German	72.00	99.50	2	75.50	57.50	115	73.50	86.00	4
Health	77.87	97.03	84	83.85	79.98	2371	82.25	93.64	100
Average	74.85	**90.58**	**43**	**77.65**	52.32	572	76.77	70.00	**43**



Neural Networks

Hybrid Training

Neural Networks

- **Mark Niklas Müller, Franziska Eckert, Marc Fischer, and Martin Vechev.** Certified training: Small Boxes Are All You Need. In ICLR, 2023.
  one of the first instances of hybrid training

- **Alessandro De Palma, Rudy Bunel, Krishnamurthy Dvijotham, M. Pawan Kumar, Robert Stanforth, Alessio Lomuscio.** Expressive Losses for Verified Robustness via Convex Combinations. In ICLR, 2024.
  characterization of expressive losses for hybrid training

Hybrid Training

Neural Networks

arXiv:2410.01617v1 [cs.LG] 2 Oct 2024

On Using Certified Training towards Empirical Robustness

Alessandro De Palma* *alessandro.de-palma@inria.fr*
Inria, École Normale Supérieure, PSL University, CNRS

Serge Durand* *serge.durand@inria.fr*
Université Paris-Saclay, CEA, List

Zakaria Chihani *zakaria.chihani@cea.fr*
Université Paris-Saclay, CEA, List

François Terrier *francois.terrier@cea.fr*
Université Paris-Saclay, CEA, List

Caterina Urban *caterina.urban@inria.fr*
Inria, École Normale Supérieure, PSL University, CNRS

Abstract

Adversarial training is arguably the most popular way to provide empirical robustness against specific adversarial examples. While variants based on multi-step attacks incur significant computational overhead, single-step variants are vulnerable to a failure mode known as catastrophic overfitting, which hinders their practical utility for large perturbations. A parallel line of work, certified training, has focused on producing networks amenable to formal guarantees of robustness against any possible attack. However, the wide gap between the best-performing empirical and certified defenses has severely limited the applicability of the latter. Inspired by recent developments in certified training, which rely on a combination of adversarial attacks with network over-approximations, and by the connections between local linearity and catastrophic overfitting, we present experimental evidence on the practical utility and limitations of using certified training towards empirical robustness. We show that, when tuned for the purpose, a recent certified training algorithm can prevent catastrophic overfitting on single-step attacks, and that it can bridge the gap to multi-step baselines under appropriate experimental settings. Finally, we present a novel regularizer for network over-approximations that can achieve similar effects while markedly reducing runtime.

1 Introduction

The discovery of adversarial examples (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015), semantic invariant perturbations that induce high-confidence misclassifications in neural networks, has led to the development of a variety of adversarial attacks (Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017) and empirical defenses (Papernot et al., 2016; Cisse et al., 2017; Tramèr et al., 2018). Adversarial training (Madry et al., 2018) is undeniably the most successful empirical defense, owing to its efficacy and conceptual simplicity. Employing a robust optimization perspective, Madry et al. (2018) train using the loss incurred at the point returned by a multi-step attack, named PGD. The large cost of PGD-based adversarial training ushered in the development of less expensive techniques (Shafahi et al., 2019). However, single-step attacks (Goodfellow et al., 2015), were shown to suffer from a phenomenon known as Catastrophic Overfitting (CO) (Wong et al., 2020), under which they display a vulnerability to multi-step attacks whilst preserving strong robustness to single-step attacks. It was shown that CO can be mitigated at no additional cost through strong noise (de Jorge et al., 2022) for smaller perturbations, yet stronger attack models require the addition of explicit regularizers based on local linearity (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024). Indeed, strong links between lack of local linearity and CO exist (Ortiz-Jimenez et al., 2023).

Empirical Robustness

hybrid training yields better empirical robustness at smaller training cost

Bibliography

[Kurd03] **Zeshan Kurd, Tim Kelly**. Establishing Safety Criteria for Artificial Neural Networks. In KES, pages 63-169, 2003.

[Li19] **Jianlin Li, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang**. Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification. In SAS, page 296–319, 2019.
symbolic abstraction

[Singh19] **Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev**. An Abstract Domain for Certifying Neural Networks. In POPL, pages 41:1 - 41:30, 2019.
deeppoly abstraction

[Munakata23] **Satoshi Munakata, Caterina Urban, Haruki Yokoyama, Koji Yamamoto, Kazuki Munakata**. Verifying Attention Robustness of Deep Neural Networks against Semantic Perturbations. In NFM, pages 37-61, 2023.
saliency map stability

Bibliography

[Mohapatra20] **Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, Luca Daniel**. Towards Verifying Robustness of Neural Networks Against A Family of Semantic Perturbations. In CVPR, pages 241-249, 2020.

[Julian16] **Kyle D. Julian, Jessica Lopez, Jeffrey S. Brush, Michael P. Owen, Mykel J. Kochenderfer**. Policy Compression for Aircraft Collision Avoidance Systems. In DASC, pages 1–10, 2016.

[Katz17] **Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, Mykel J. Kochenderfer**. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In CAV, pages 97-117, 2017.

[Mazzucato21] **Denis Mazzucato and Caterina Urban**. Reduced Products of Abstract Domains for Fairness Certification of Neural Networks. In SAS, 2021.
product abstraction

Bibliography

[Urban20] **Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang**. Perfectly Parallel Fairness Certification of Neural Networks. In OOPSLA, pages 185:1–185:30, 2020.
hypersafety verification

[Galhotra17] **Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou**. Fairness Testing: Testing Software for Discrimination. In FSE, pages 498–510, 2017.
dependency fairness

[Urban21] **Caterina Urban and Antoine Miné**. A Review of Formal Methods applied to Machine Learning. https://arxiv.org/abs/2104.02466, 2021.
survey on formal methods for machine learning

[Pal24] **Abhinandan Pal, Francesco Ranzato, Caterina Urban, and Marco Zanella**. Abstract Interpretation-Based Feature Importance for Support Vector Machines. In VMCAI, pages 27-49, 2024.
abstract feature importance

Bibliography

[Marques-Silva21] **João Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska**. Explanations for Monotonic Classifiers. In ICML, pages 7469-7479, 2021.
[Wu23] **Min Wu, Haoze Wu, Clark W. Barrett**. VeriX: Towards Verified Explainability of Deep Neural Networks. In NeurIPS, 2023.
logic-based explanations

[Ranzato21] **Francesco Ranzato, Caterina Urban, and Marco Zanella**. Fairness-Aware Training of Decision Trees by Abstract Interpretation. In CIKM, pages 1508-1517, 2021.
hybrid training for random forests