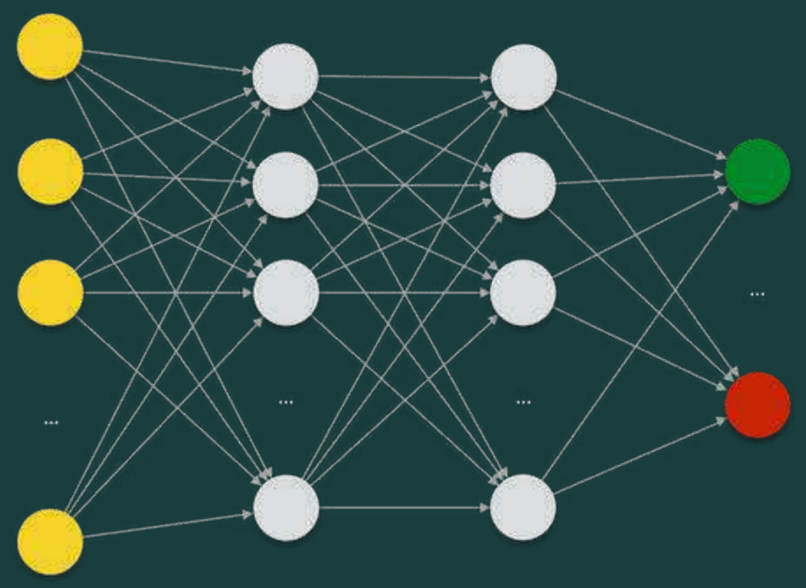
Formal Methods for Machine Learning Pipelines

VTSA 2024



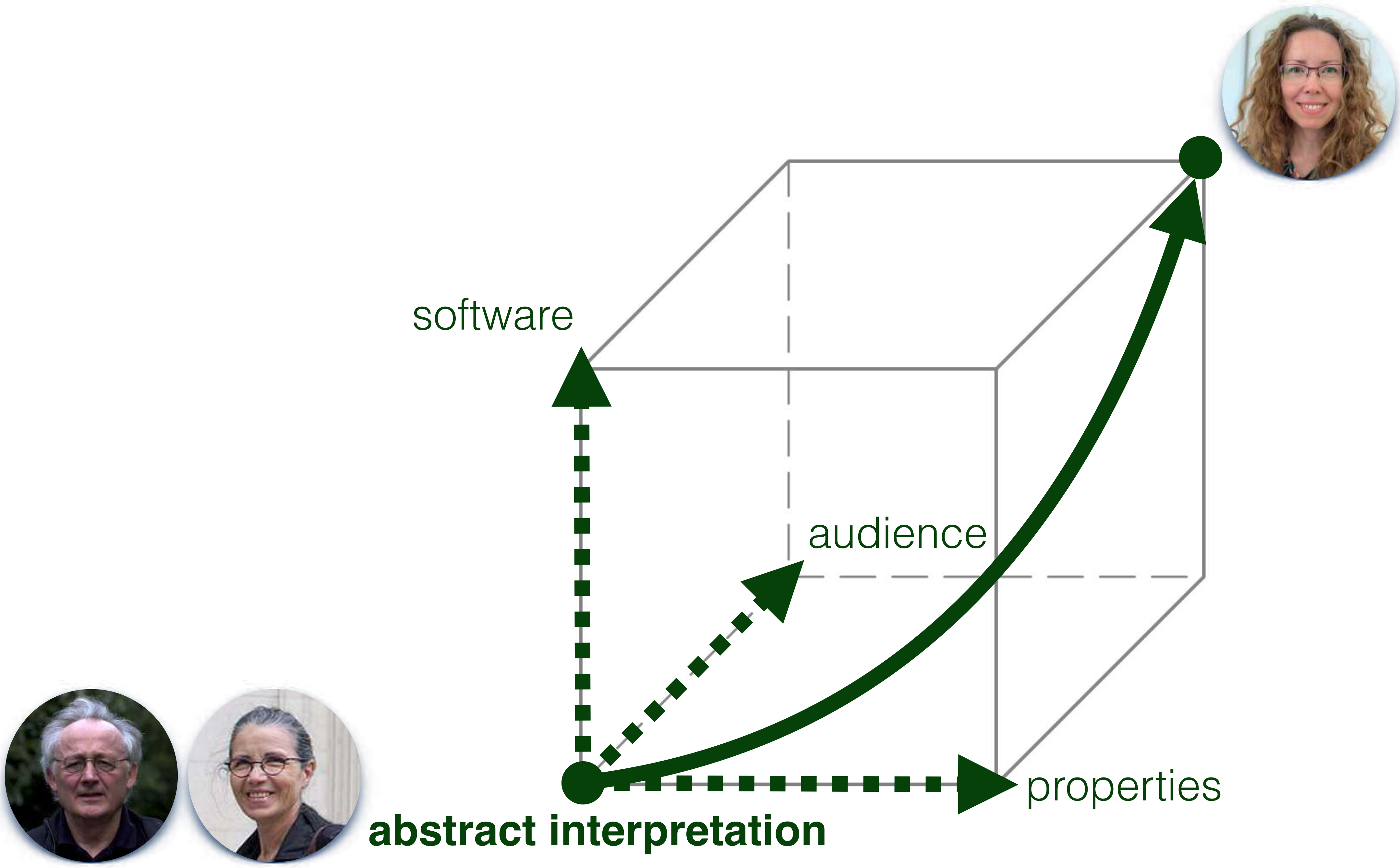
Caterina Urban

July 11th-12th, 2024

Who am I?

1987	Udine, Italie	
2006 - 2011	Università degli Studi di Udine	BSc, MSc
2011 - 2015	École Normale Supérieure	PhD
2015	NASA & Carnegie Mellon University	Internship
2015 - 2019	ETH Zurich	Postdoc
Since 2019	Inria	

What do I do?

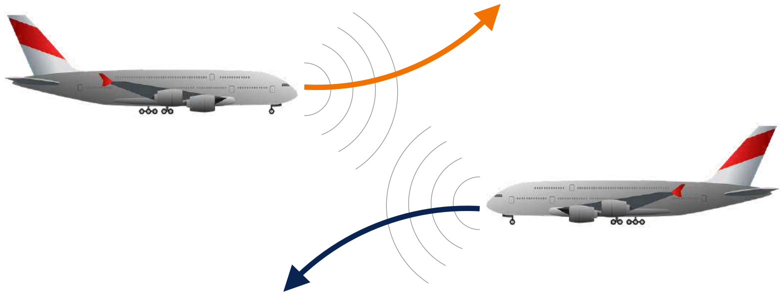


ML in High-Stakes Applications

data

ML software

perform tasks that are impossible to program explicitly



act as surrogate model

automate decision-making

ML in High-Stakes Applications

Machine Learning Pipeline

Machine Learning Development Process

Machine Learning Pipeline

Data Preparation is **Fragile**

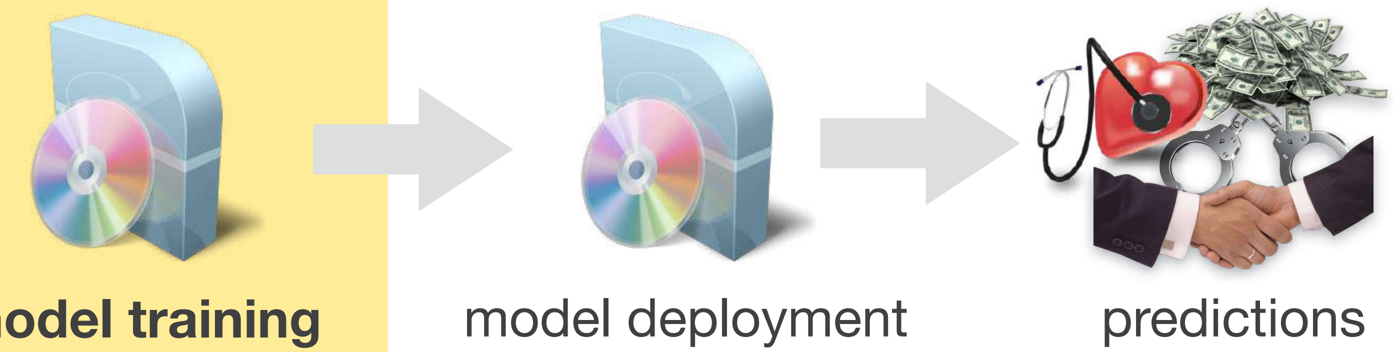
insidious silent bugs

https://xkcd.com/2054/

Machine Learning Pipeline

Model Training is Highly Non-Deterministic

https://xkcd.com/1838/



no predictability and traceability

Machine Learning Pipeline

Models Only Give Probabilistic Guarantees

data

https://xkcd.com/2451/

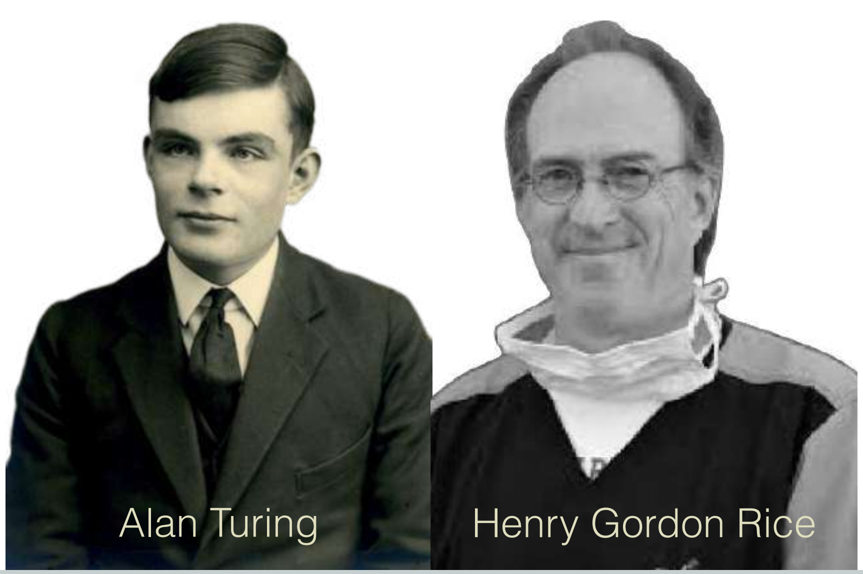
model deployment

predictions

not sufficient for guaranteeing an acceptable failure rate under any circumstance

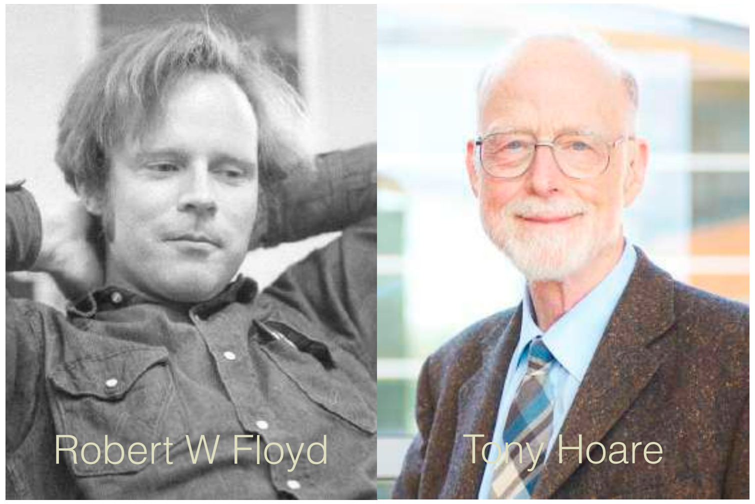
Correctness Guarantees

A Mathematically Proven Hard Problem

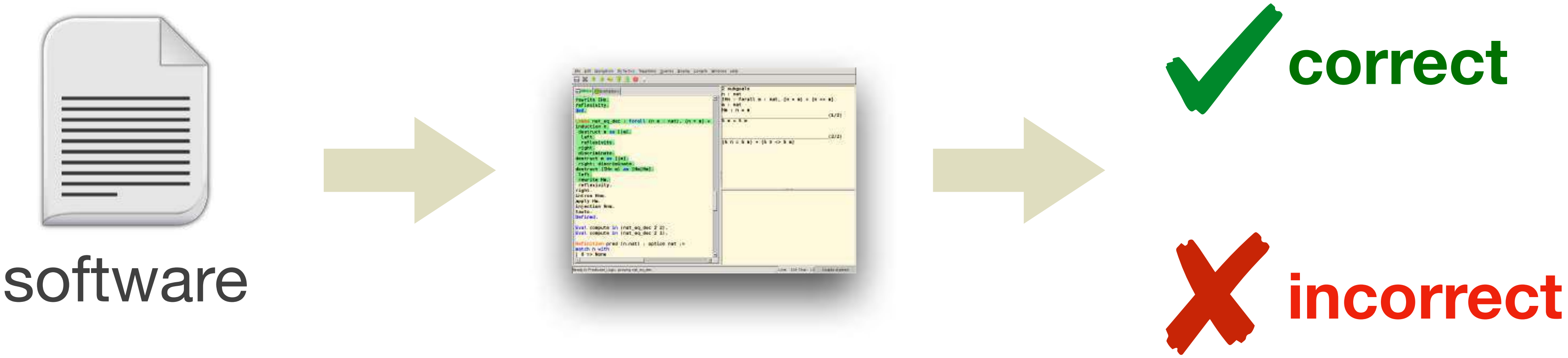


Formal Methods

Deductive Verification

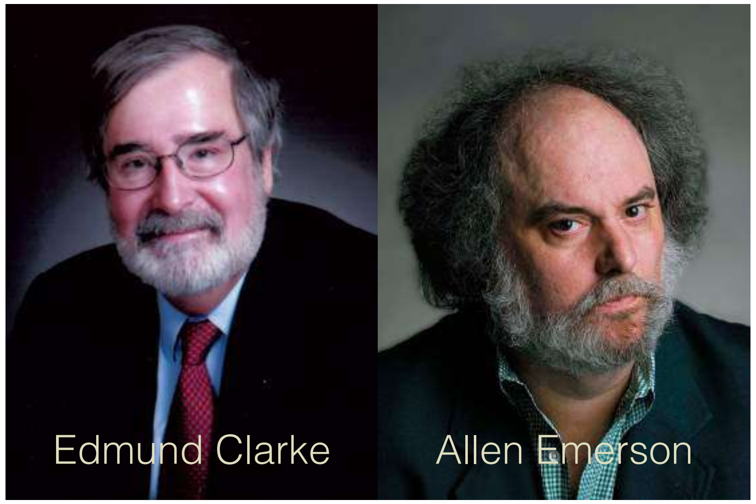


- extremely **expressive**
- **relies on the user** to guide the proof



Formal Methods

Model Checking



- analysis of a **model** of the software
- **sound and complete with respect to the model**

model

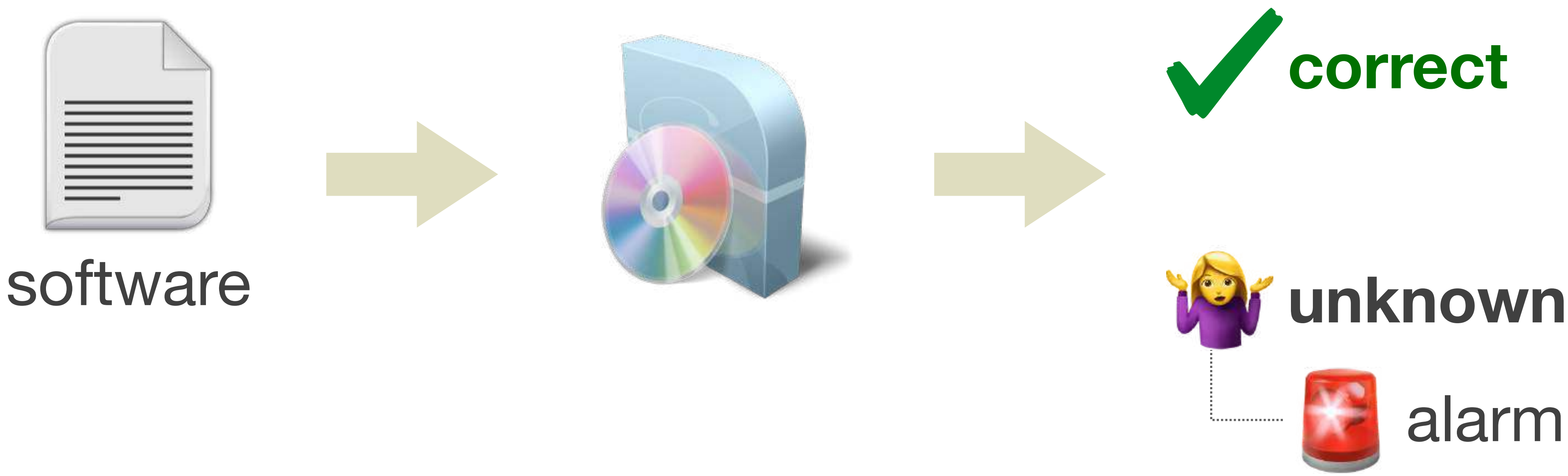
correct

incorrect

Formal Methods

Static Analysis by Abstract Interpretation

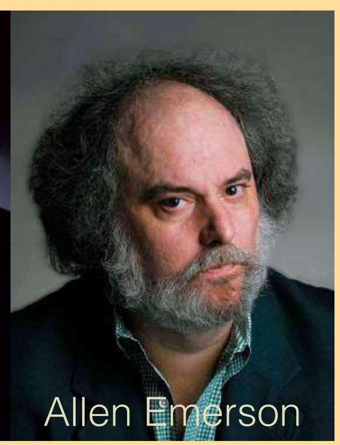
- analysis of the **source or object code**
- fully **automatic** and **sound** by construction
- generally **not complete**




FORMAL METHODS COMMUNITY
MACHINE LEARNING SOFTWARE
TRADITIONAL SAFETY-CRITICAL SOFTWARE

Formal Methods for ML

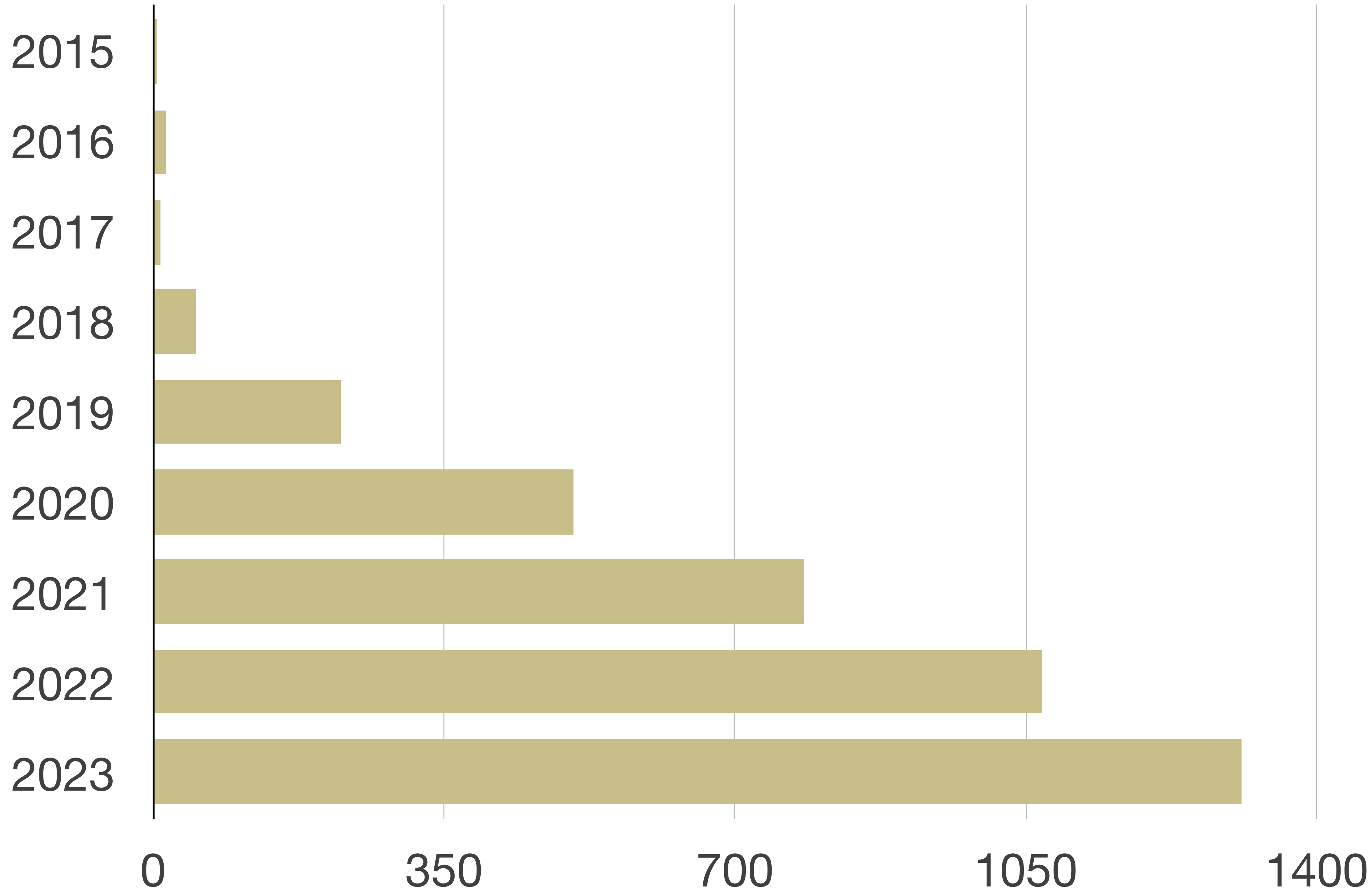
Deductive Verification



Model Checking

Static Analysis

Formal Methods for ML



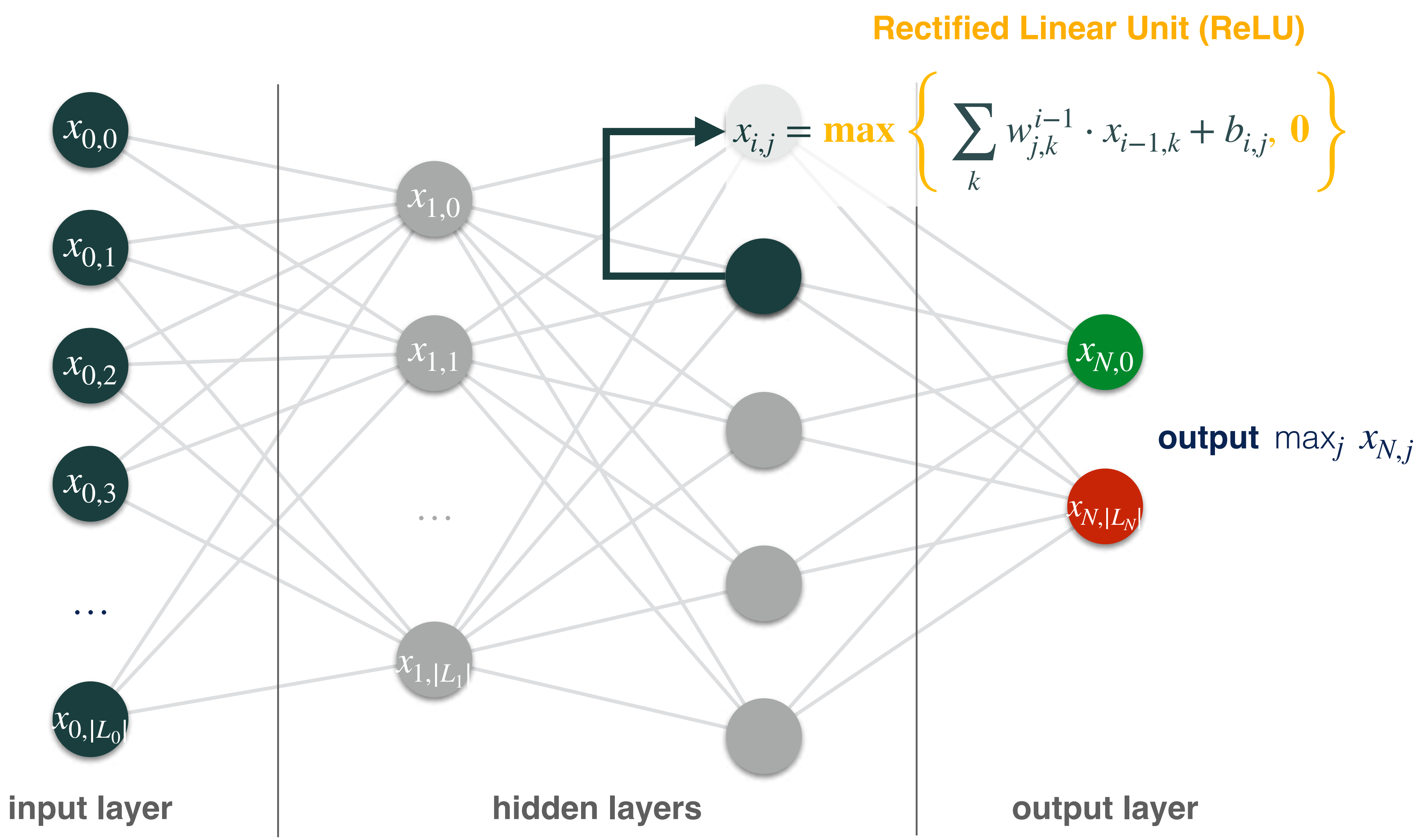
Results for “neural network robustness” on Google Scholar

Formal Methods for Trained Models

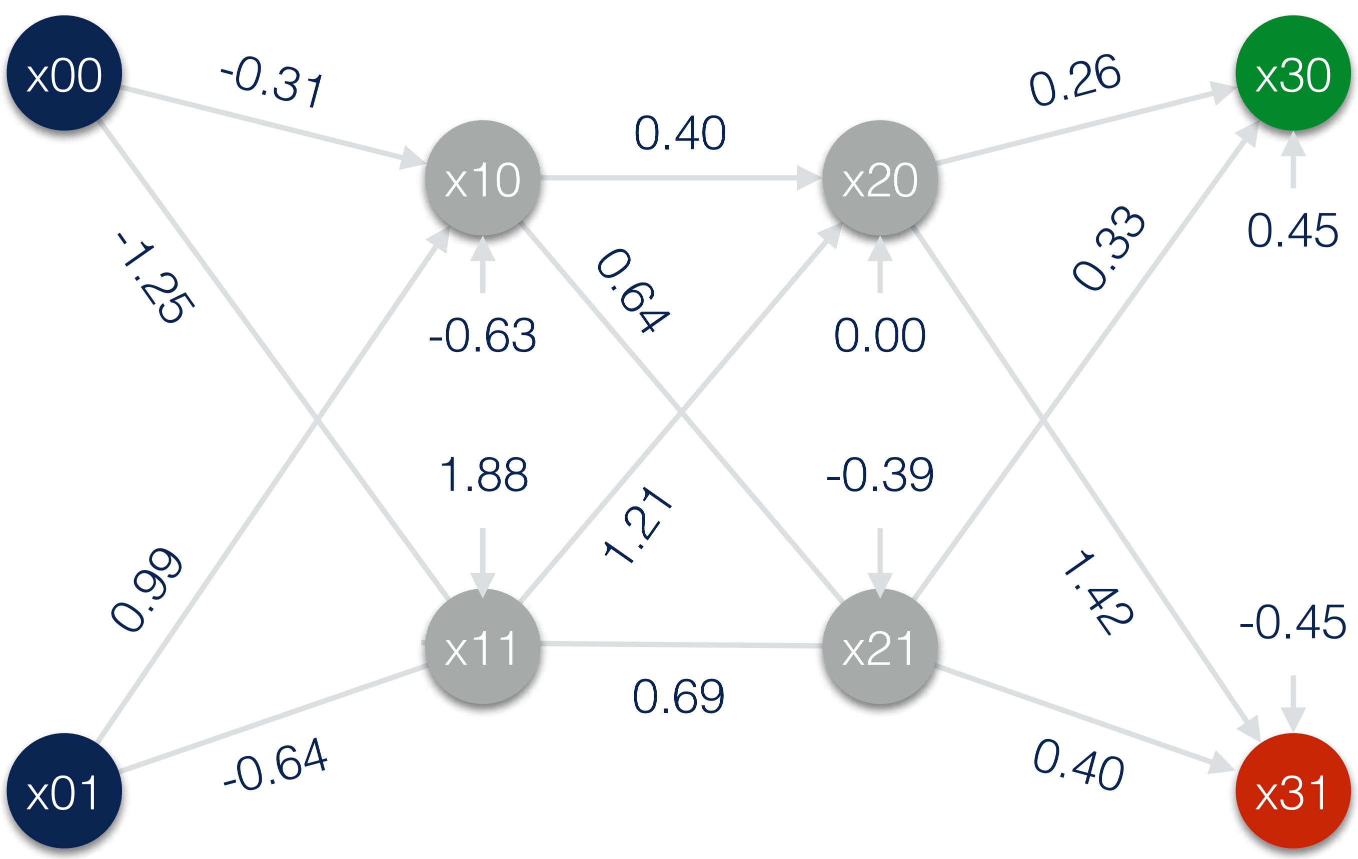
Neural Networks

Neural Networks

Feed-Forward **ReLU-Activated** Neural Networks

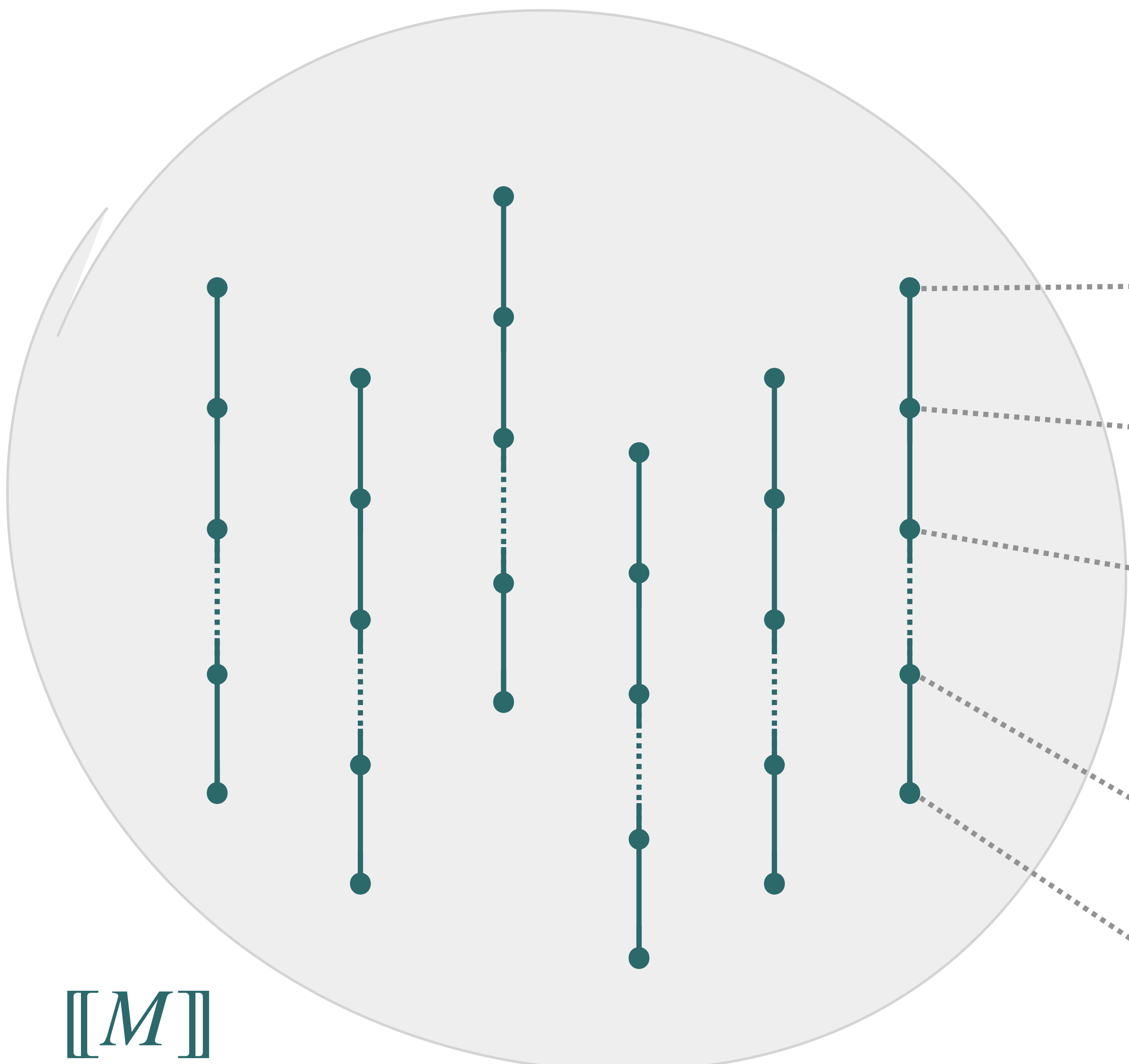


Neural Networks as Programs




```
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

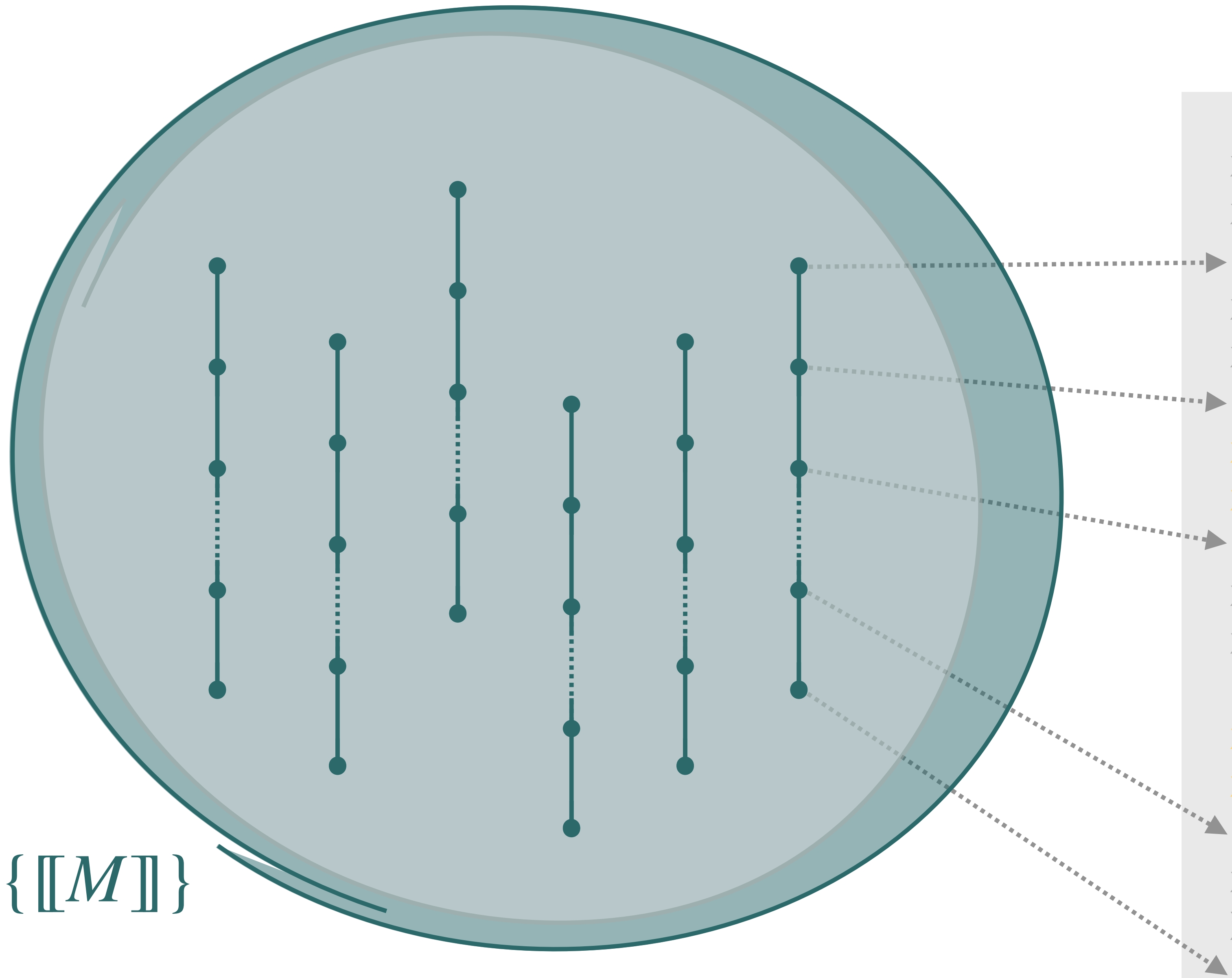


M

```
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

```
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 ‘ ’
```

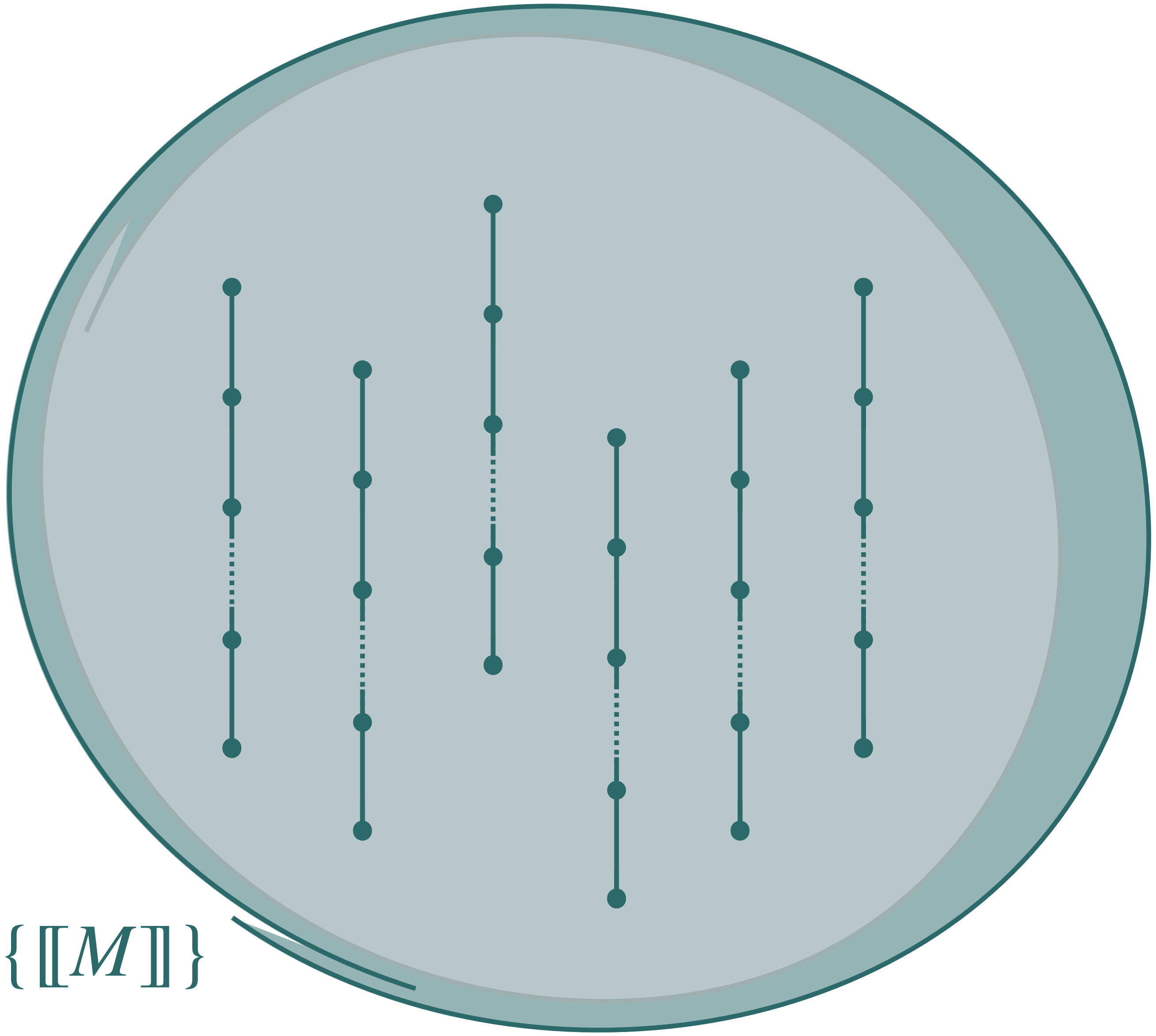
Collecting Semantics

Intuition

Property (by extension): set of elements that have that property

Property "being Jun Pang"

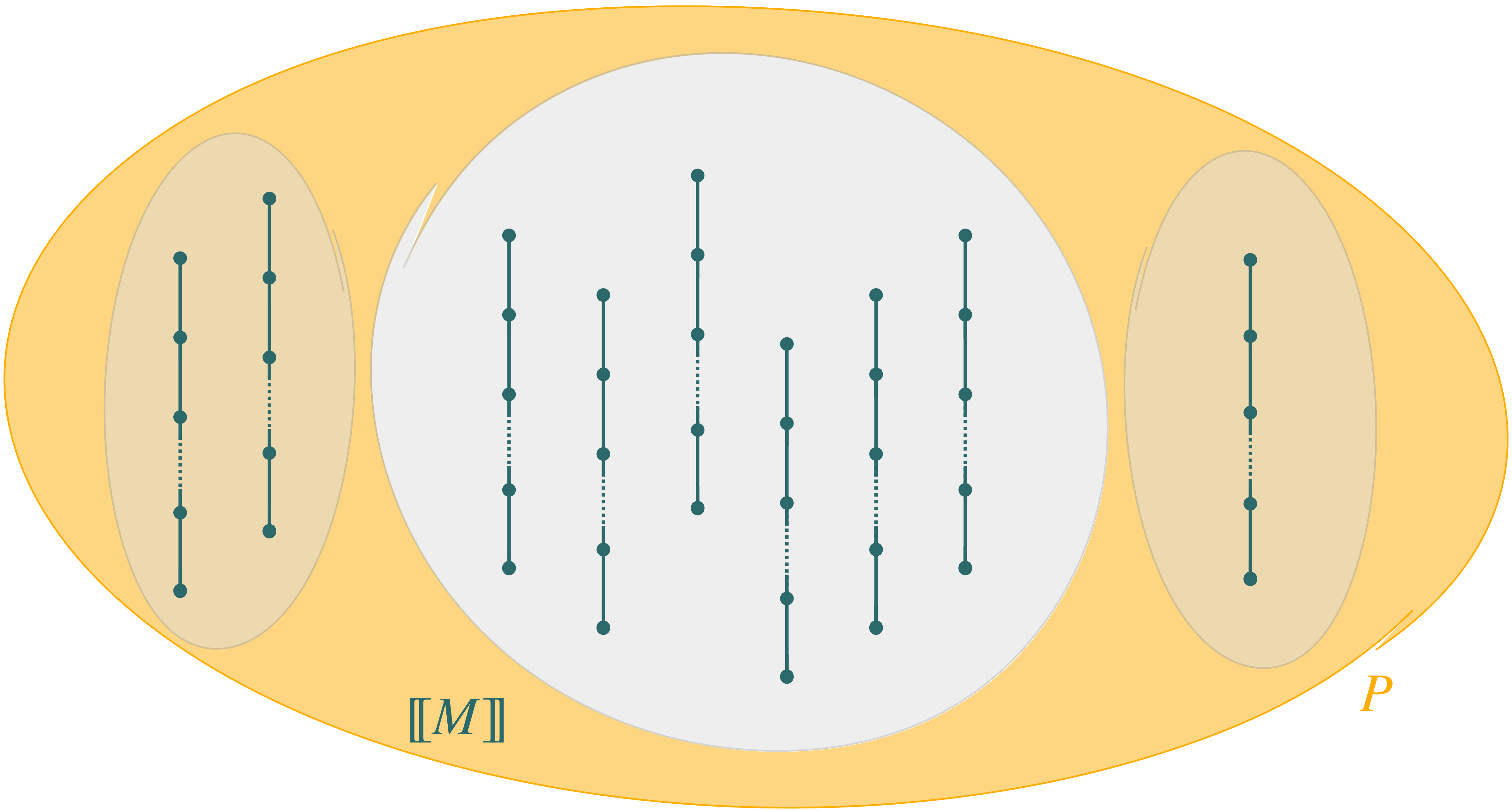
Property "being neural network M"



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

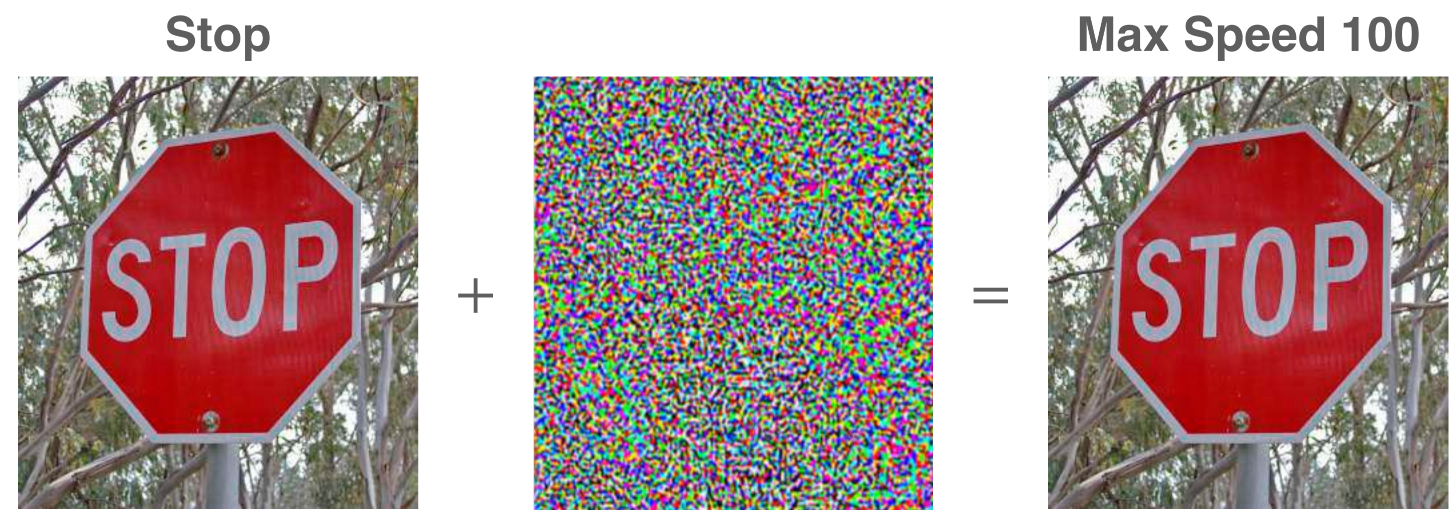
Property Verification

$$\mathscr{M} \in P \Leftrightarrow \{\mathscr{M}\} \subseteq P$$



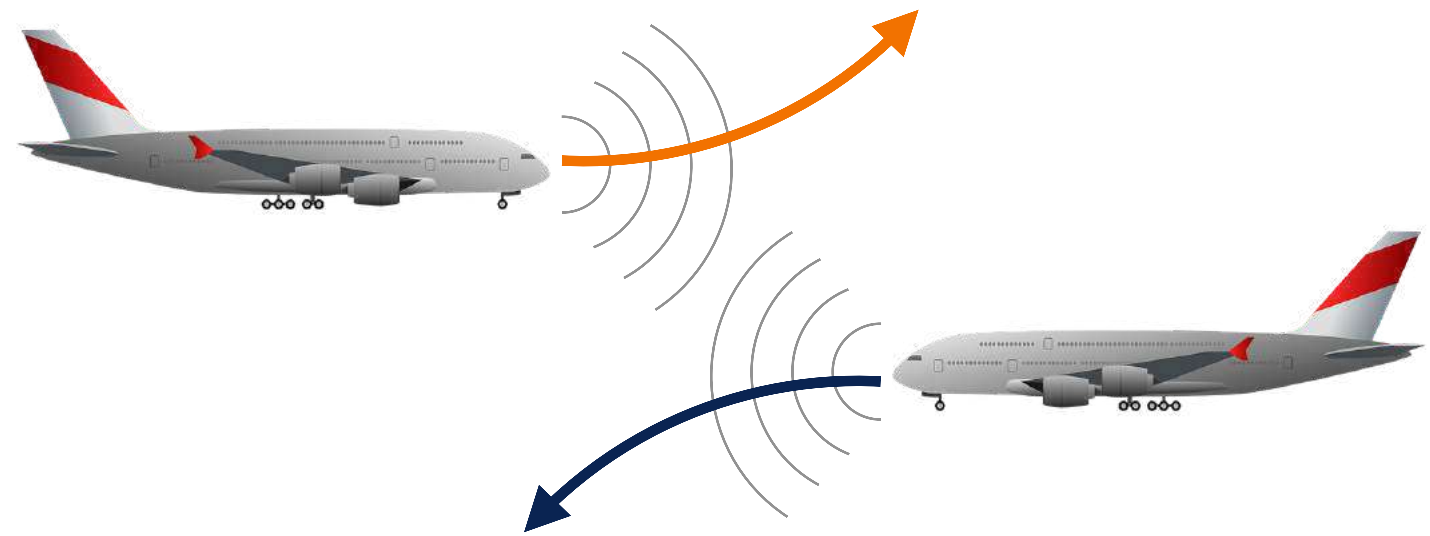
Stability

Goal G3 in [Kurd03]



Safety

Goal G4 in [Kurd03]



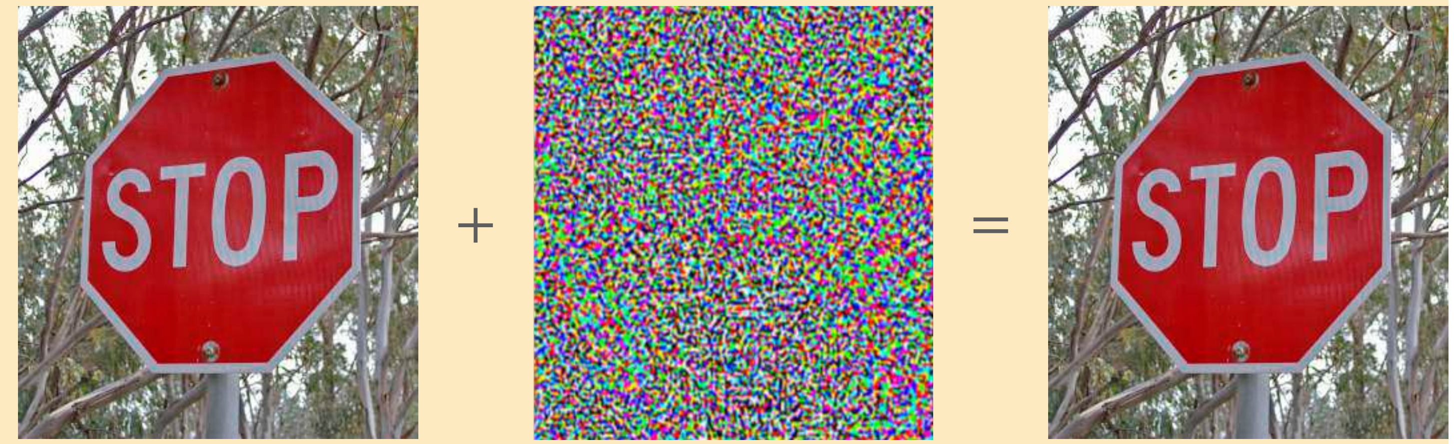
Fairness

Stability

Goal G3 in [Kurd03]

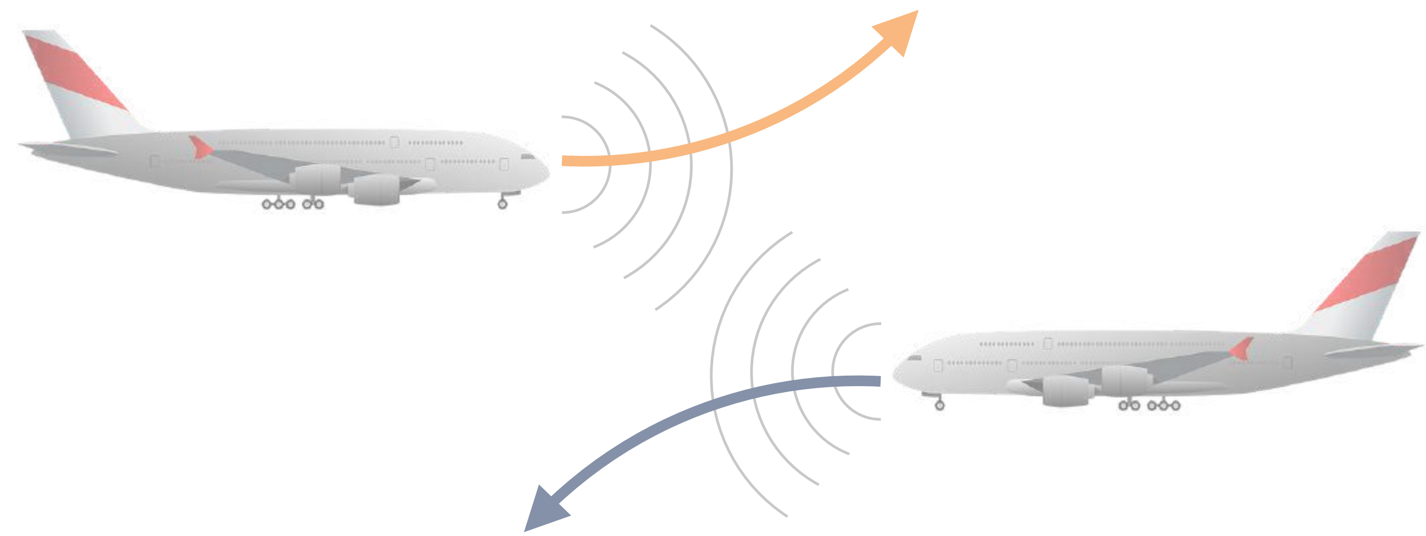
Stop

Max Speed 100

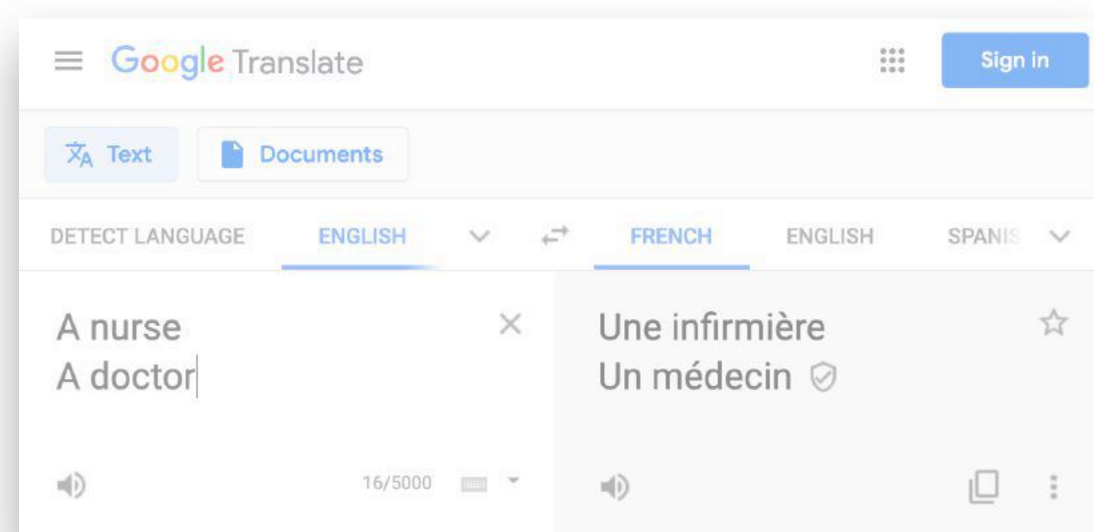


Safety

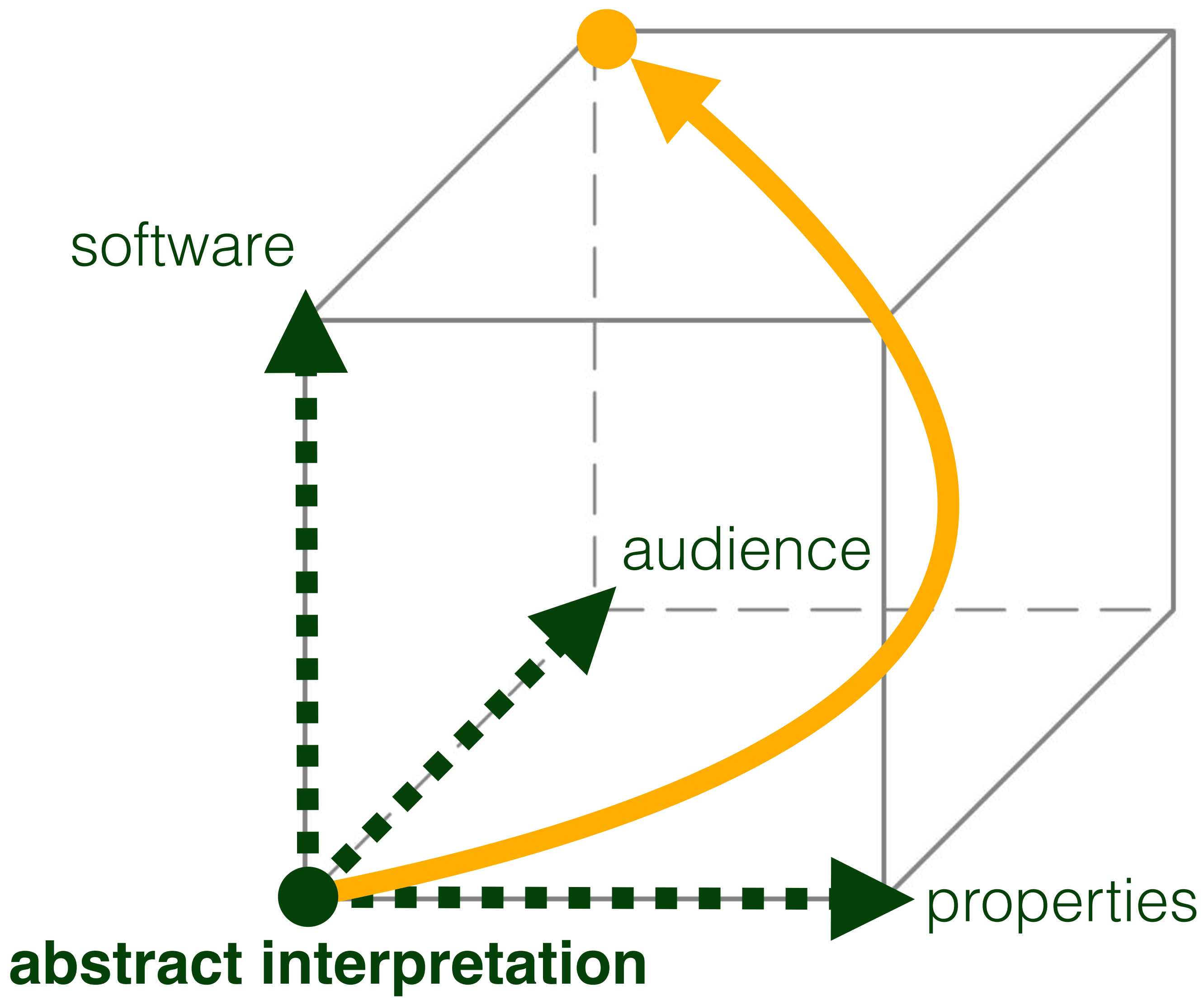
Goal G4 in [Kurd03]



Fairness

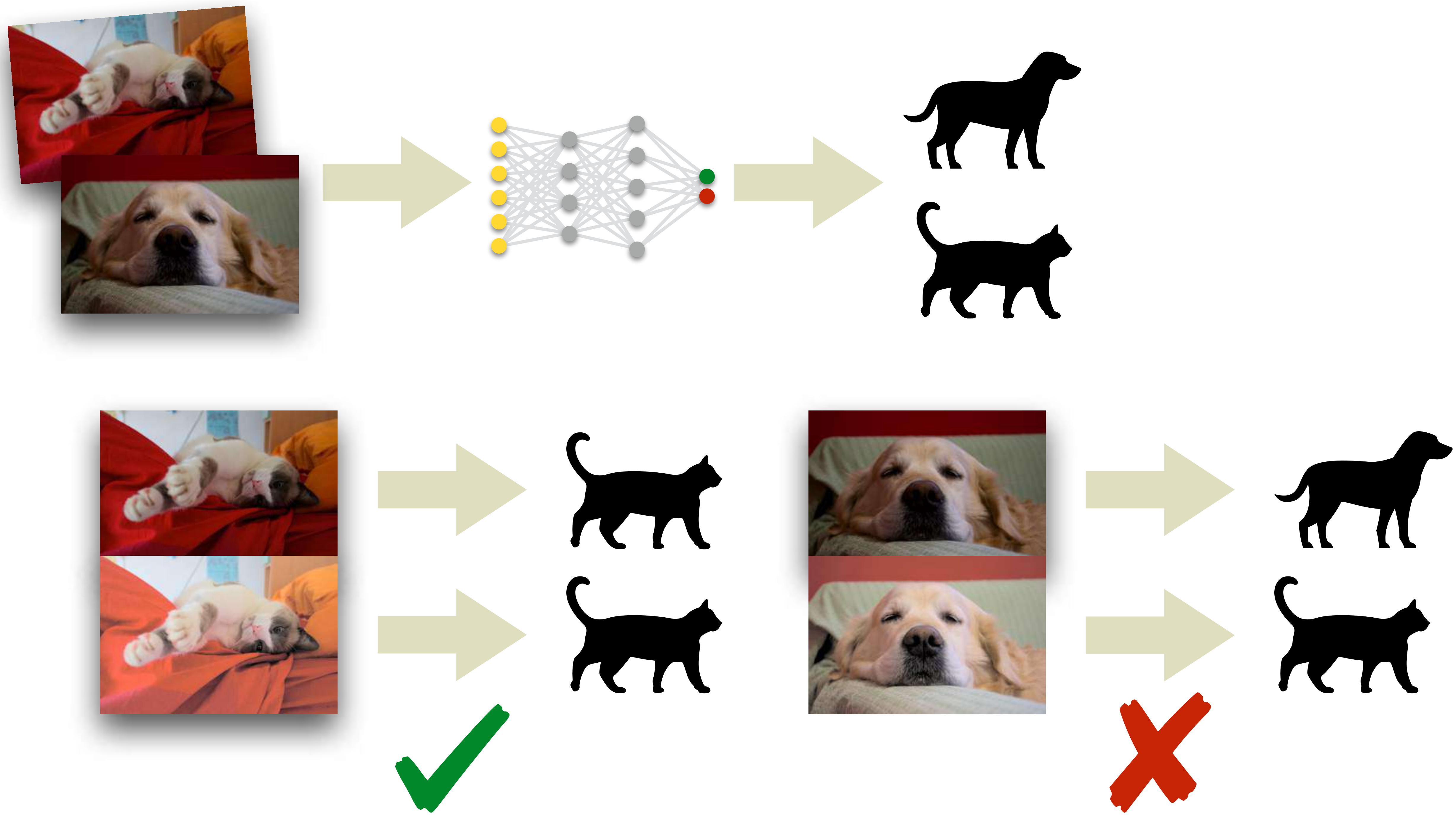


Stability Verification



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

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

$\mathscr{R}_{\mathbf{x}}^{\delta,\epsilon}$ 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 $\mathbf{x}$

$\text{STABLE}_{\mathbf{x}}^{\delta,\epsilon}(T) \stackrel{\text{def}}{=} \forall t, t' \in T\colon\; t_0 = \mathbf{x} \wedge t'_0 \in P_{\delta,\epsilon}(\mathbf{x}) \Rightarrow t_\omega = t'_\omega$

Theorem

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

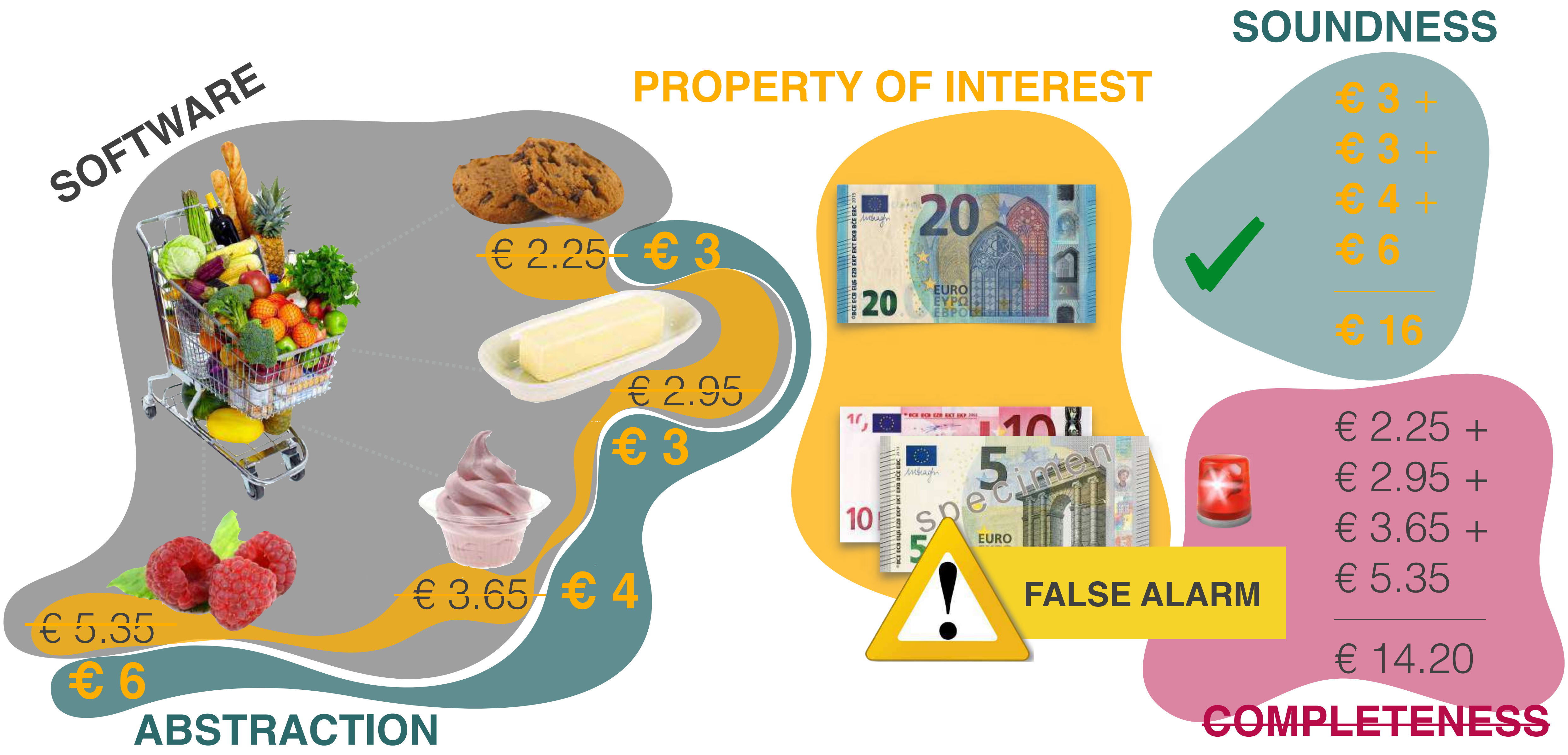
Corollary

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

Static Analysis Methods

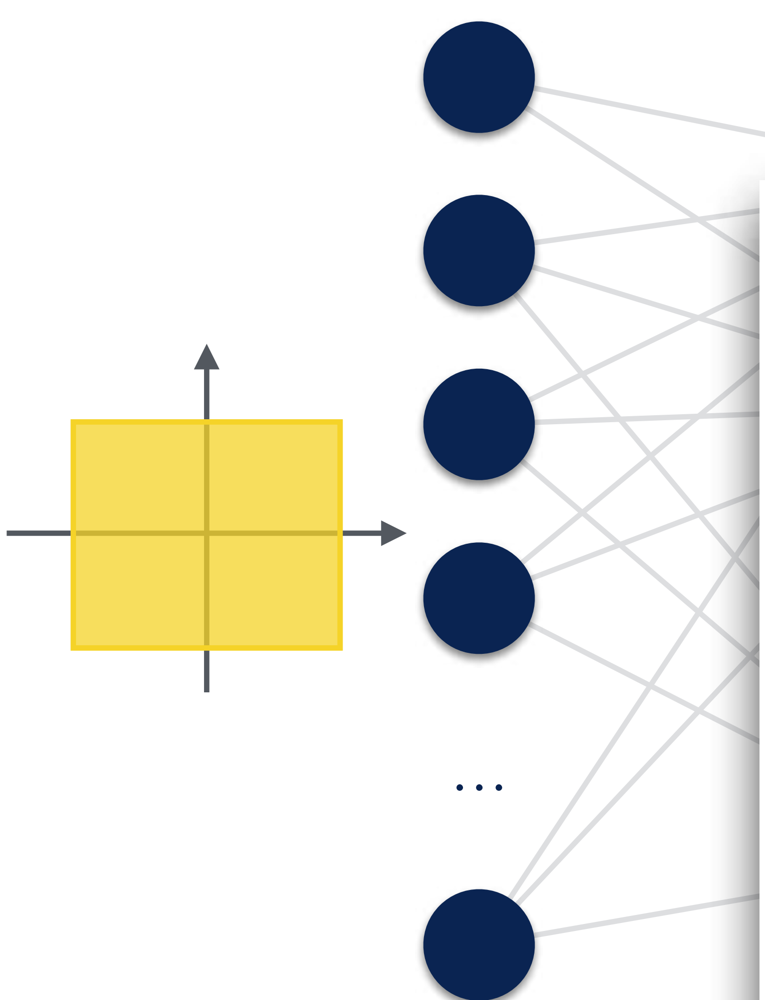
Abstract Interpretation

Intuition



Forward Analysis

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



Local Prediction Stability

Distance-Based Perturbations

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

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

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

$\mathscr{R}_{\mathbf{x}}^{\delta,\epsilon}$ 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 $\mathbf{x}$

$\text{STABLE}_{\mathbf{x}}^{\delta,\epsilon}(T) \overset{\text{def}}{=} \forall t, t' \in T\colon t'_0 = \mathbf{x} \wedge t_0 \in P_{\delta,\epsilon}(\mathbf{x}) \Rightarrow t_\omega = t'_\omega$

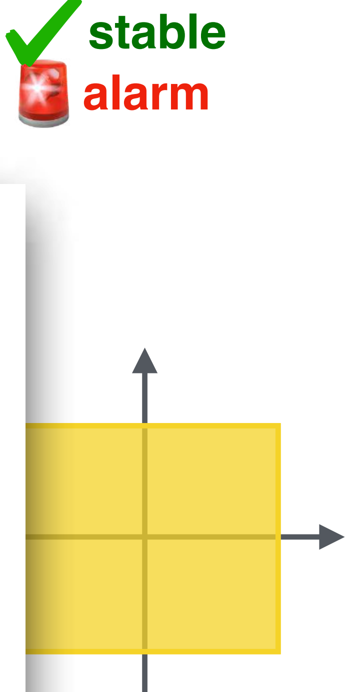
Theorem

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

Corollary

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

29

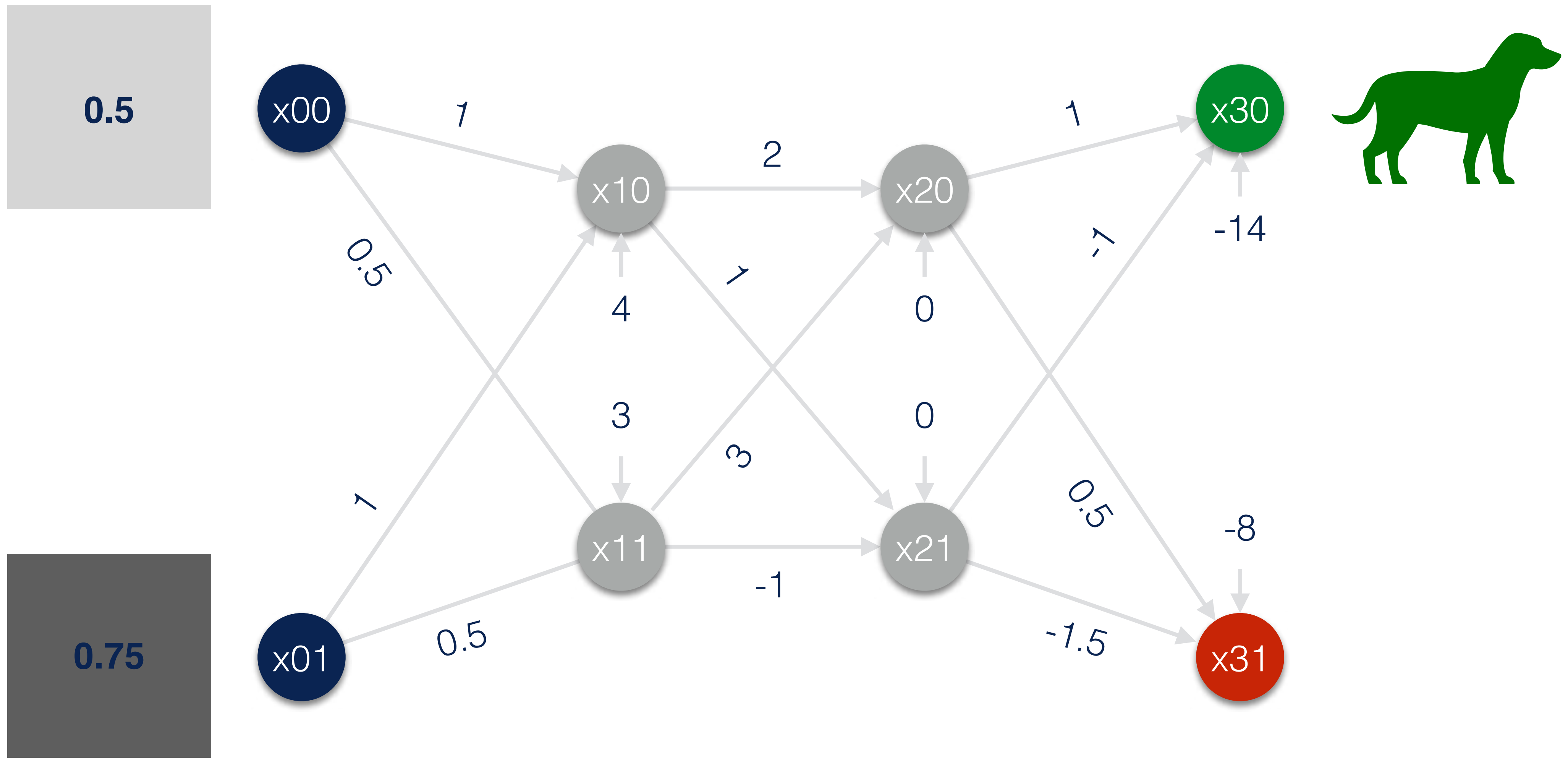


Theorem

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

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

Example

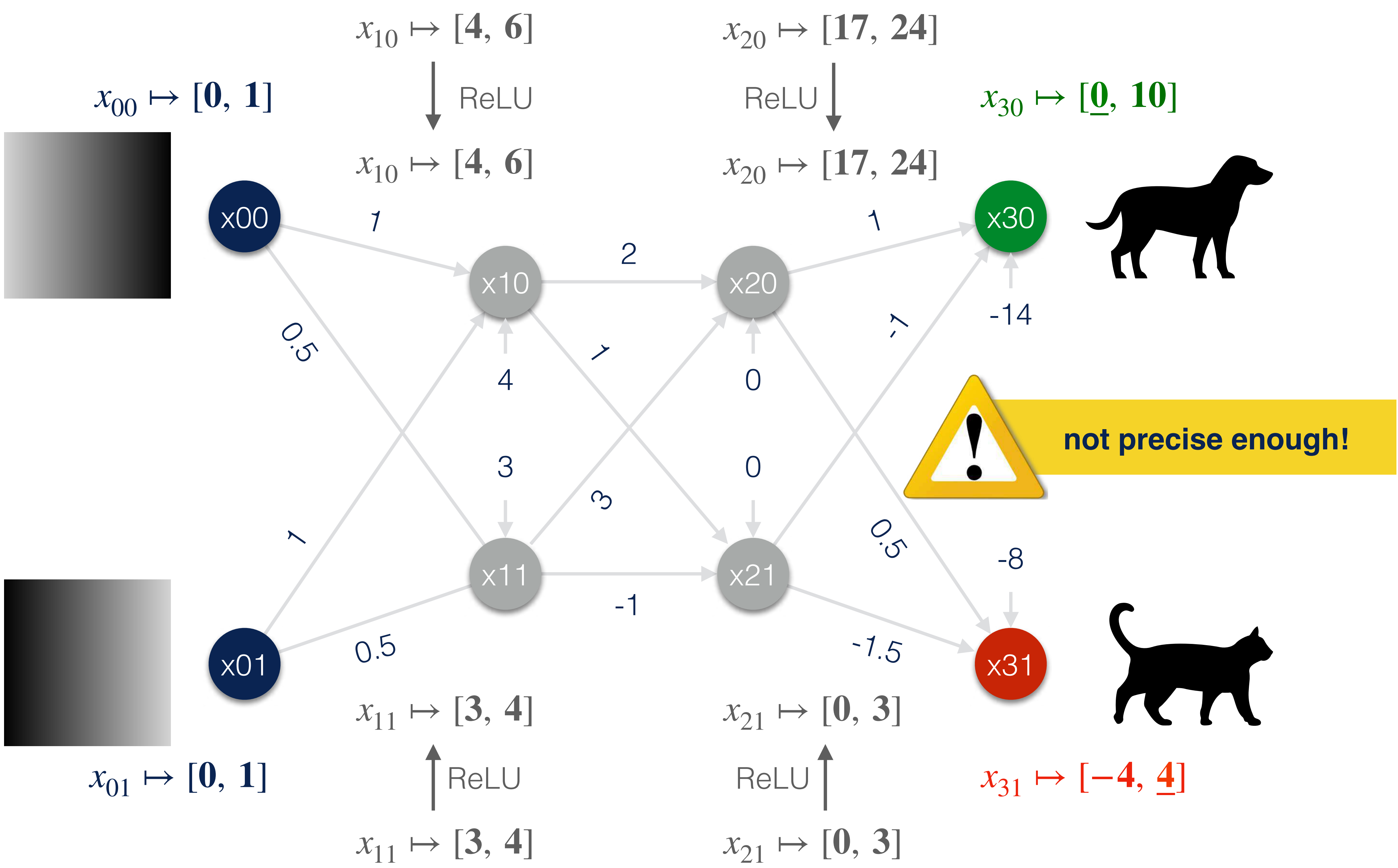


$$P(\langle 0.5,0.75\rangle) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathscr{R} \times \mathscr{R} \mid 0 \leq \mathbf{x}_0 \leq 1 \wedge 0 \leq \mathbf{x}_1 \leq 1\}$$

Interval Abstraction

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

$a, b \in \mathscr{R}$



Abstract Interpretation

Improving Precision

~~€ 2.25~~ **€ 2,5**

~~€ 2.95~~ **€ 3**

~~€ 3.65~~ **€ 4**

~~€ 5.35~~ **€ 5,5**

€ 2,5 +
€ 3 +
€ 4 +
€ 5,5

€ 15

€ 2.25 +
€ 2.95 +
€ 3.65 +
€ 5.35

€ 14.20

Interval Abstraction

with Symbolic Constant Propagation [Li19]

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

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

$x_{i-1,0} \mapsto \mathbf{E_{i-1,0}}$
$\dots$
$x_{i-1,j} \mapsto \mathbf{E_{i-1,j}}$
$\dots$

$$\xrightarrow{x_{i,j} = \sum_k w_{j,k}^{i-1} \cdot x_{i-1,k} + b_{i,j}}$$

$$x_{i,j} \mapsto \sum_k w_{j,k}^{i-1} \cdot \mathbf{E_{i-1,k}} + b_{i,j}$$

$$x_{i,j} \mapsto \begin{cases} \mathbf{E_{i,j}} \\ [\mathbf{a}, \mathbf{b}] \end{cases} \xrightarrow{\text{ReLU}}$$

$$x_{i,j} \mapsto \begin{cases} \mathbf{E_{i,j}} \\ [\mathbf{a}, \mathbf{b}] \end{cases} \qquad 0 \leq a$$

$$x_{i,j} \mapsto \begin{cases} \mathbf{x_{i,j}} \\ [\mathbf{0}, \mathbf{b}] \end{cases} \qquad a < 0 \wedge 0 < b$$

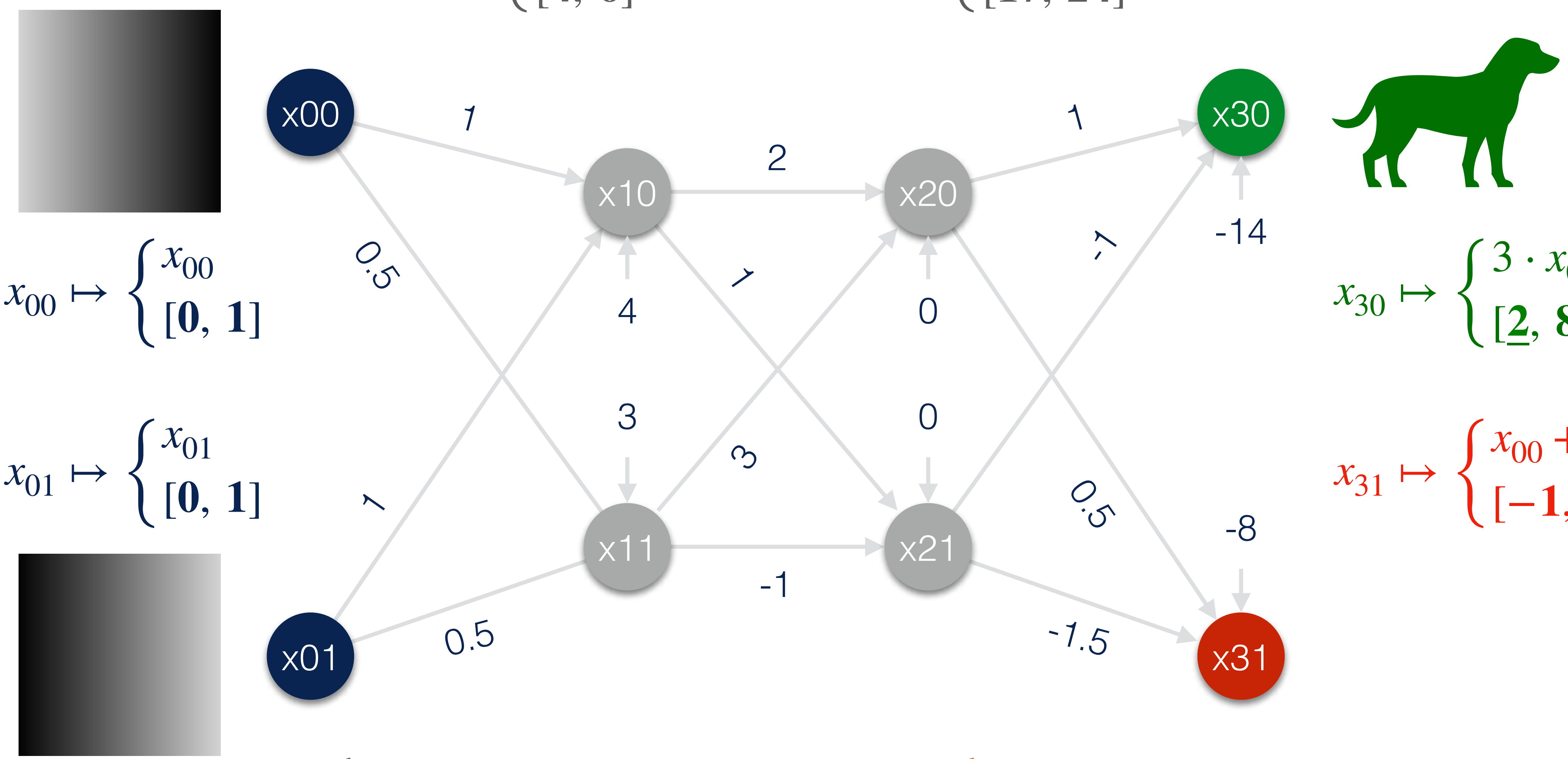
$$x_{i,j} \mapsto \begin{cases} \mathbf{0} \\ [\mathbf{0}, \mathbf{0}] \end{cases} \qquad b \leq 0$$

Interval Abstraction

with Symbolic Constant Propagation [Li19]

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

$x_{20} \mapsto \begin{cases} 2 \cdot (x_{00} + x_{01} + 4) + 3 \cdot (0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3) \\ [\mathbf{17},\ \mathbf{24}] \end{cases}$



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

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

$x_{30} \mapsto \begin{cases} 3 \cdot x_{00} + 3 \cdot x_{01} + 2 \\ [\underline{\mathbf{2}},\ \mathbf{8}] \end{cases}$

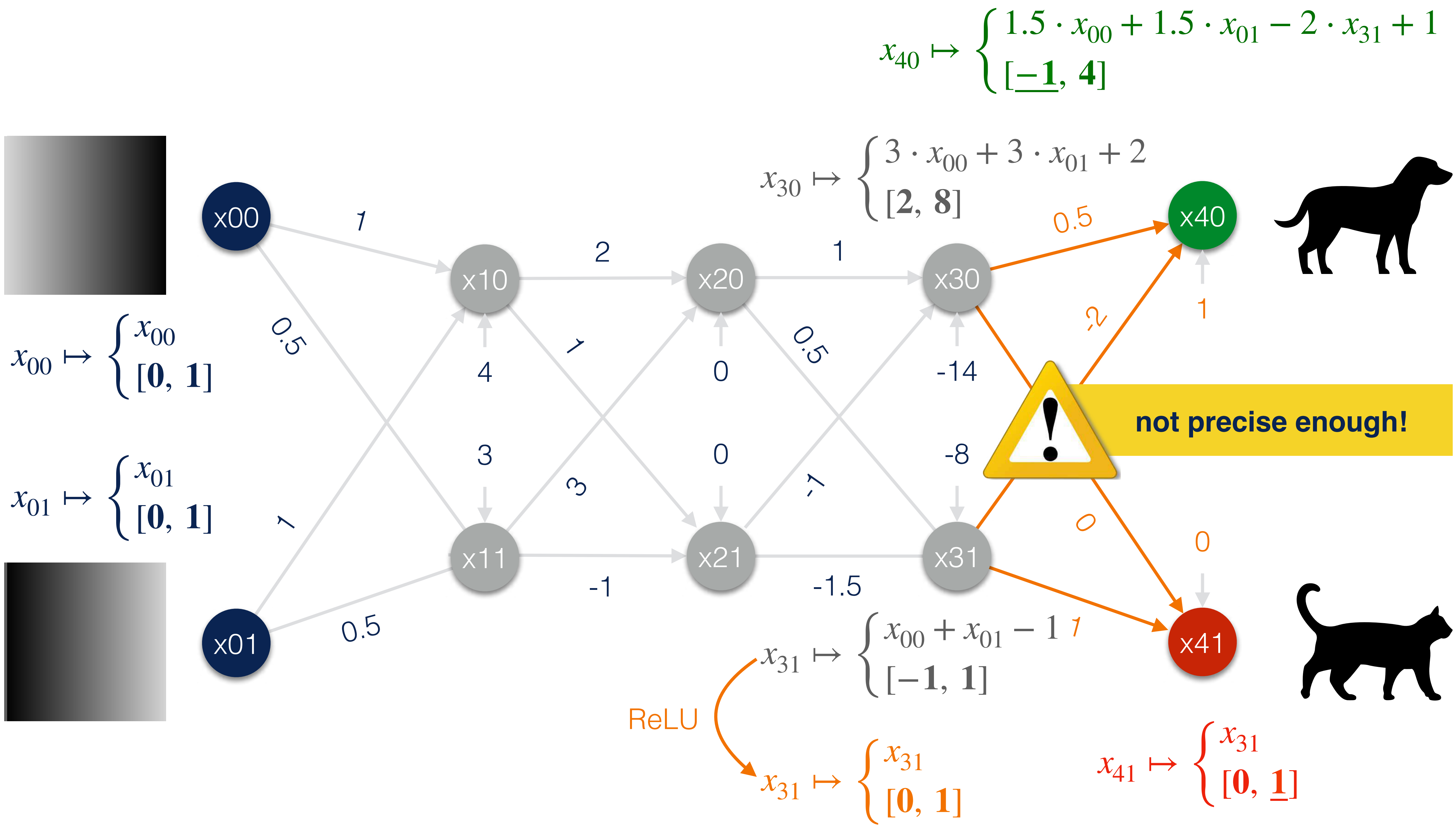
$x_{31} \mapsto \begin{cases} x_{00} + x_{01} - 1 \\ [\mathbf{-1},\ \underline{\mathbf{1}}] \end{cases}$

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

$x_{21} \mapsto \begin{cases} (x_{00} + x_{01} + 4) - 1 \cdot (0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 3) \\ [\mathbf{1},\ \mathbf{2}] \end{cases}$

Interval Abstraction

with Symbolic Constant Propagation [Li19]



DeepPoly [Singh19]

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

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

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

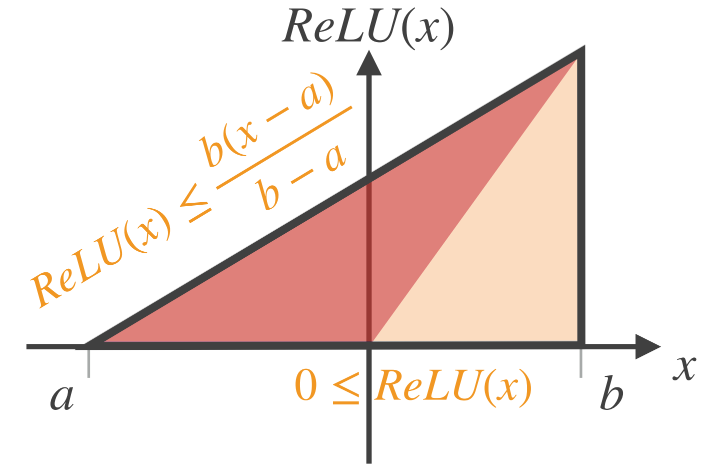
ReLU $0 \leq a$:

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

ReLU $a < 0 \wedge 0 < b$

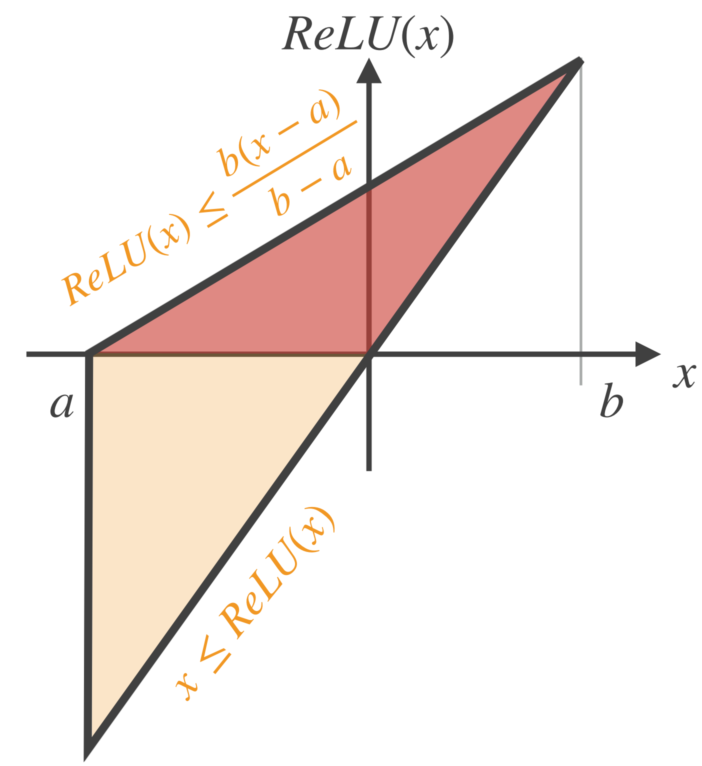
$b \leq -a$:

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



$-a < b$:

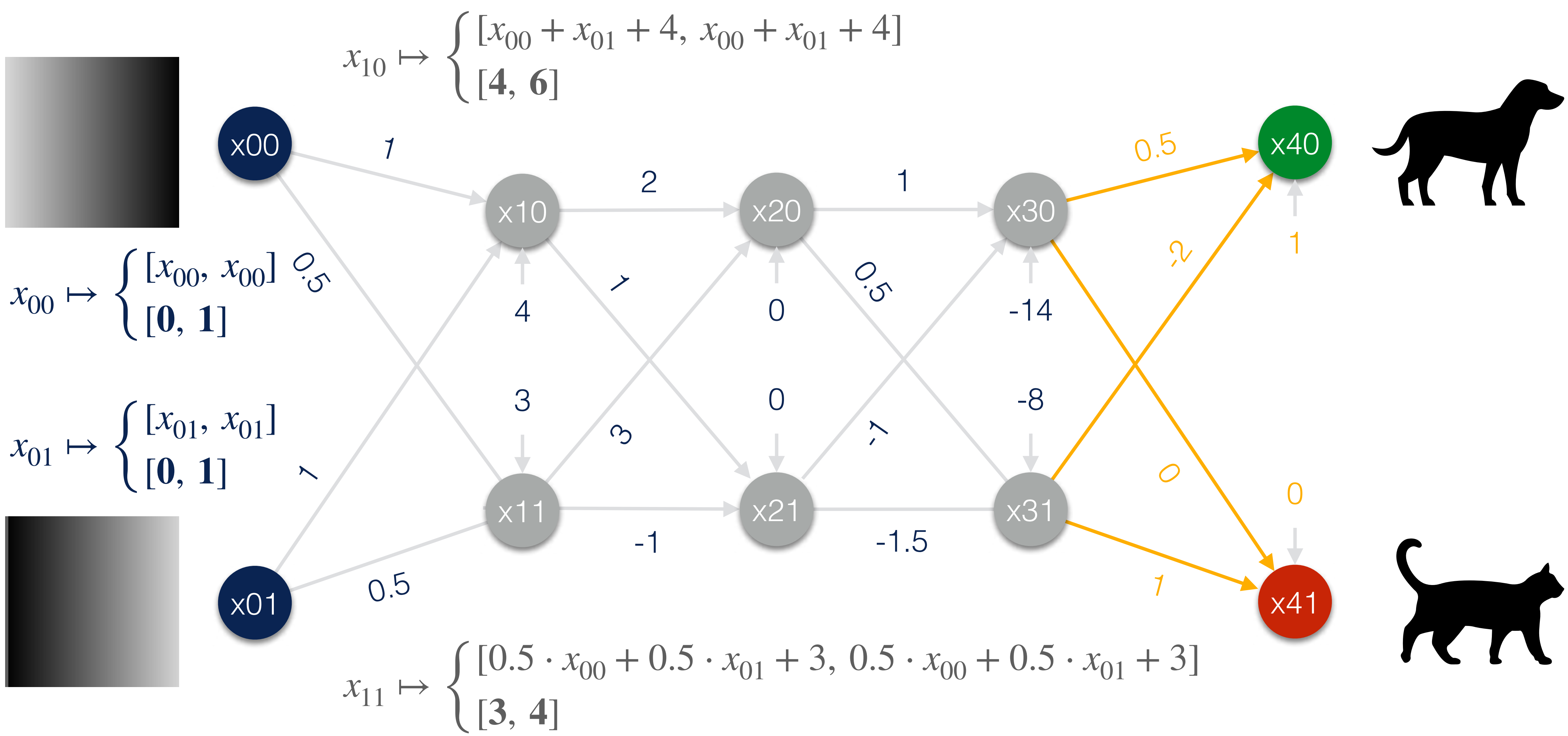
$$x_{i,j} \mapsto \begin{cases} \left[\mathbf{x_{i,j}}, \dfrac{\mathbf{b(x_{i,j} - a)}}{\mathbf{b - a}}\right] \\ [\mathbf{a}, \mathbf{b}] \end{cases}$$



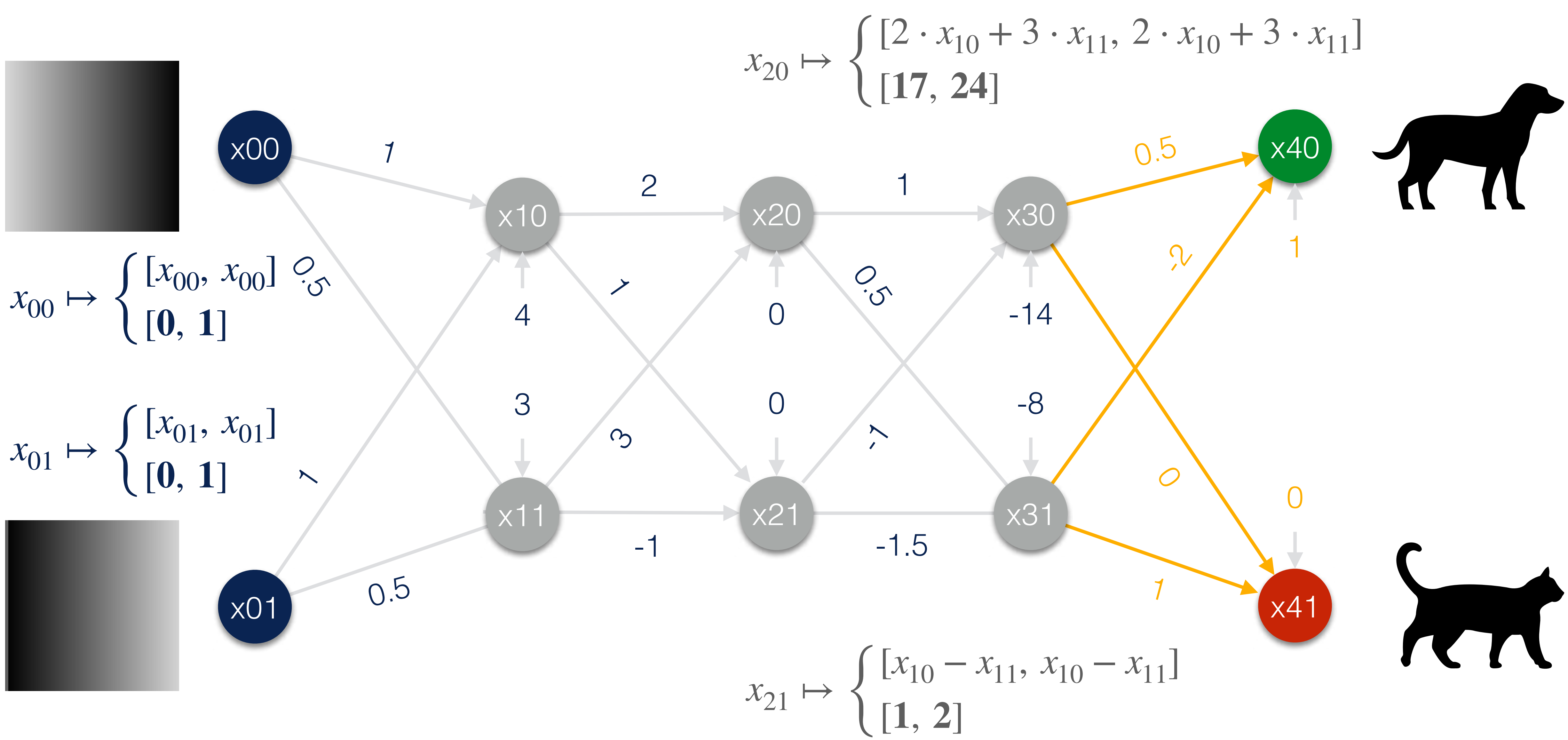
ReLU $b \leq 0$:

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

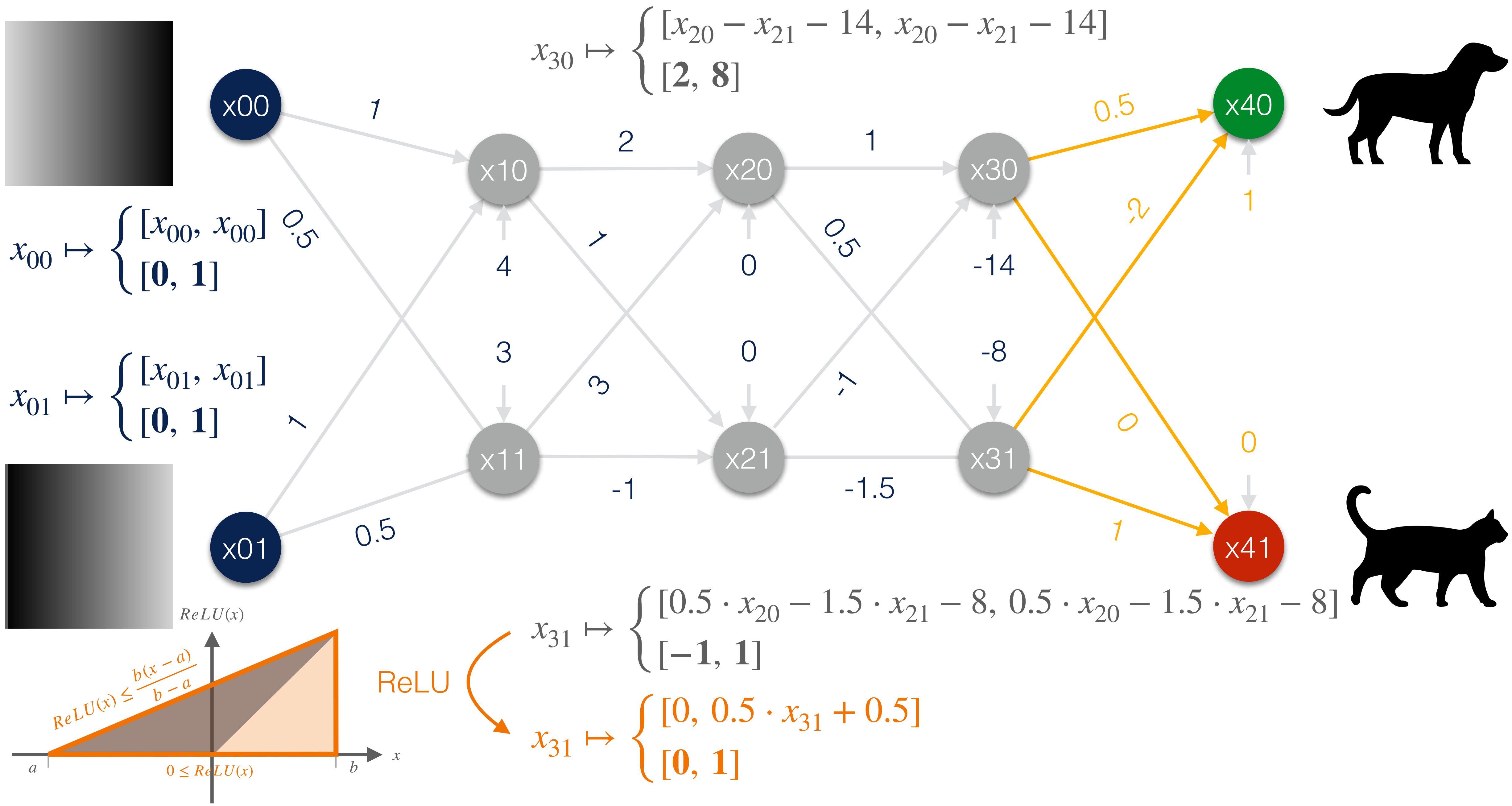
DeepPoly [Singh19]



DeepPoly [Singh19]

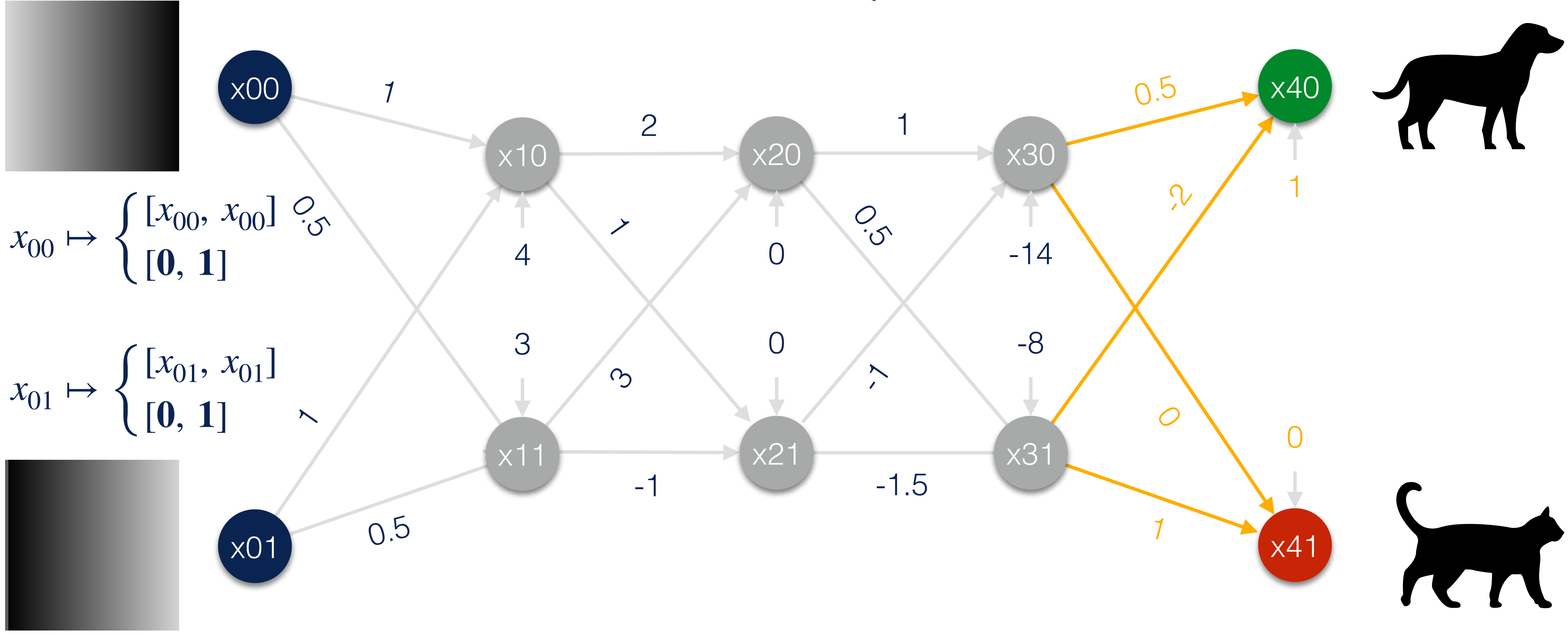


DeepPoly [Singh19]



DeepPoly [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 [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 [Singh19]

Partial Back-Substitution

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

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

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

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

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

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

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

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

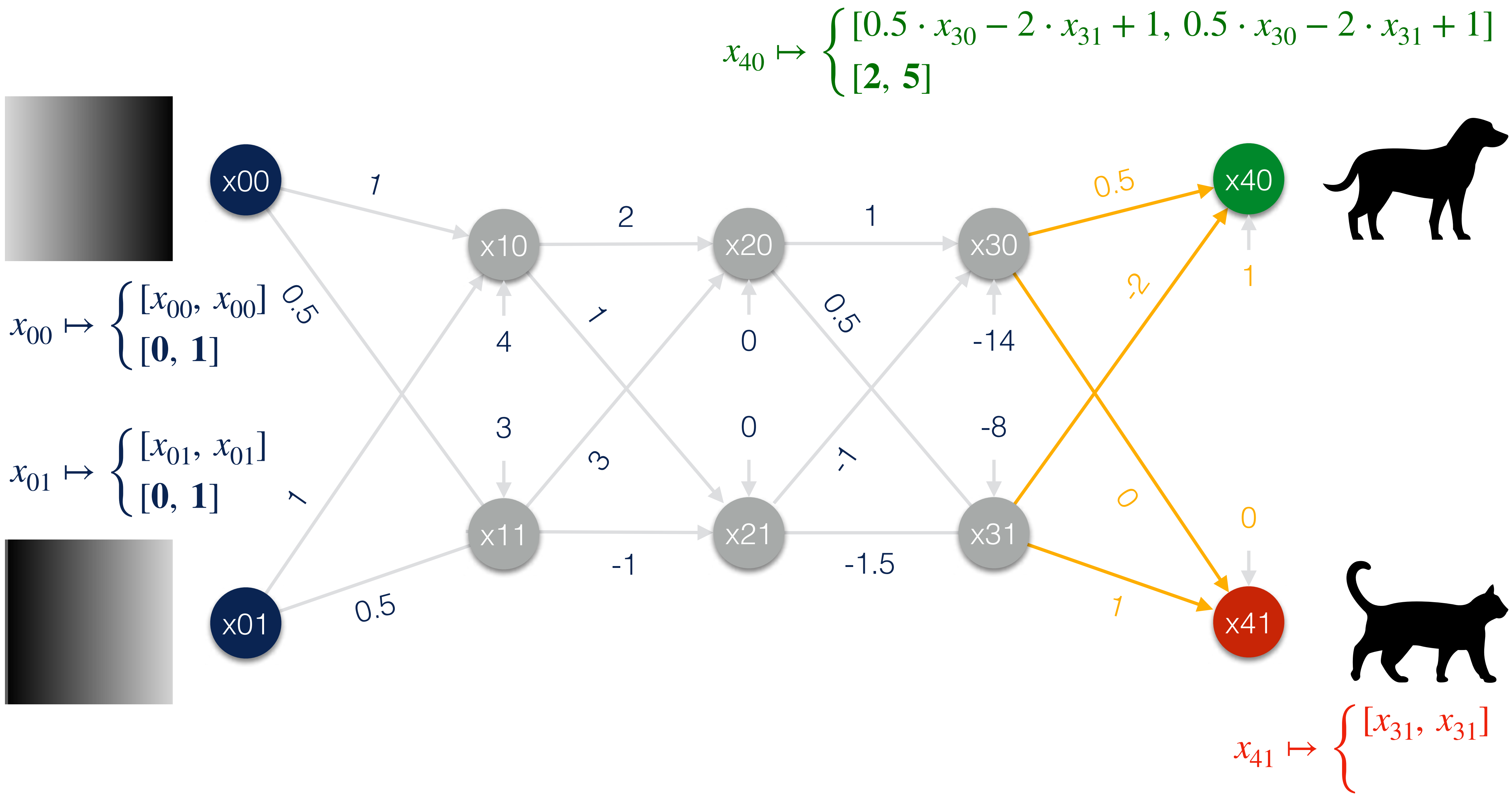
$$x_{40} \mapsto \begin{cases} [0.5 \cdot x_{30} - 2 \cdot x_{31} + 1, 0.5 \cdot x_{30} - 2 \cdot x_{31} + 1] \\ [\underline{\mathbf{0}}, \mathbf{5}] \end{cases}$$

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

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

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

DeepPoly [Singh19]



DeepPoly [Singh19]

Back-Substitution

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

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

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

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

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

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

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

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

$$x_{41} \mapsto \begin{cases} [x_{31},\ x_{31}] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ 0.25 \cdot x_{20} - 0.75 \cdot x_{21} - 3.5] \\ \end{cases}$$

$$\mapsto \begin{cases} [0,\ \ - 0.25 \cdot x_{10} + 1.5 \cdot x_{11} - 3.5] \\ \end{cases}$$

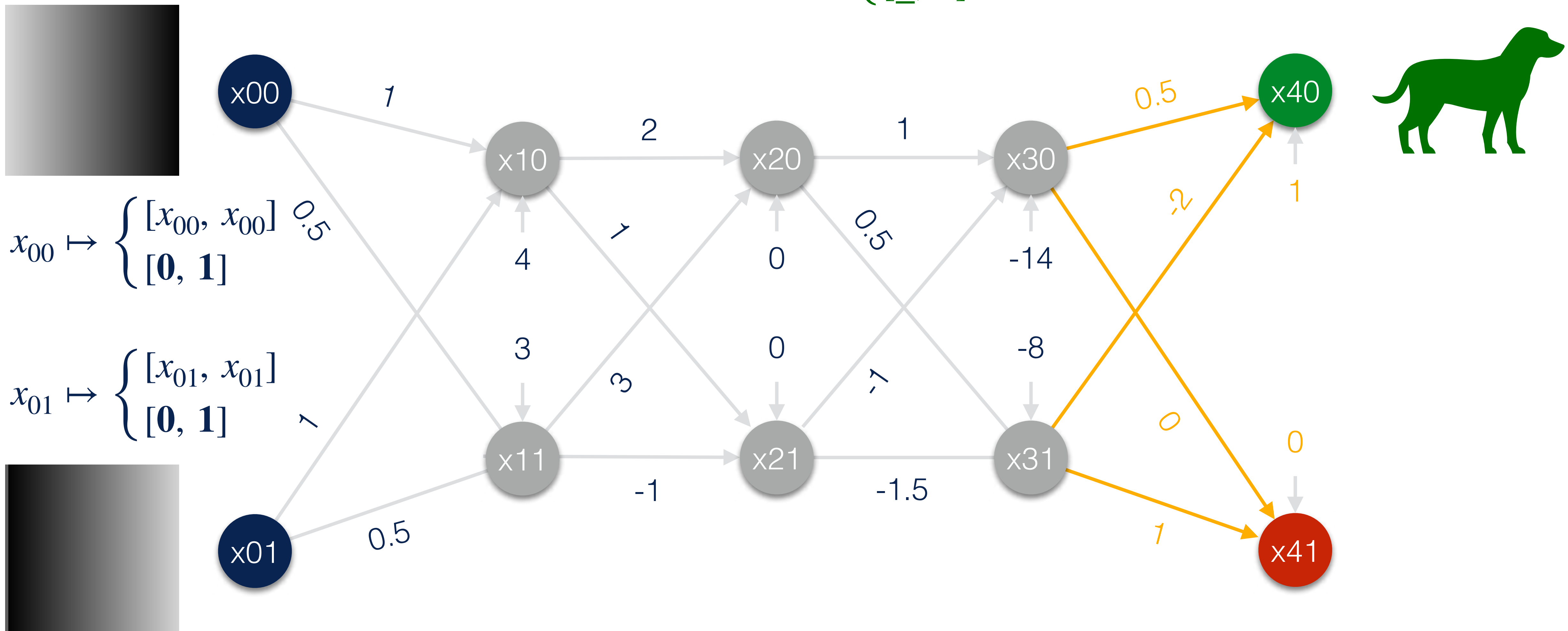
$$\mapsto \begin{cases} [0,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01}] \\ [\mathbf{0},\ \underline{\mathbf{1}}] \end{cases}$$

DeepPoly [Singh19]

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

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

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



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

DeepPoly [Singh19]

Maintaining Symbolic Bounds wrt Inputs

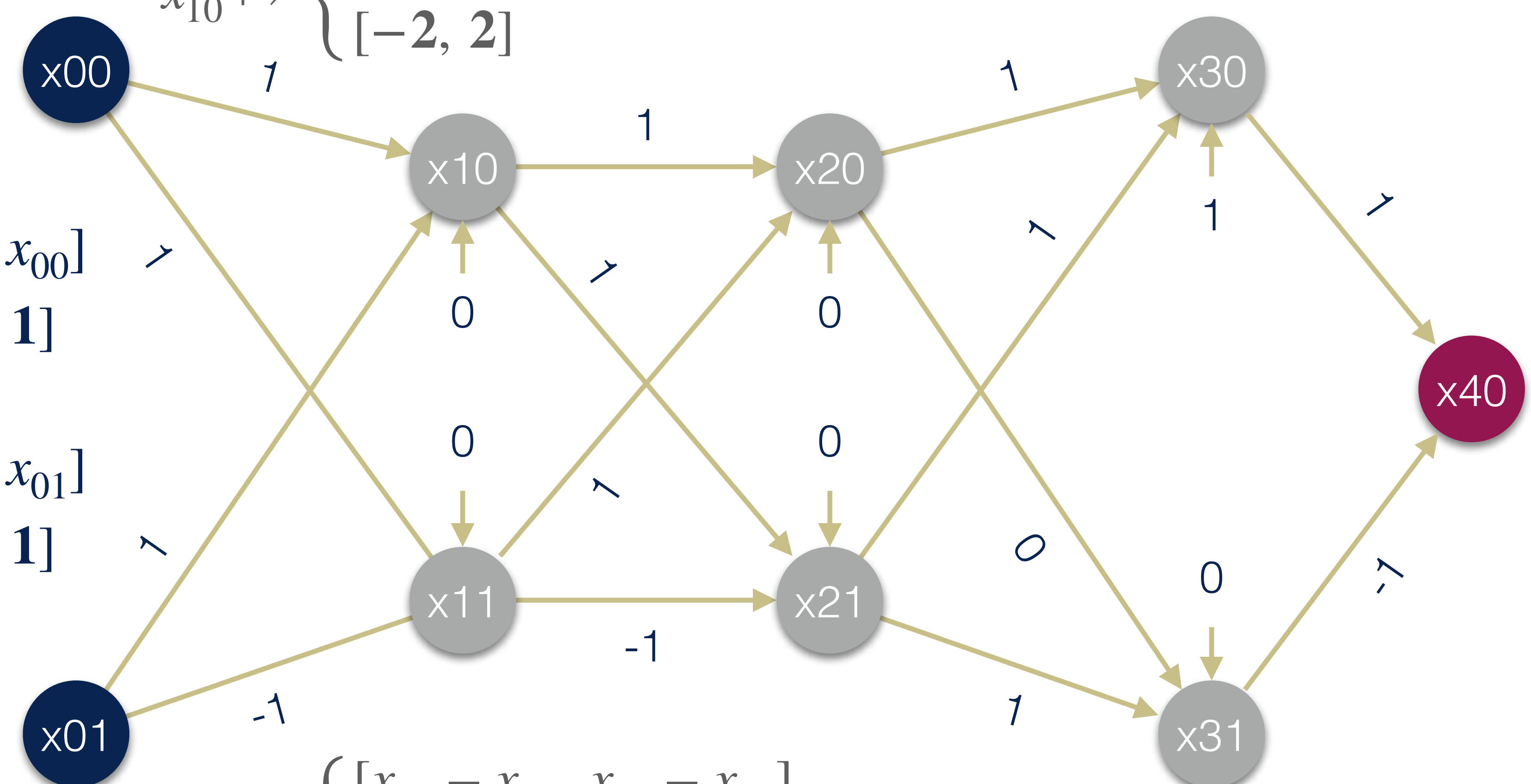
$x_{10} \mapsto [0,\ 0.5 \cdot x_{10} + 1] \rightarrow [0,\ 0.5 \cdot (x_{00} + x_{01}) + 1] = [0,\ 0.5 \cdot x_{00} + 0.5 \cdot x_{01} + 1]$

ReLU

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

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

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



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

ReLU

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

DeepPoly [Singh19]

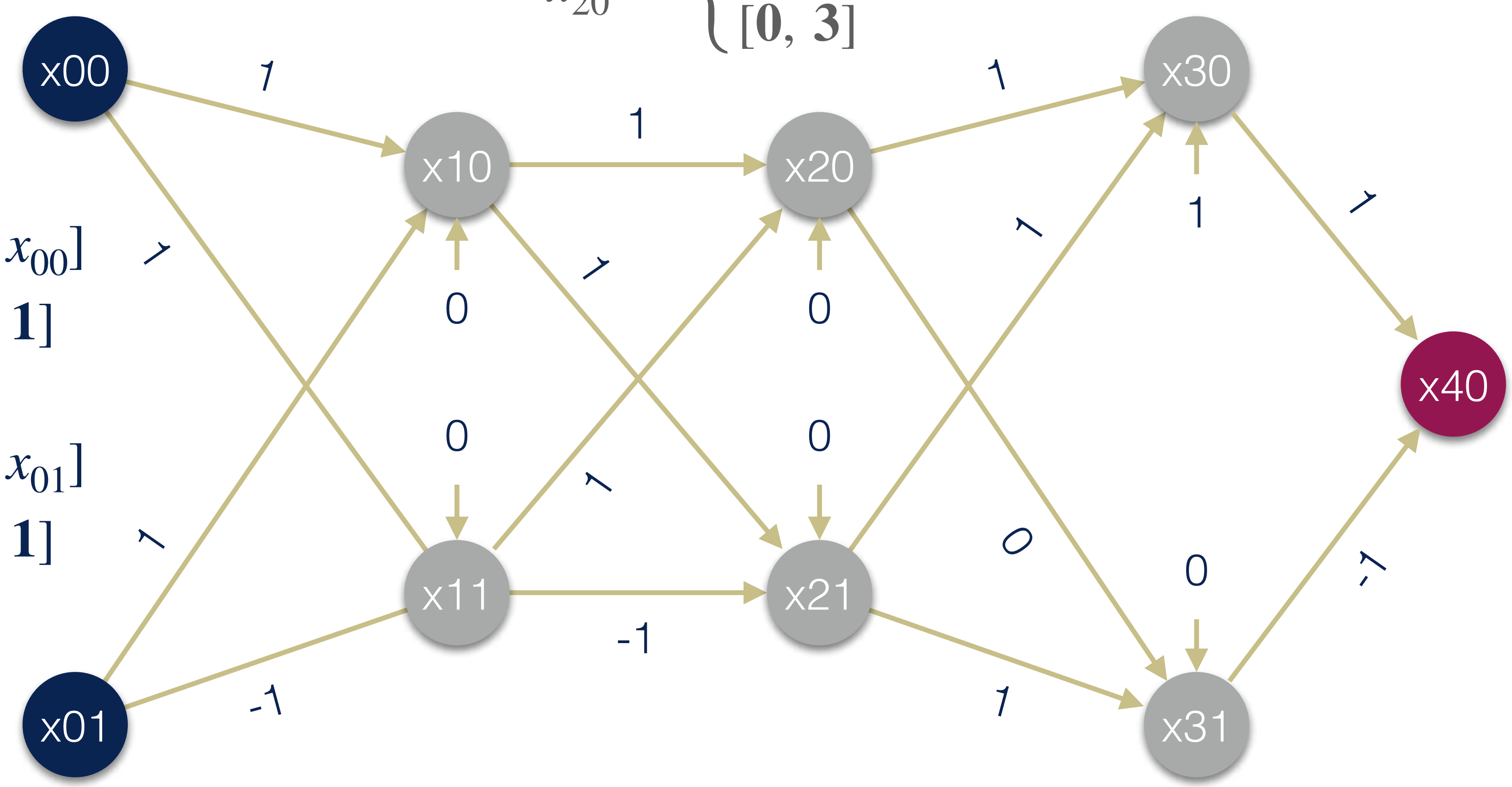
Maintaining Symbolic Bounds wrt Inputs

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

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

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

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



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

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

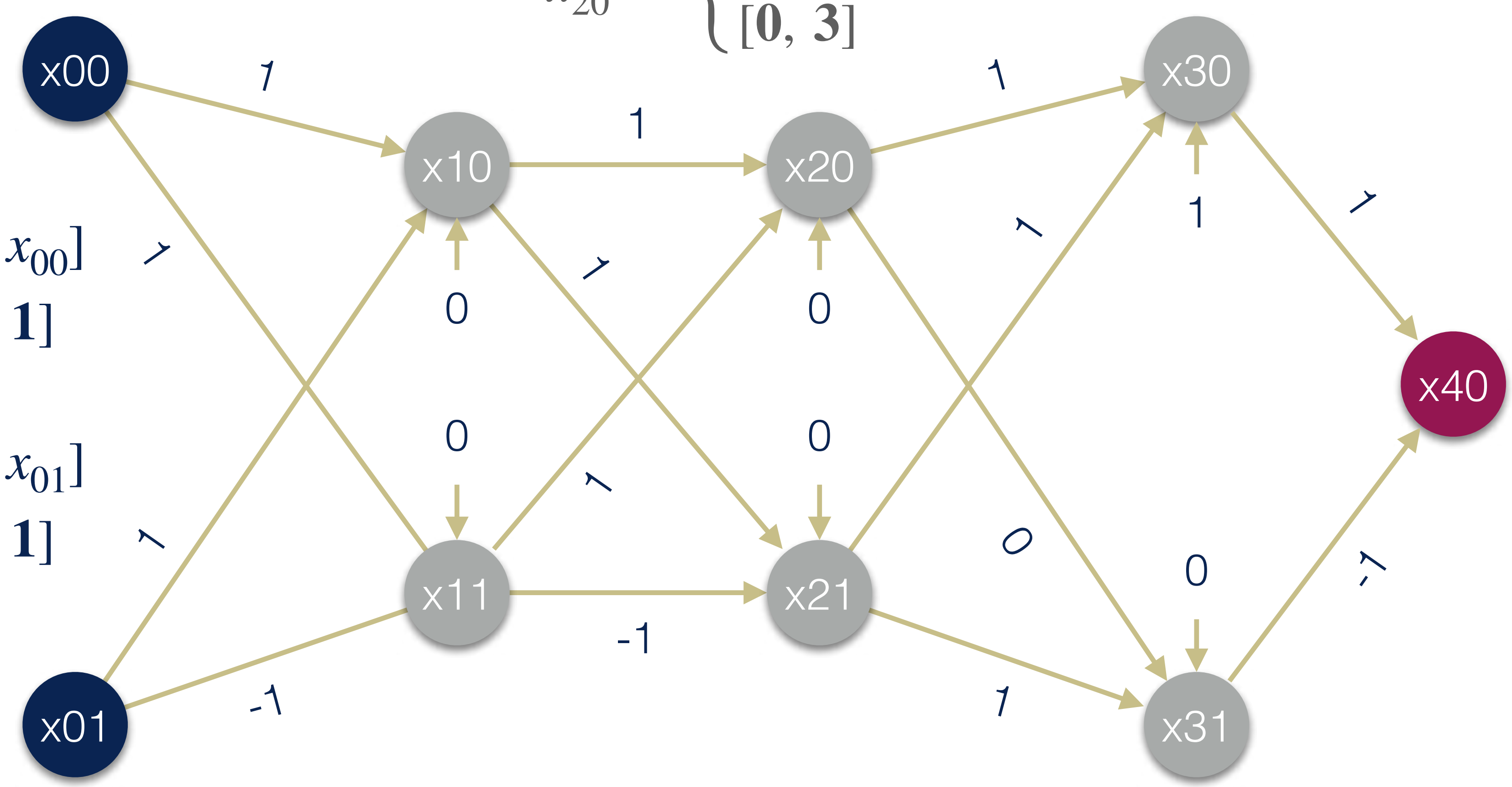
DeepPoly [Singh19]

Maintaining Symbolic Bounds wrt Inputs

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

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

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



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

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 [Singh19]

Maintaining Symbolic Bounds wrt Inputs

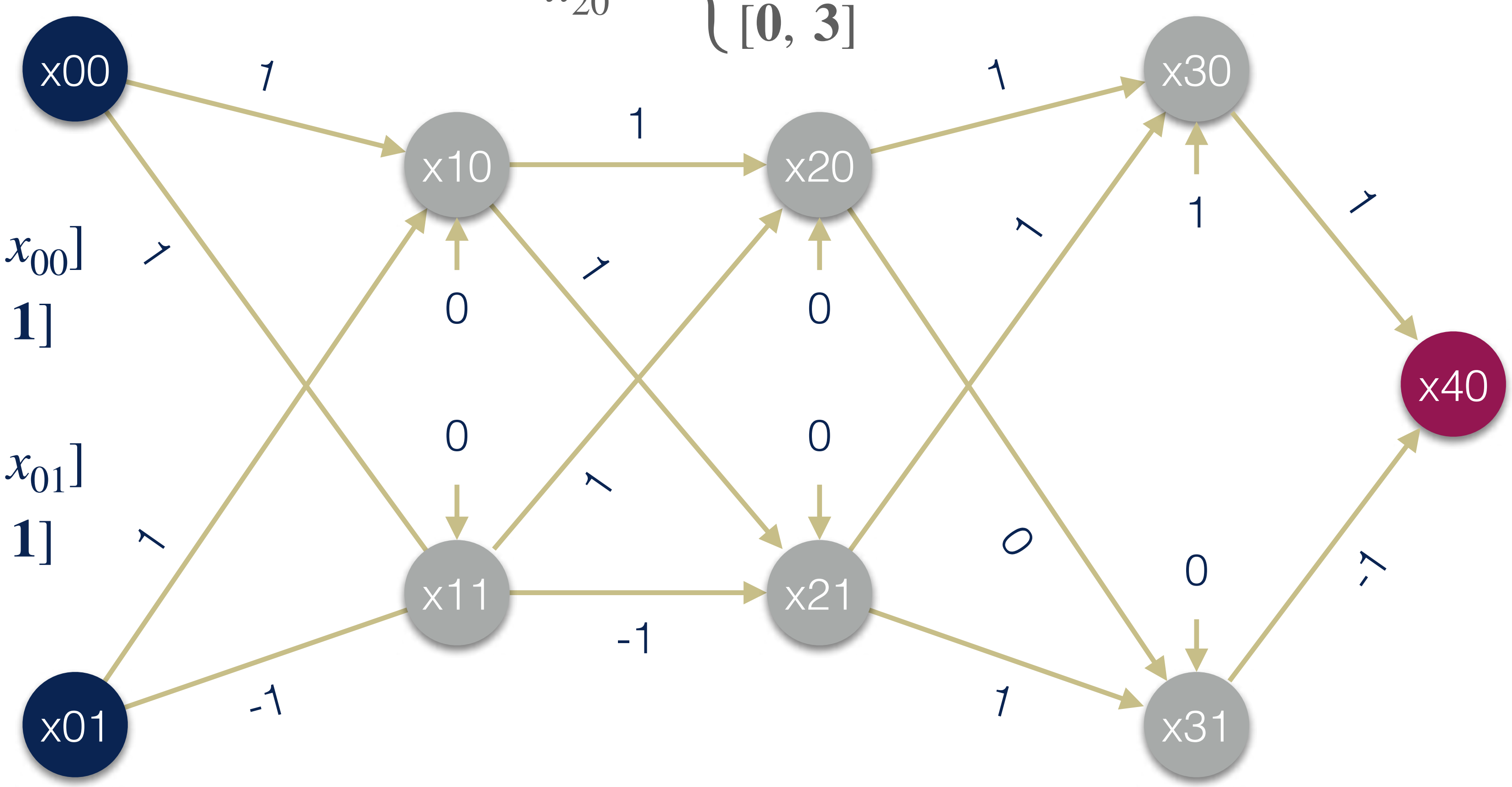
$$x_{30} \mapsto \begin{cases} [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] \rightarrow [1,\ 1.25 \cdot x_{00} - 0.25 \cdot x_{01} + 4.5] \\ [\mathbf{1},\ \underline{\mathbf{6}}] \end{cases}$$

$\leftarrow [\mathbf{1}, \mathbf{5.5}]$ with back-substitution

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

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

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



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

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

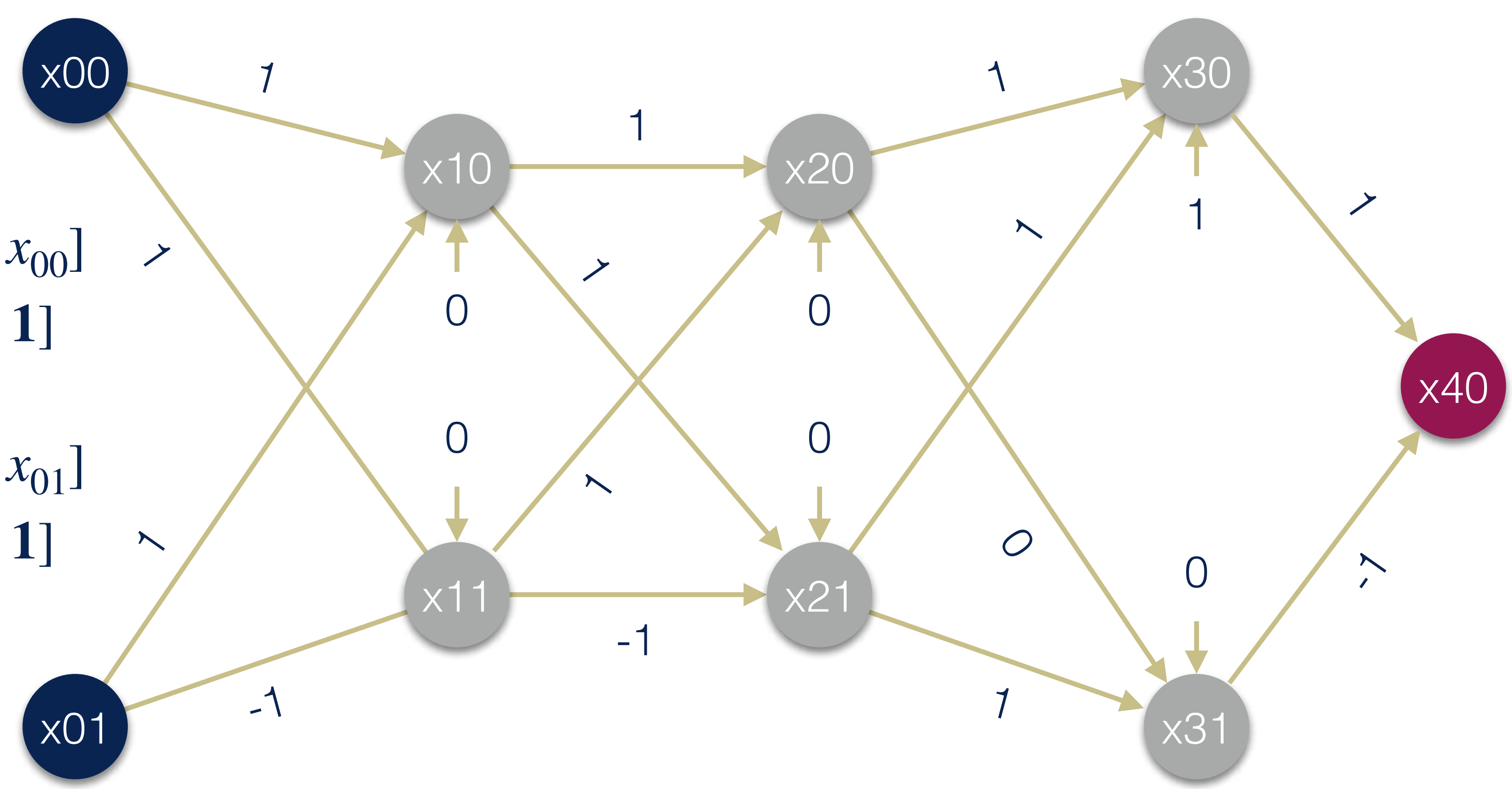
DeepPoly [Singh19]

Maintaining Symbolic Bounds wrt Inputs

$$x_{30} \mapsto \begin{cases} [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] \rightarrow [1,\ 1.25 \cdot x_{00} - 0.25 \cdot x_{01} + 4.5] \\ [\mathbf{1},\ \underline{\mathbf{6}}] \leftarrow [\mathbf{1}, \mathbf{5.5}] \text{ with back-substitution} \end{cases}$$

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

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



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

$$x_{40} \mapsto \begin{cases} \dots \\ [\mathbf{-1},\ \underline{\mathbf{6}}] \leftarrow [\mathbf{1}, \mathbf{4}] \text{ with back-substitution} \end{cases}$$

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 networks

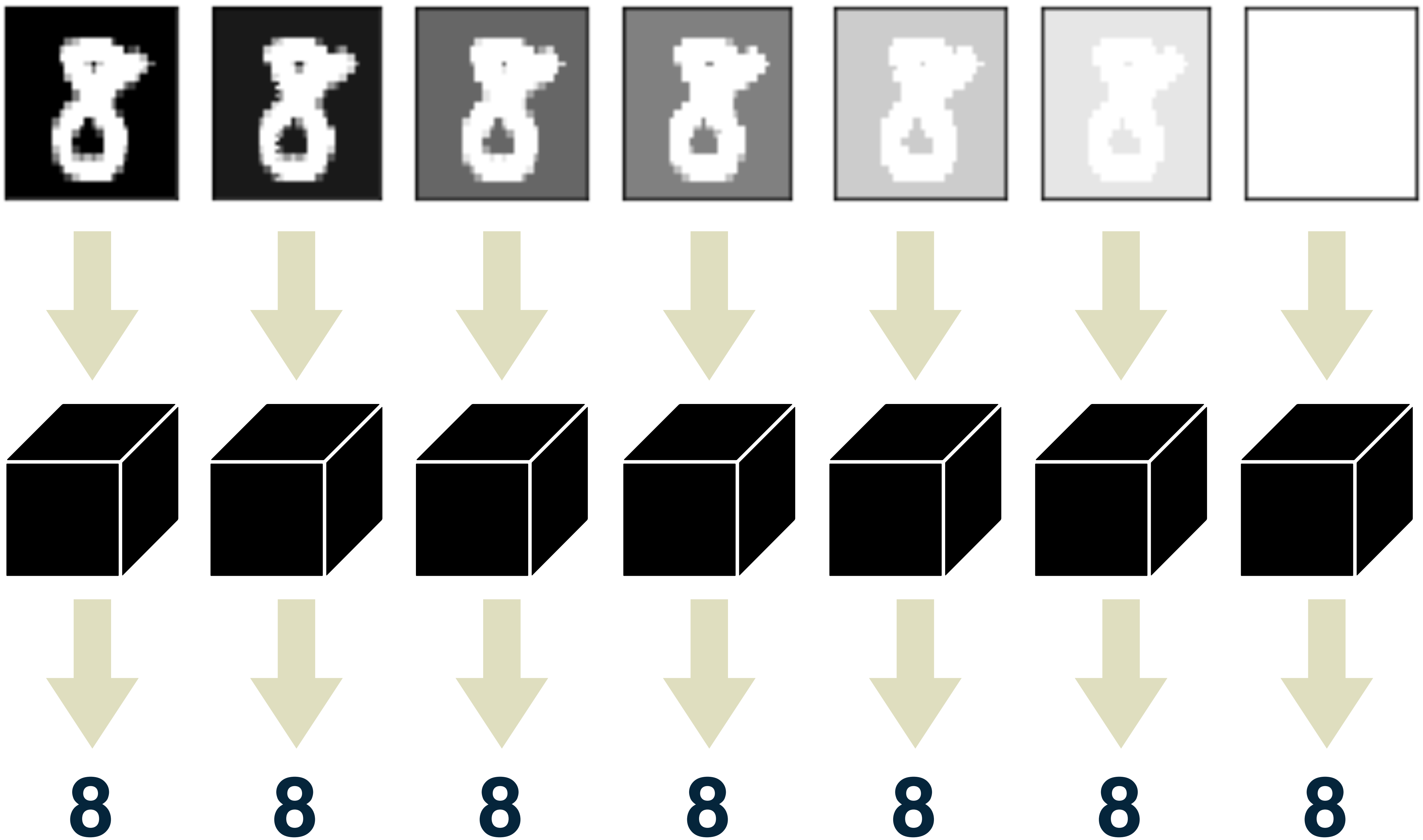
- **G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev**. *Fast and Effective Robustness Certification*. In NeurIPS, 2018.
  a custom zonotope domain for certifying neural networks

- **G. Singh, R. Ganvir, M. Püschel, and M. Vechev**. *Beyond the Single Neuron Convex Barrier for Neural Network Certification*. In NeurIPS, 2019.
  a framework to jointly approximate k ReLU activations

- **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

Local Prediction Stability

Not Enough!



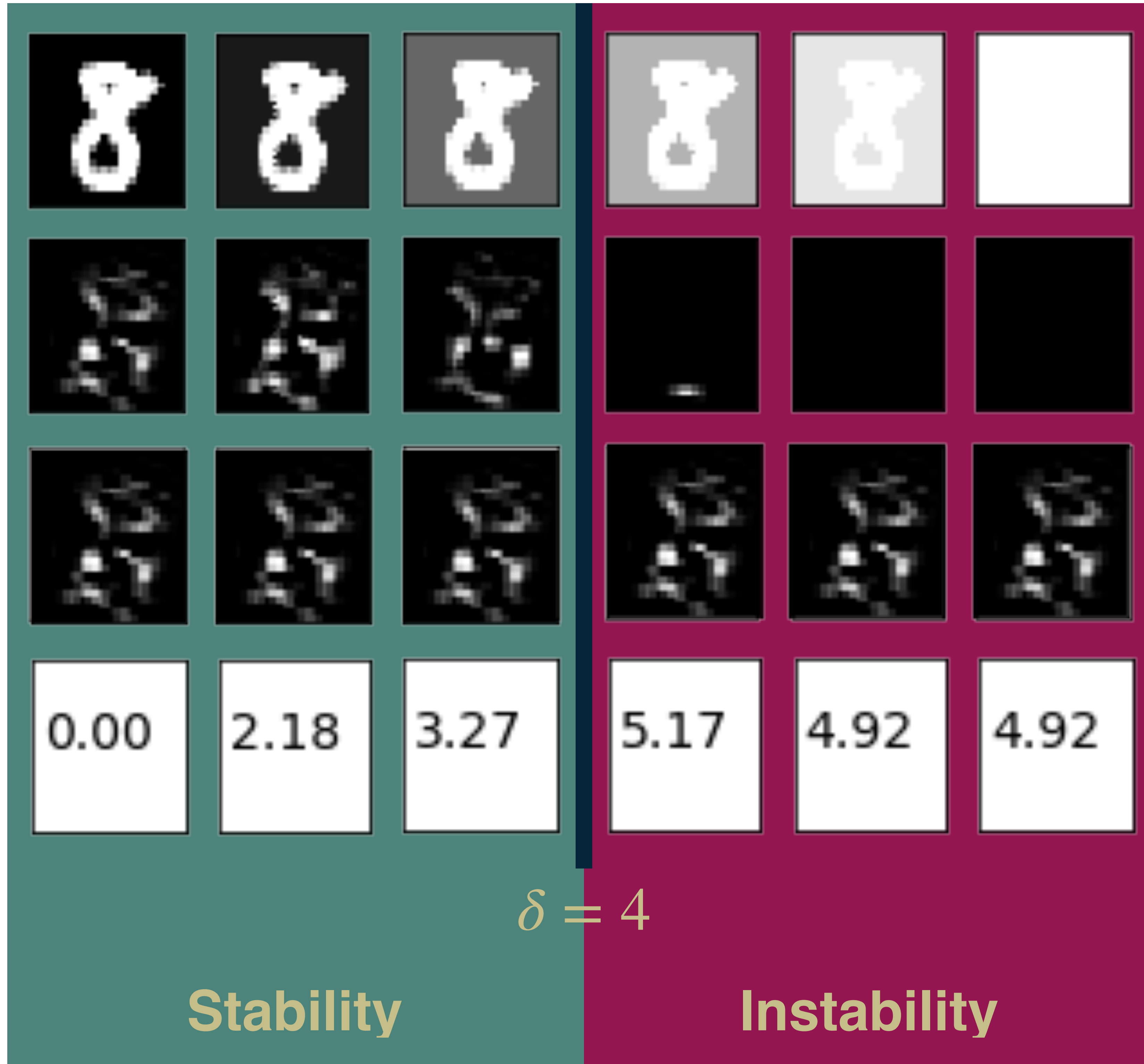
Local Explanation Stability [Munakata23]

Input

Saliency Map

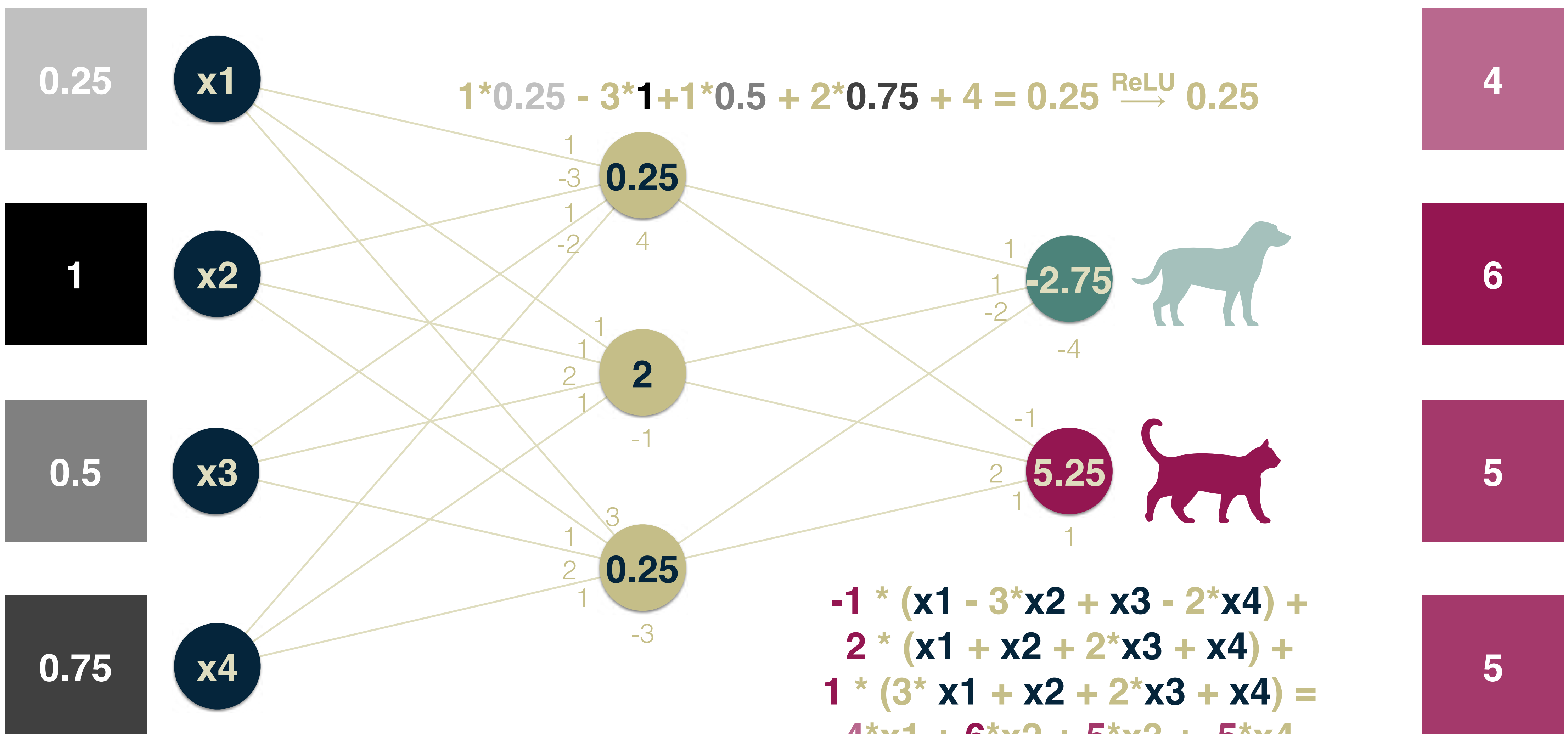
Expected Saliency

Distance



Example

Saliency Maps

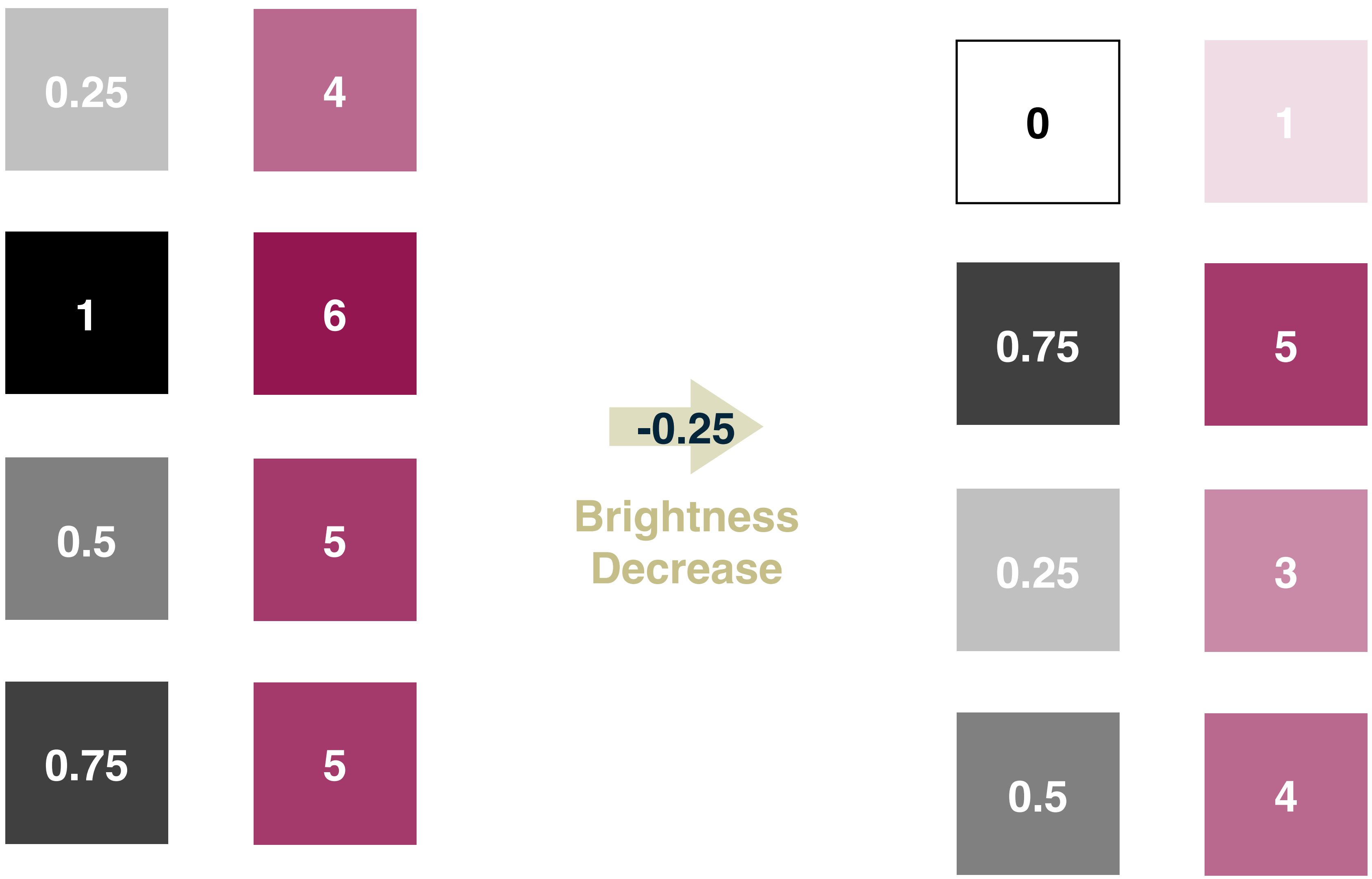


$$1*0.25 - 3*1 + 1*0.5 + 2*0.75 + 4 = 0.25 \xrightarrow{\text{ReLU}} 0.25$$

$$\begin{aligned}
&-1 * (x1 - 3*x2 + x3 - 2*x4) + \\
&2 * (x1 + x2 + 2*x3 + x4) + \\
&1 * (3* x1 + x2 + 2*x3 + x4) = \\
&4*x1 + 6*x2 + 5*x3 + 5*x4
\end{aligned}$$

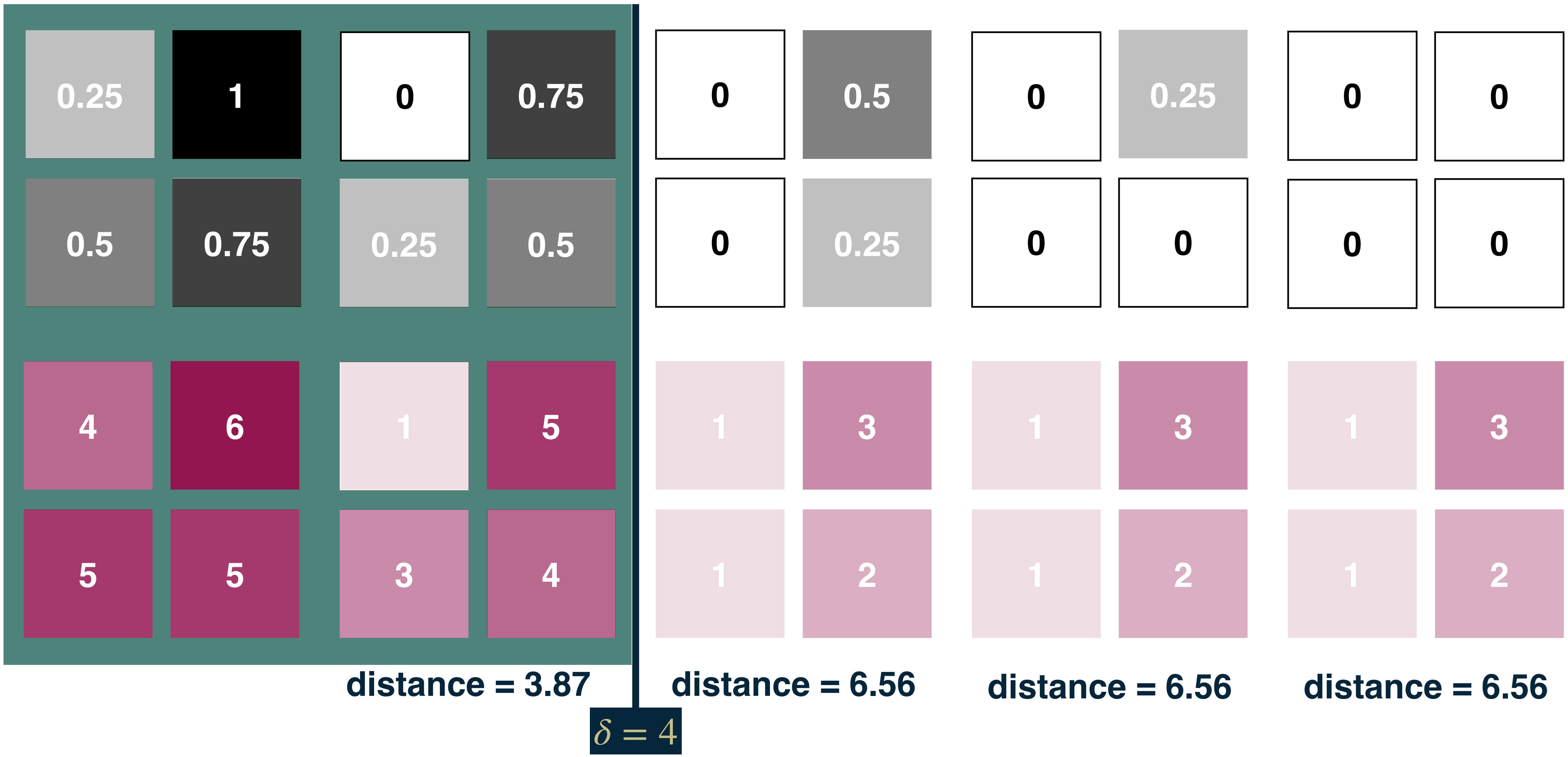
Example

Semantic Perturbations



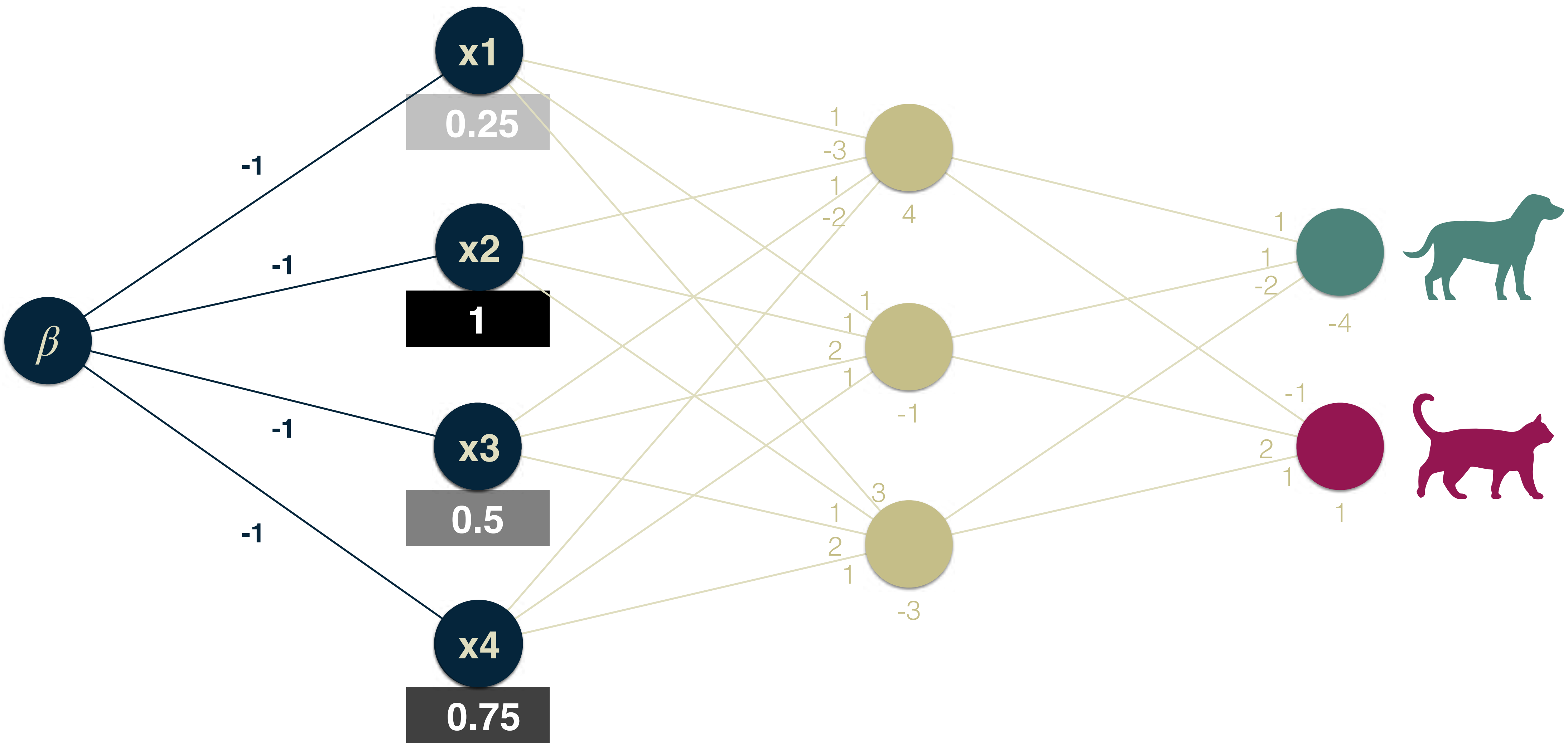
Example

Saliency Map Stability



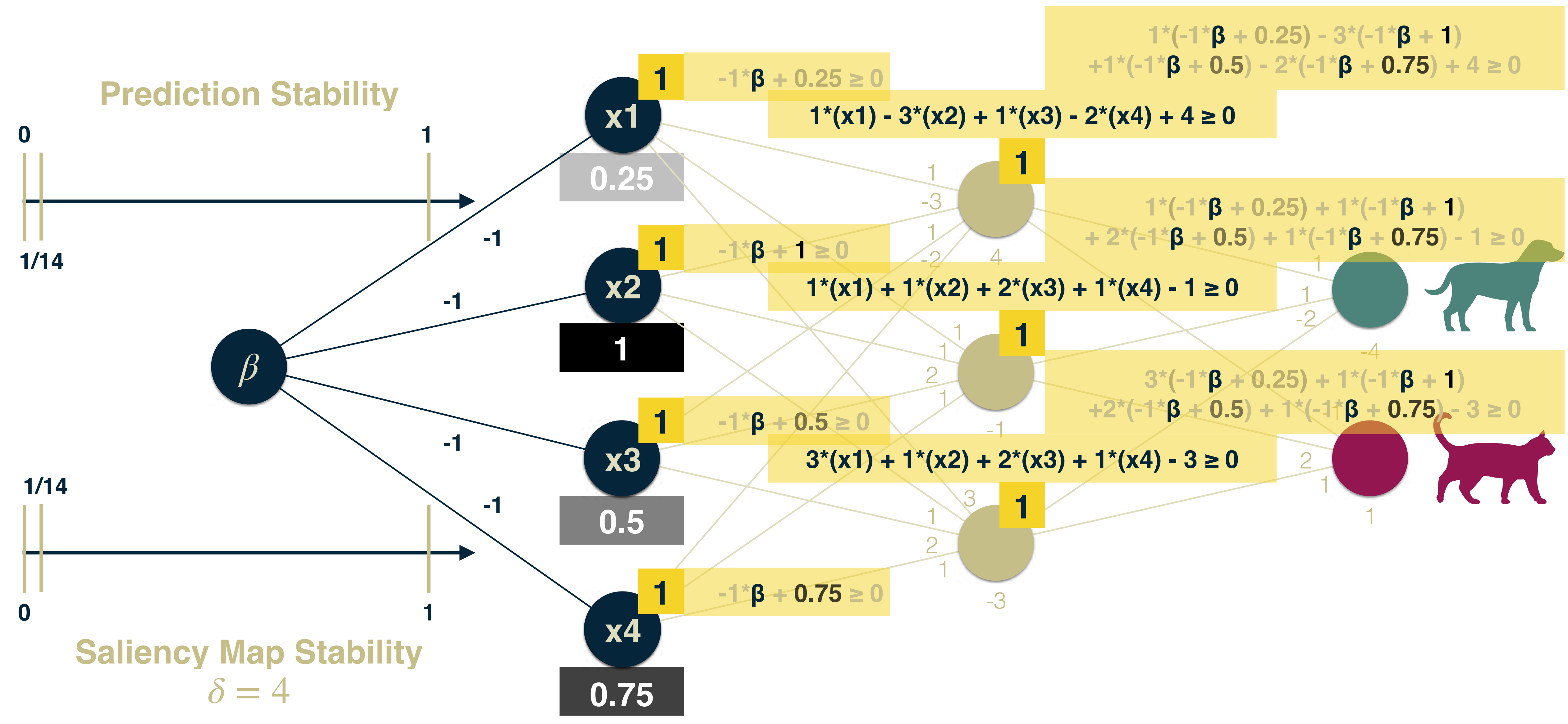
Example

Encoding Semantic Perturbations [Mohapatra20]



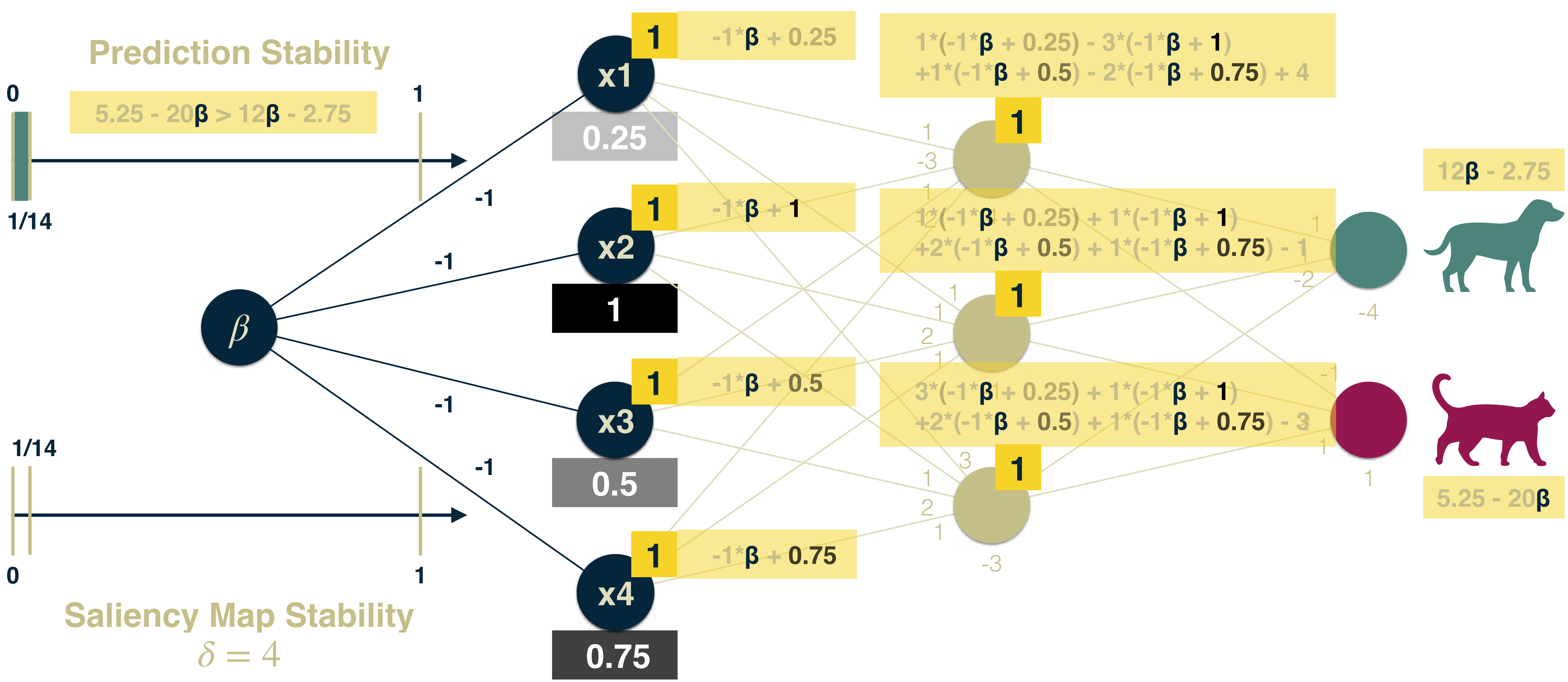
Example

Activation Patterns



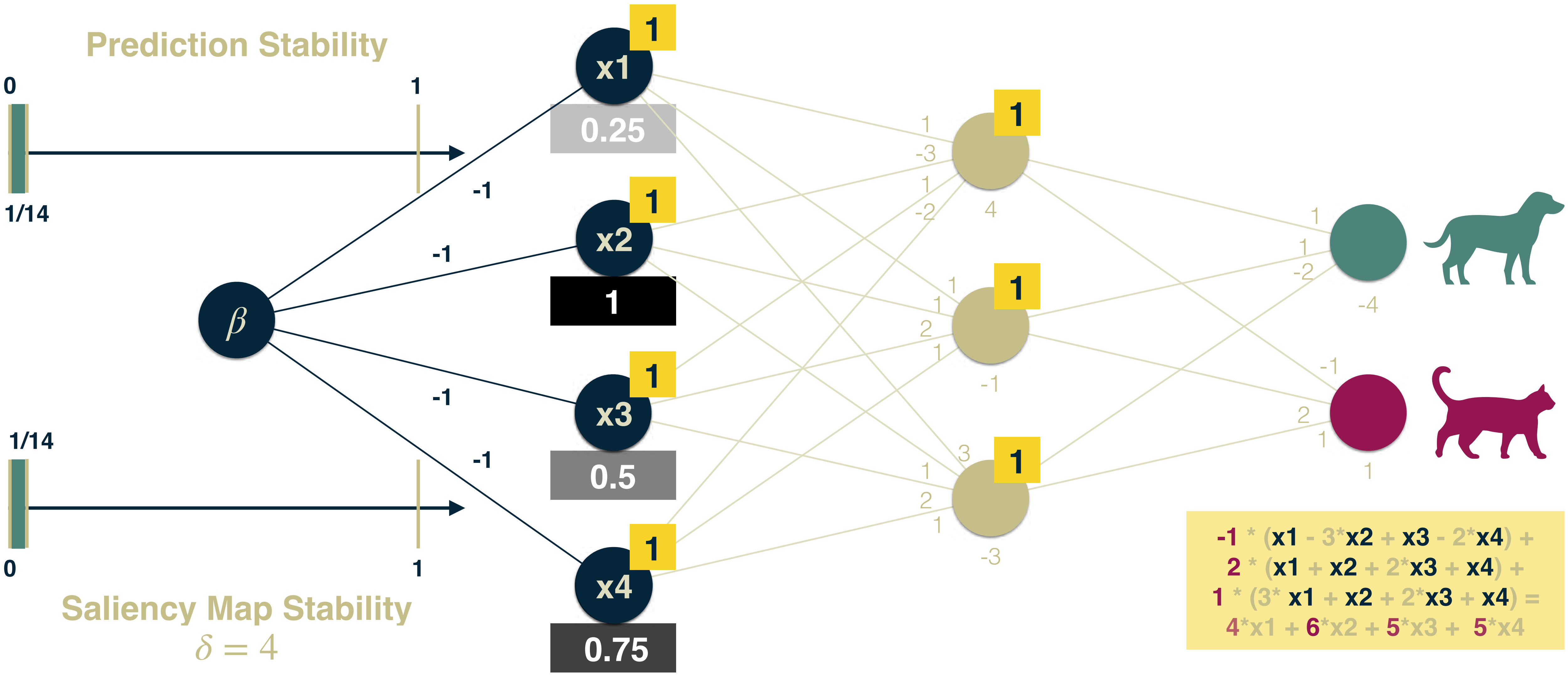
Example

Prediction Stability



Example

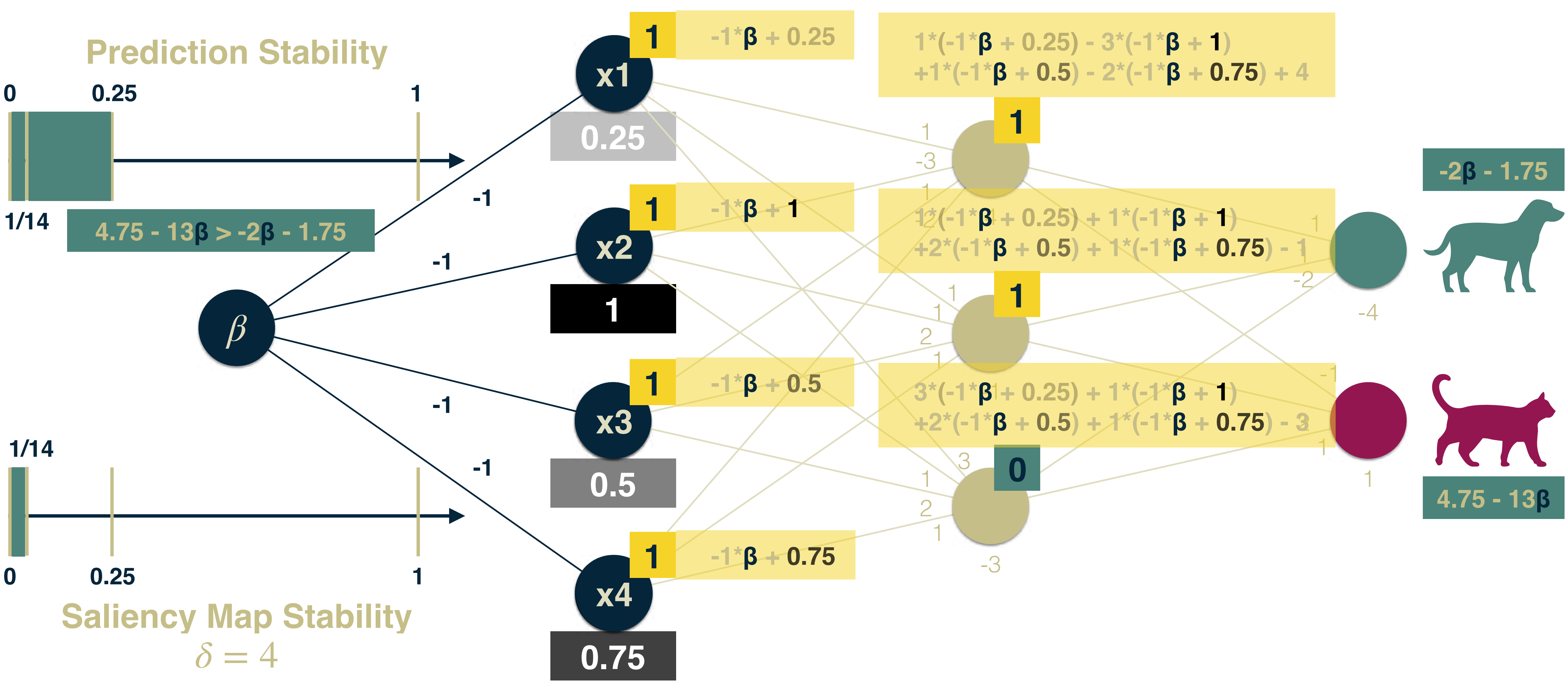
Saliency Map Stability



-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

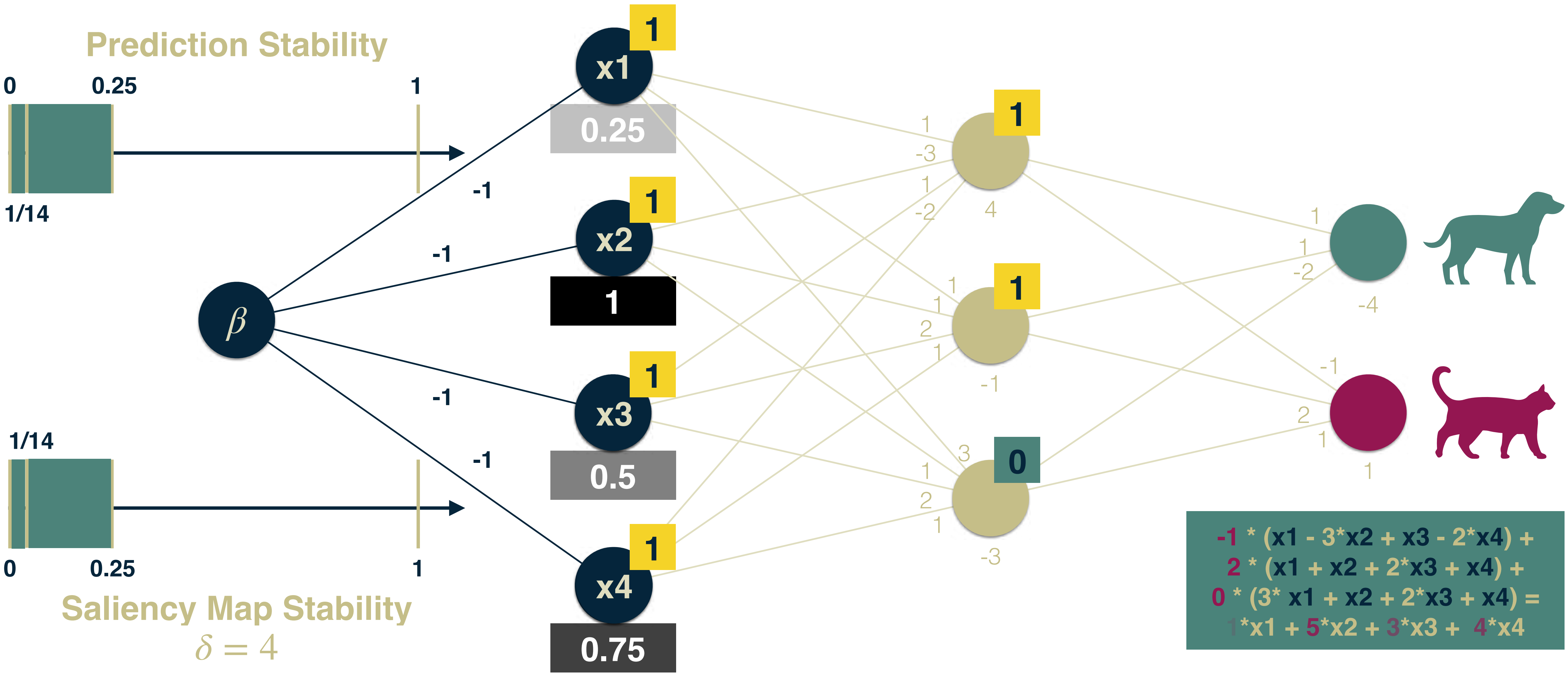
Example

Naïve Breadth-First Search



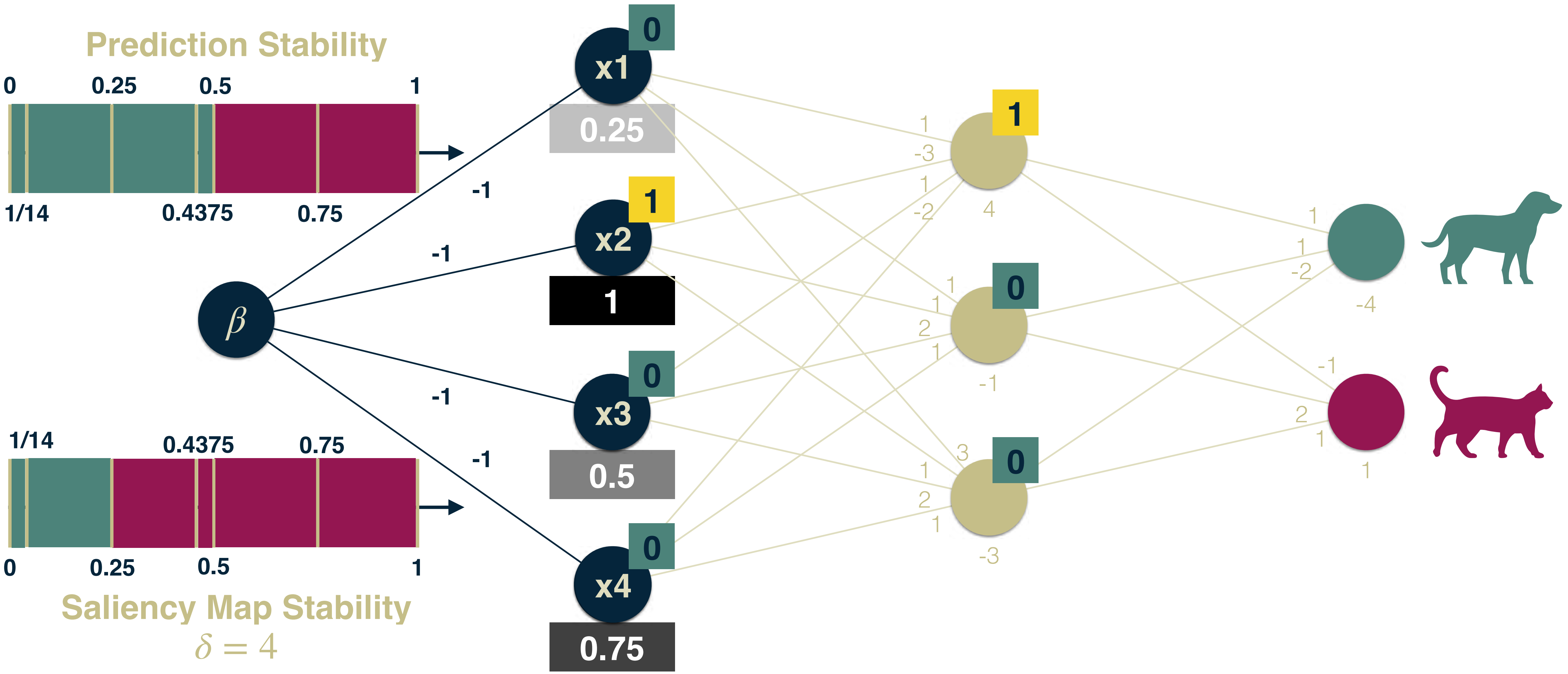
Example

Naïve Breadth-First Search



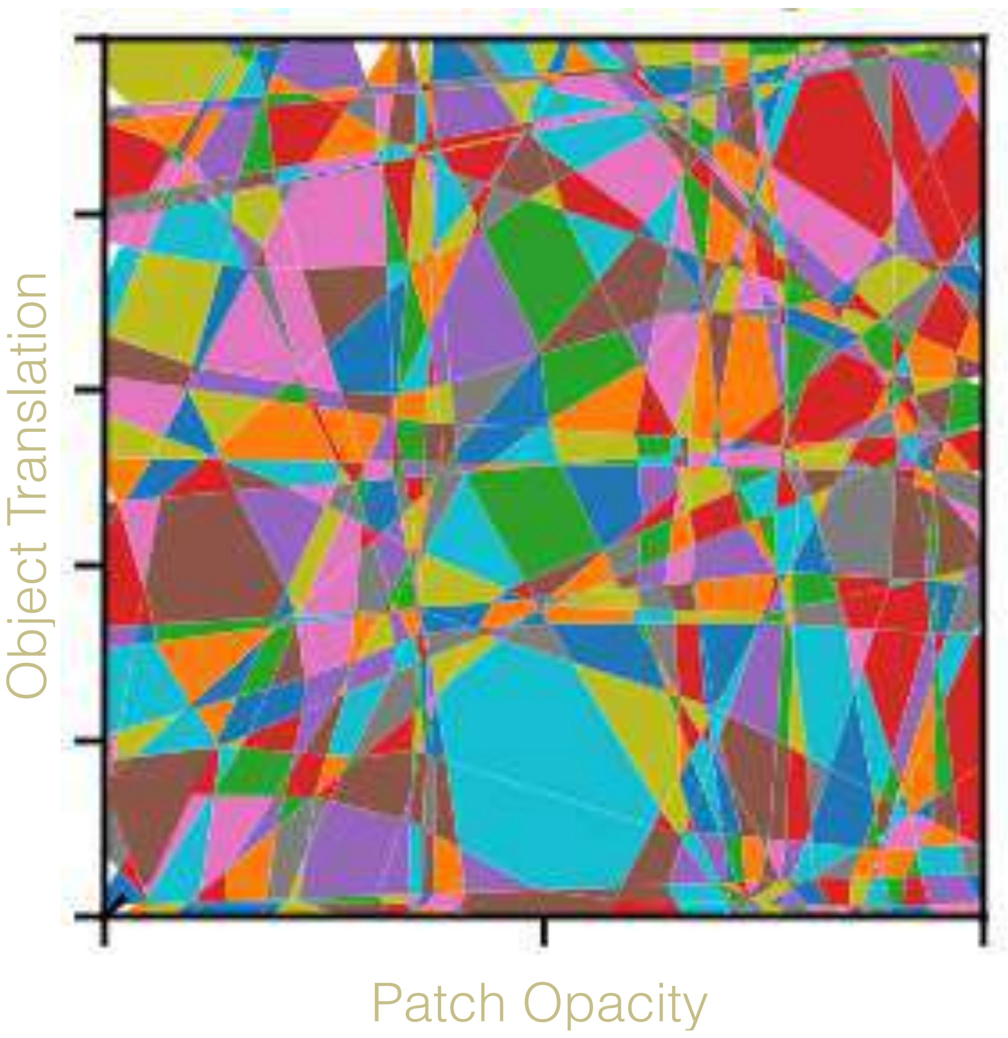
Example

Naïve Breadth-First Search

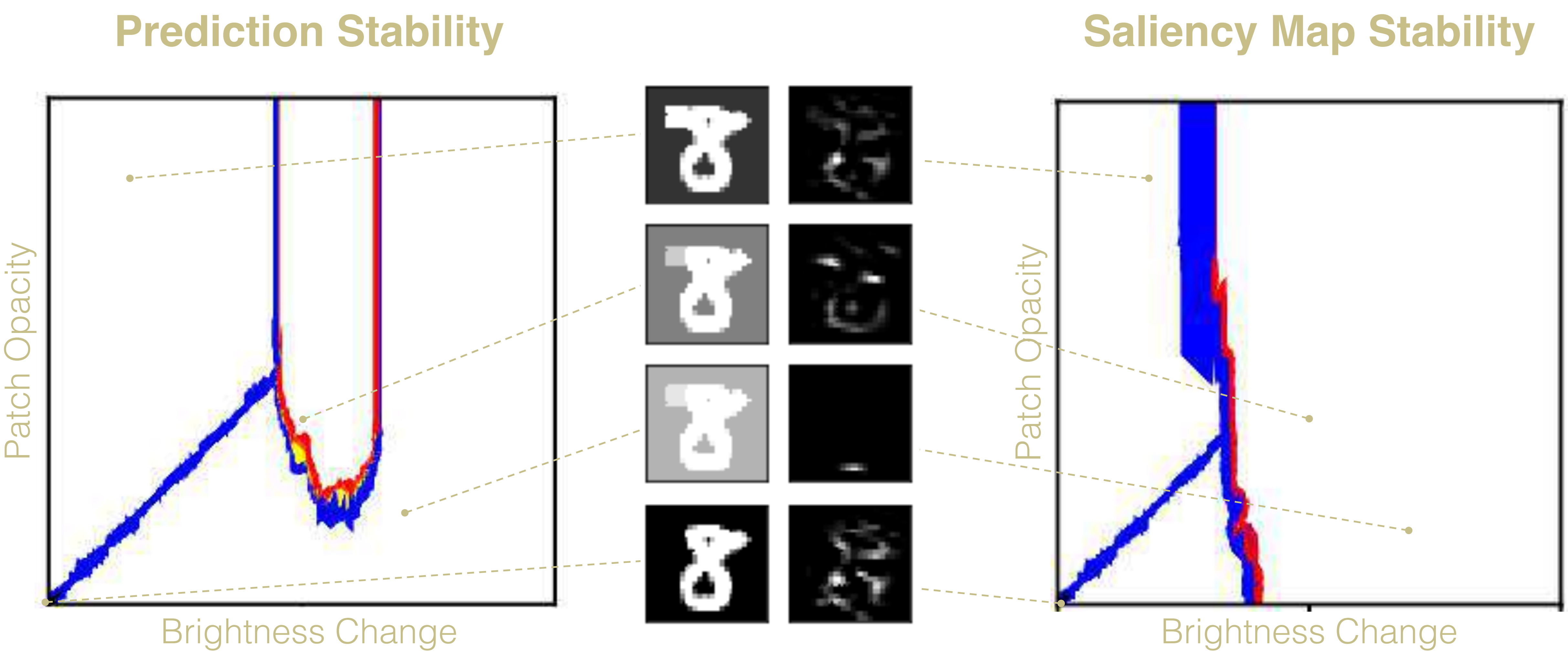


Naïve Breadth-First Search

Too Many Activation Patterns!

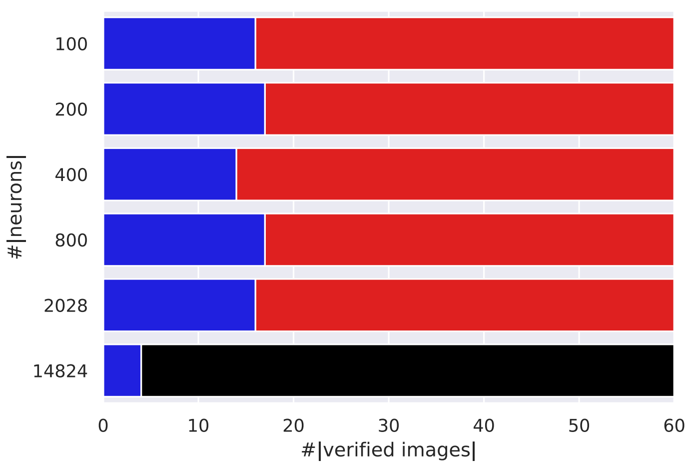


Geometric Boundary Search [Munakata23]

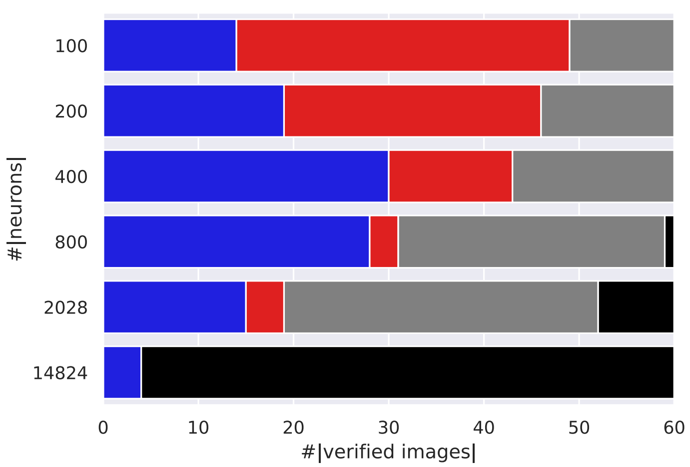


Geometric Boundary Search

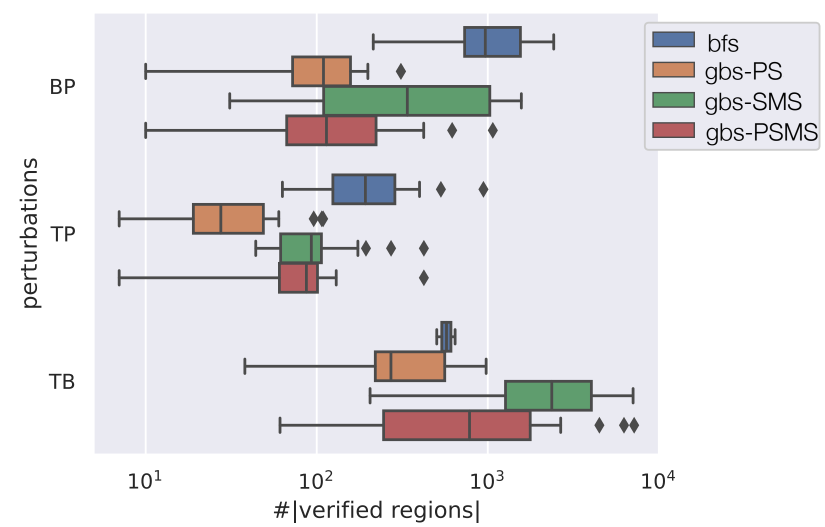
Experimental Results

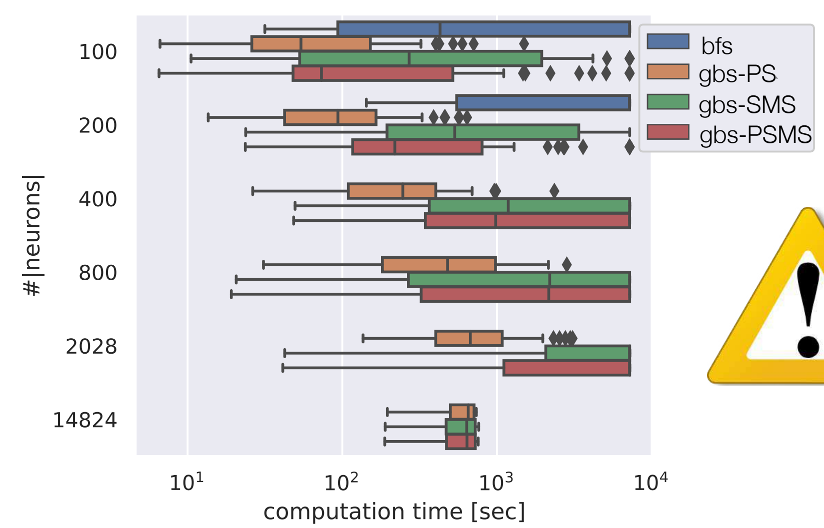


Prediction Stability (PS)



Saliency Map Stability (SMS)

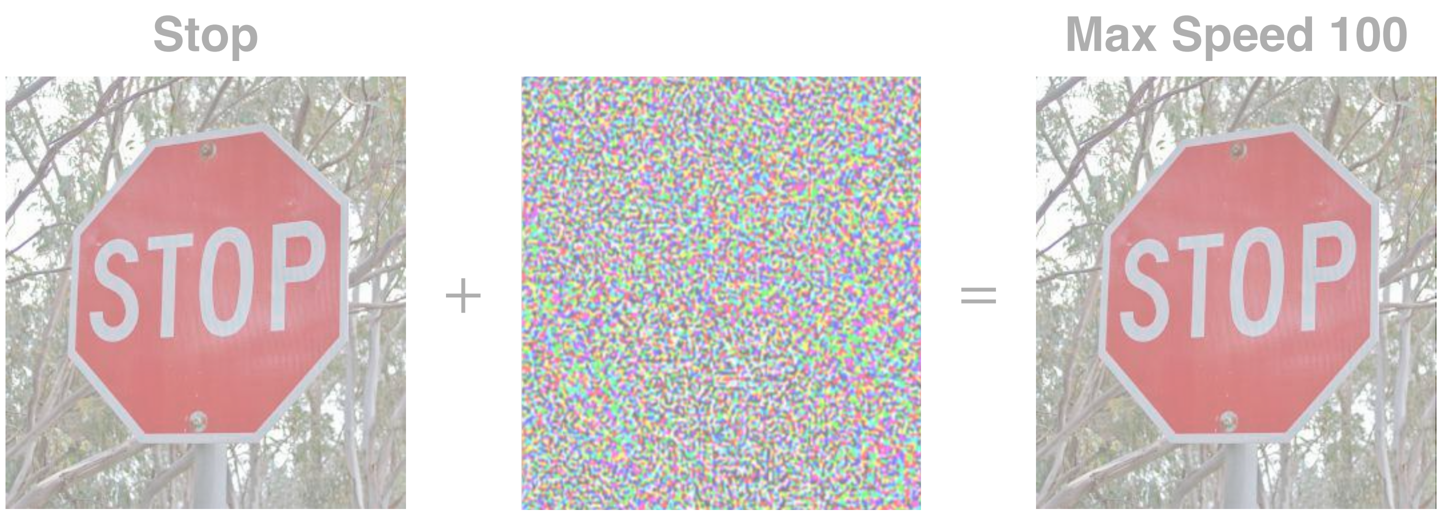




Abstract (Boundary) Search

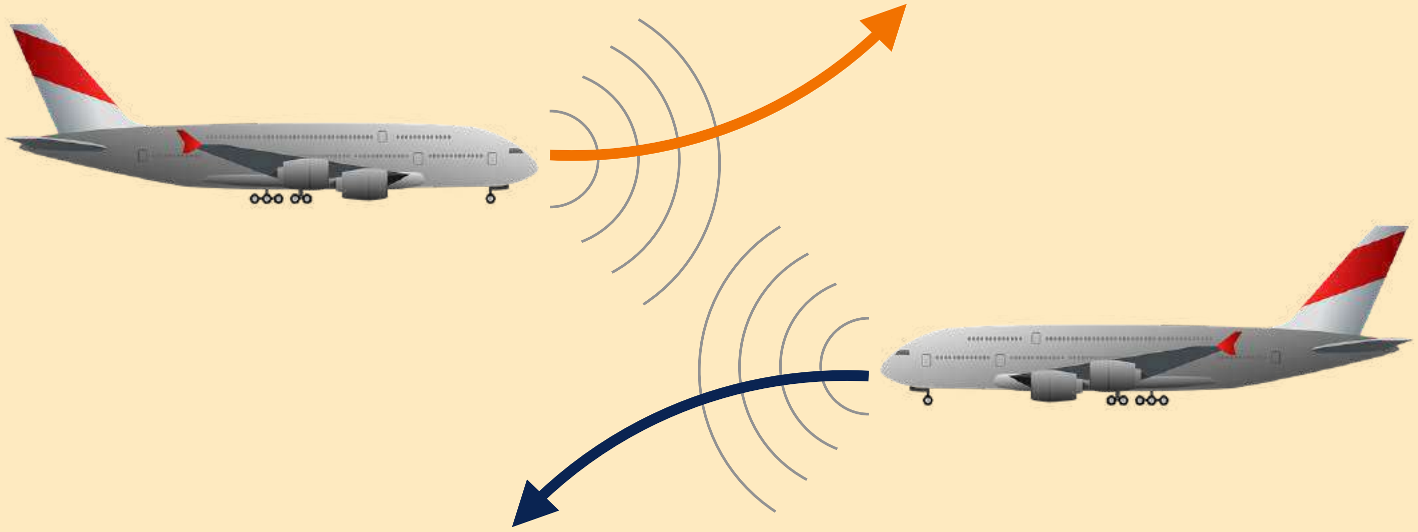
Stability

Goal G3 in [Kurd03]

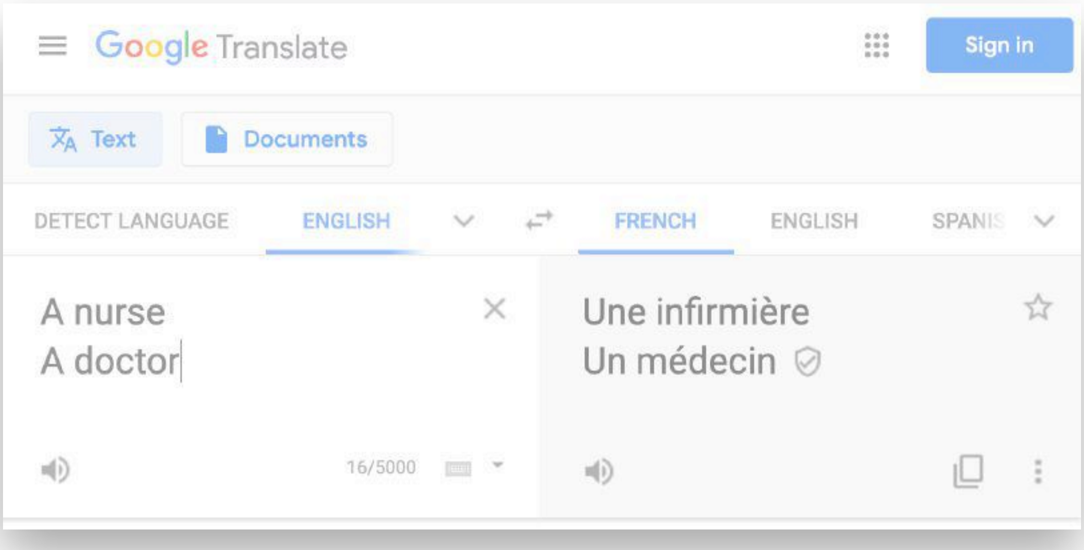


Safety

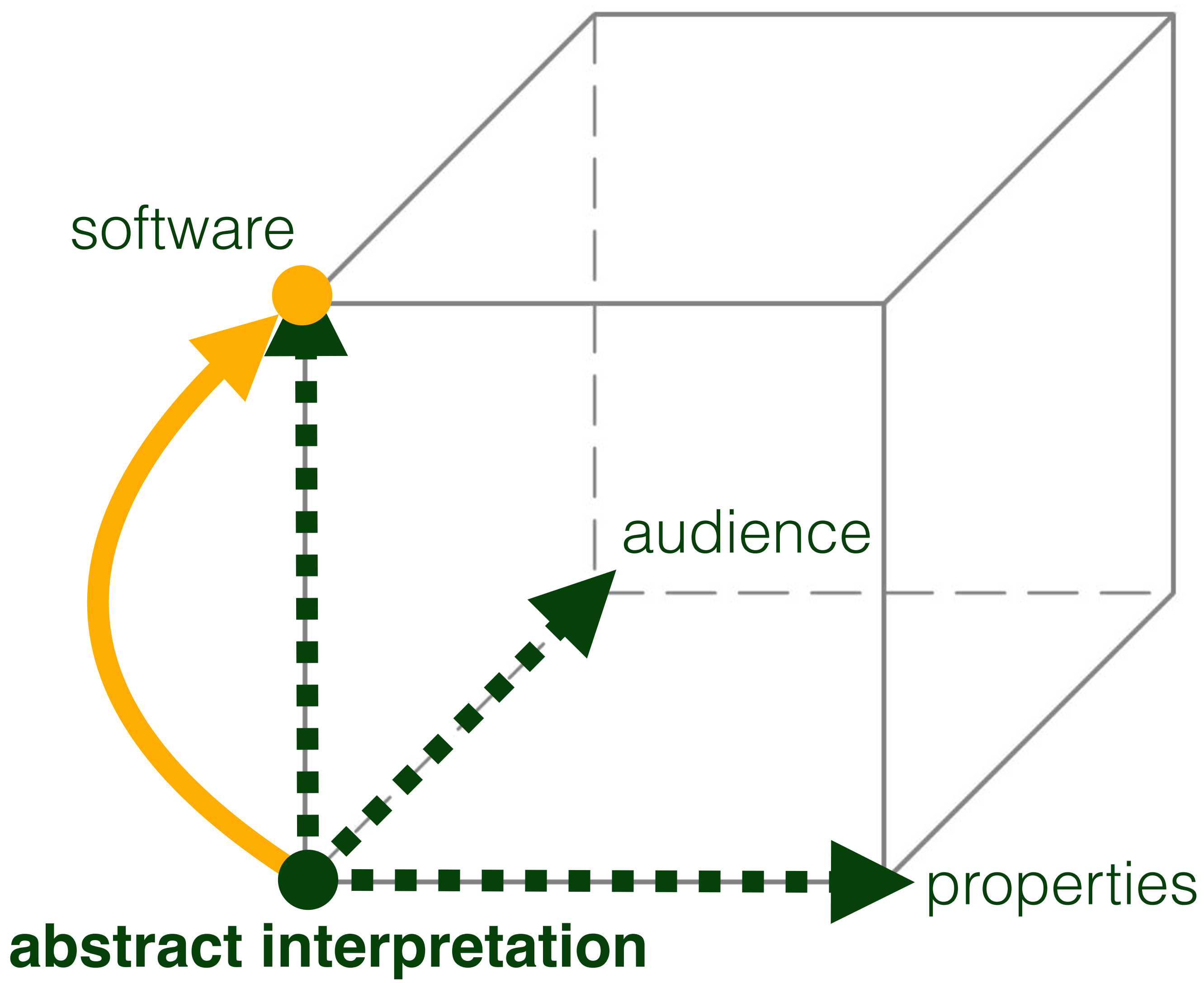
Goal G4 in [Kurd03]



Fairness



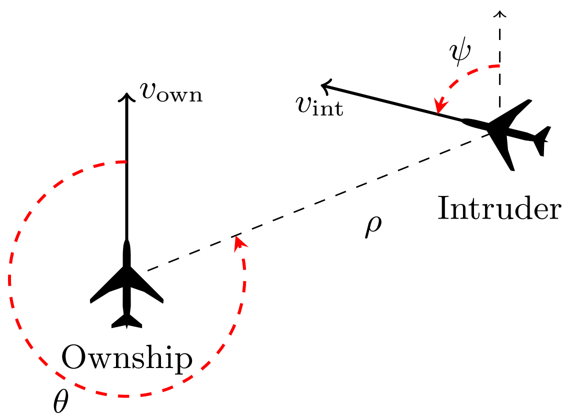
Safety Verification

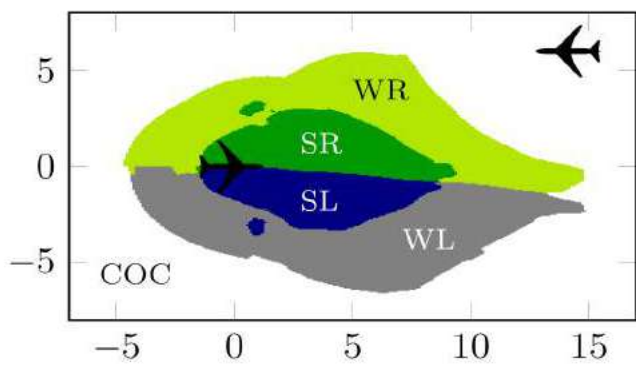


ACAS Xu [Julian16][Katz17]

Airborne Collision Avoidance System for Unmanned Aircraft

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





5 input **sensor measurements**

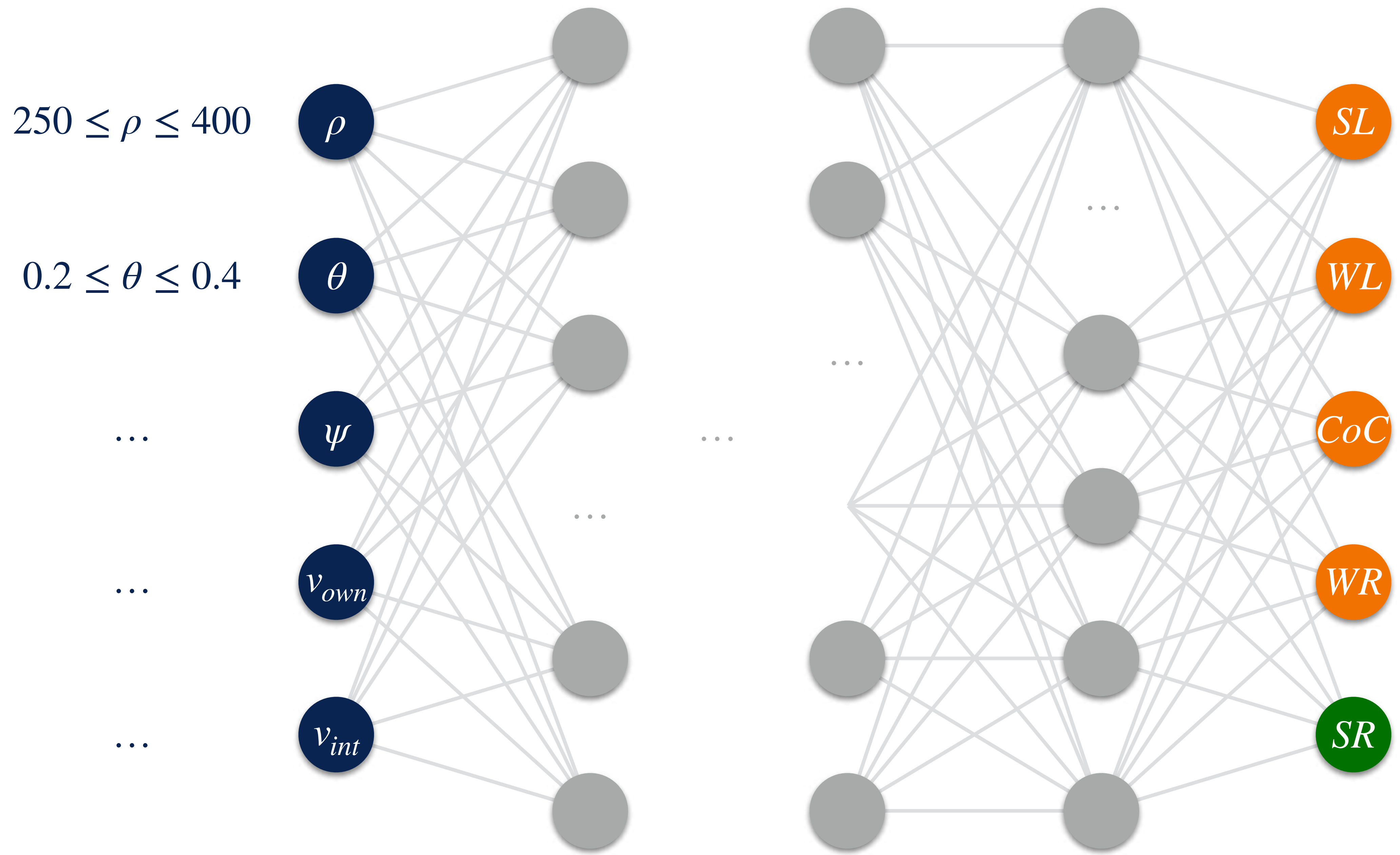
- ρ: distance from ownship to intruder
- θ: angle to intruder relative to ownship heading direction
- ψ: heading angle to intruder relative to ownship heading direction
- v_{own}: speed of ownship
- v_{int}: speed of intruder

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 $\mathbf{O}$

$$\text{SAFE}^{\mathbf{I}}_{\mathbf{O}}([\![M]\!]) \stackrel{\text{def}}{=} \forall t \in [\![M]\!] : t_0 \models \mathbf{I} \Rightarrow t_\omega \models \mathbf{O}$$

Theorem

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

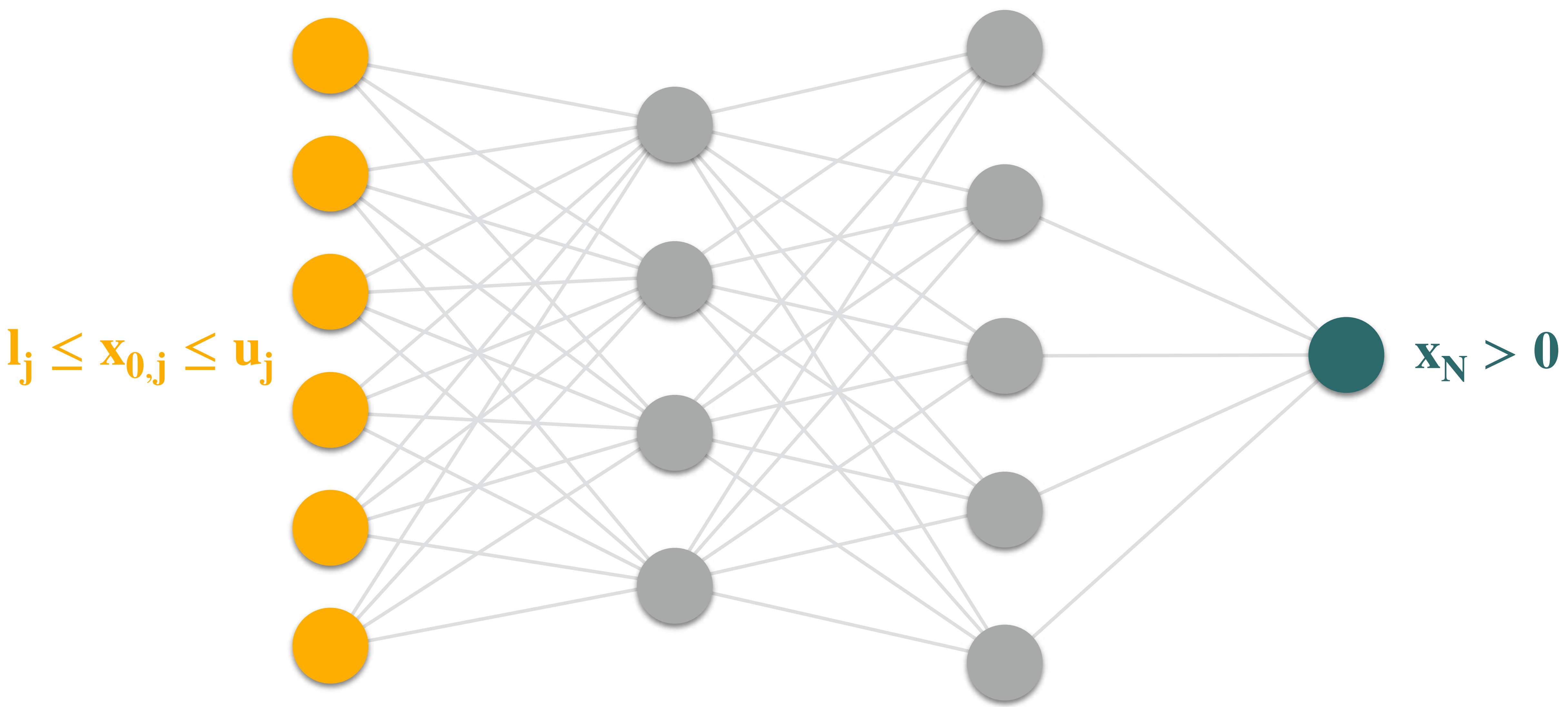
Corollary

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

Model Checking Methods

Safety

Example



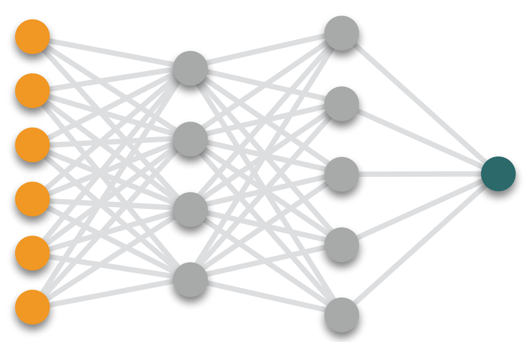
SMT-Based Methods

Verification Reduced to Constraint Satisfiability

$\mathbf{l_j \leq x_{0,j} \leq u_j}$ $\quad j \in \{0, \dots, |\mathbf{X}_0|\}$ **input specification**

$$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j} \qquad i \in \{0, \dots, n-1\}$$

$$x_{i,j} = \max\{0, \hat{x}_{i,j}\} \qquad i \in \{1, \dots, n-1\}, \; j \in \{0, \dots, |\mathbf{X}_i|\}$$



$\mathbf{x_N \leq 0}$ (negation of) **output specification**

satisfiable → ✗ **counterexample**

otherwise → ✓ **safe**

Planet

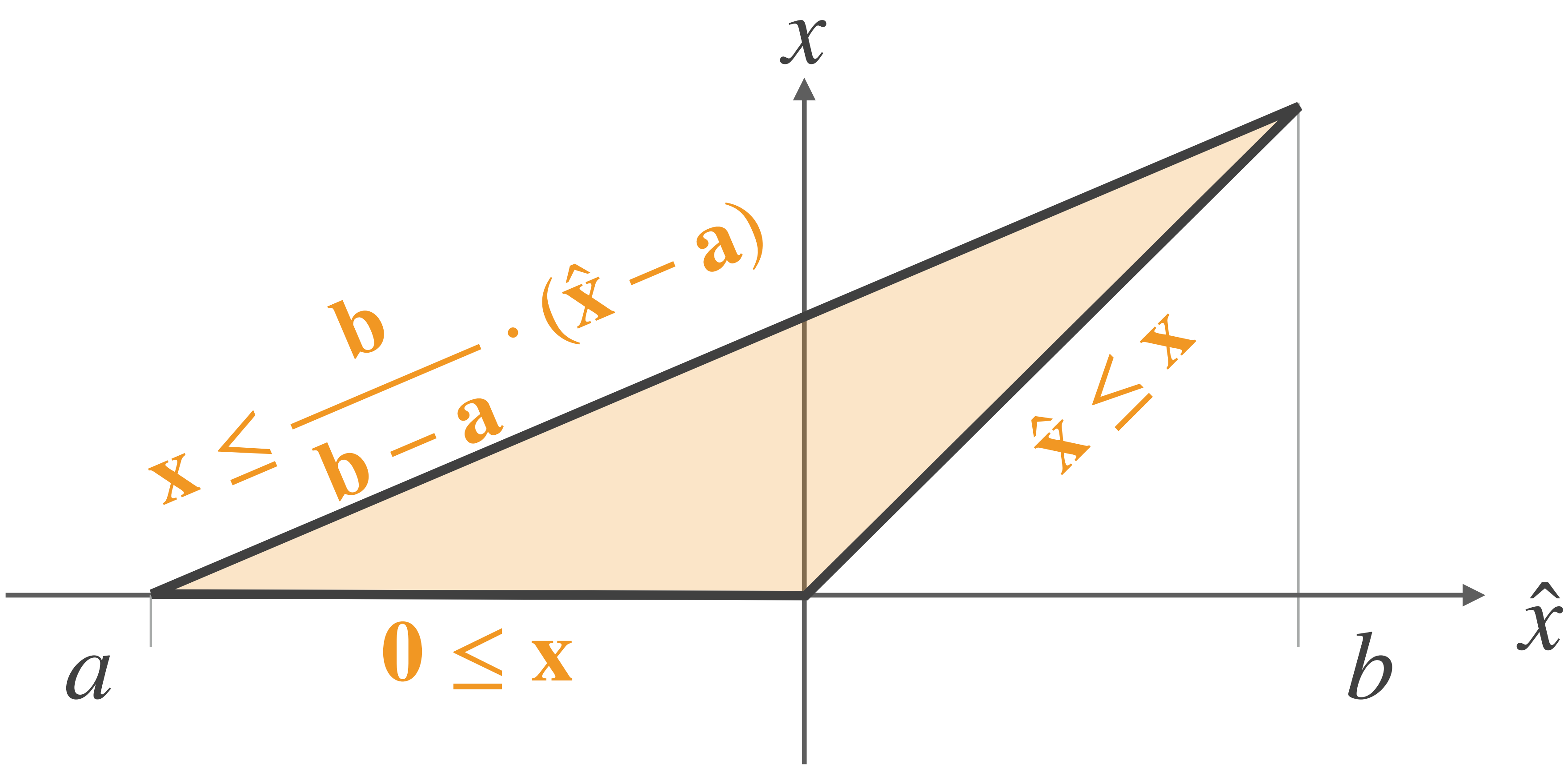
use **approximations** to reduce the solution search space

$$x_{i,j} = \max\{0, \hat{x}_{i,j}\}$$

$$\mathbf{0 \leq x_{i,j}}$$

$$\mathbf{\hat{x}_{i,j} \leq x_{i,j}}$$

$$\mathbf{x_{i,j} \leq \frac{b_{i,j}}{b_{i,j} - a_{i,j}} \cdot (\hat{x}_{i,j} - a_{i,j})}$$



R. Ehlers - Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks (ATVA 2017)

Reluplex

based on the simplex algorithm extended to support ReLUs

Variable	Value
$\mathbf{x_{00}}$	v_{00}
$\dots$	$\dots$
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}_{ij}$
$\mathbf{x_{ij}}$	v_{ij}
$\dots$	$\dots$
$\mathbf{x_N}$	v_N

Variable	Value
$\mathbf{x_{00}}$	v_{00}
$\dots$	$\dots$
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x_{ij}}$	v_{ij}
$\dots$	$\dots$
$\mathbf{x_N}$	v_N

Variable	Value
$\mathbf{x_{00}}$	v_{00}
$\dots$	$\dots$
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x_{ij}}$	$\hat{v}'_{ij}$
$\dots$	$\dots$
$\mathbf{x_N}$	v_N

Variable	Value
$\mathbf{x_{00}}$	v_{00}
$\dots$	$\dots$
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x_{ij}}$	0
$\dots$	$\dots$
$\mathbf{x_N}$	v_N

G. Katz et al. - Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (CAV 2017)

Reluplex

based on th
extended

Follow-up Work

G. Katz et al. - The Marabou Framework for Verification and Analysis of Deep Neural Networks (CAV 2019)

Variable	Value
$\mathbf{x}_{00}$	v_{00}
...	...
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}_{ij}$
$\mathbf{x}_{\mathbf{ij}}$	v_{ij}
...	...
$\mathbf{x}_{\mathbf{N}}$	v_N

Variable	Value
$\mathbf{x}_{00}$	v_{00}
...	...
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x}_{\mathbf{ij}}$	v_{ij}
...	...
$\mathbf{x}_{\mathbf{N}}$	v_N

Variable	
$\mathbf{x}_{00}$	v_{00}
...	...
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
...	...
$\mathbf{x}_{\mathbf{N}}$	v_N

Variable	Value
$\mathbf{x}_{00}$	v_{00}
...	...
$\hat{\mathbf{x}}_{\mathbf{ij}}$	$\hat{v}'_{ij}$
$\mathbf{x}_{\mathbf{ij}}$	0
...	...
$\mathbf{x}_{\mathbf{N}}$	v_N

G. Katz et al. - Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks (CAV 2017)

Other SMT-Based Methods

- **L. Pulina and A. Tacchella**. *An Abstraction-Refinement Approach to Verification of Artificial Neural Networks*. In CAV, 2010.
  the first formal verification method for neural networks

- **O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi**. *Measuring Neural Net Robustness with Constraints*. In NeurIPS, 2016.
  an approach for finding the nearest adversarial example according to the L_∞ distance

- **X. Huang, M. Kwiatkowska, S. Wang, and M. Wu**. *Safety Verification of Deep Neural Networks*. In CAV, 2017.
  an approach for proving local robustness to adversarial perturbations

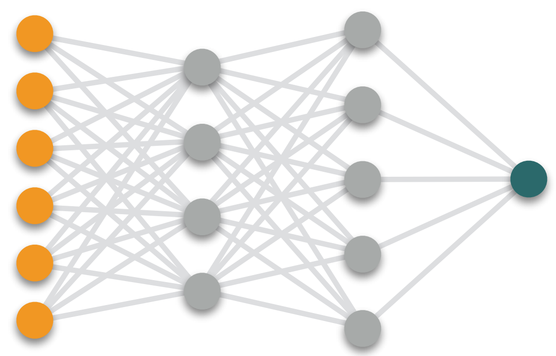
- **N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh**. *Verifying Properties of Binarized Deep Neural Networks*. In AAAI, 2018.
  C. H. Cheng, G. Nührenberg, C. H. Huang, and H. Ruess. *Verification of Binarized Neural Networks via Inter-Neuron Factoring*. In VSTTE, 2018.
  approaches focusing on binarized neural networks

MILP-Based Methods

Verification Reduced to Mixed Integer Linear Program

$\mathbf{l_j} \leq \mathbf{x_{0,j}} \leq \mathbf{u_j}$ $\quad j \in \{0, \ldots, |\mathbf{X}_0|\}$ **input specification**

$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j}$ $\quad i \in \{0, \ldots, n-1\}$



$x_{i,j} = \delta_{\mathbf{i,j}} \cdot \hat{x}_{i,j}$ $\quad \delta_{\mathbf{i,j}} \in \{\mathbf{0}, \mathbf{1}\}$

$\delta_{\mathbf{i,j}} = 1 \Rightarrow \hat{x}_{i,j} \geq 0$ $\quad i \in \{1, \ldots, n-1\}$

$\delta_{\mathbf{i,j}} = 0 \Rightarrow \hat{x}_{i,j} < 0$ $\quad j \in \{0, \ldots, |\mathbf{X}_i|\}$

min $\mathbf{x_N}$ **objective function**

min $\mathbf{x_N} \leq \mathbf{0} \rightarrow$ ✗ **counterexample**

otherwise $\rightarrow$ ✓ **safe**

MILP-Based Methods

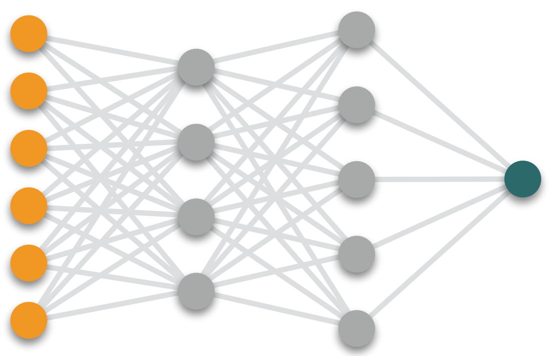
Bounded Encoding with Symmetric Bounds

$$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j} \qquad i \in \{0, \ldots, n-1\}$$

$$0 \leq x_{i,j} \leq \mathbf{M_{i,j}} \cdot \delta_{i,j} \qquad \delta_{\mathbf{i,j}} \in \{\mathbf{0}, \mathbf{1}\}$$

$$\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - \mathbf{M_{i,j}} \cdot (1 - \delta_{i,j}) \qquad i \in \{1, \ldots, n-1\}$$

$$\mathbf{M_{i,j}} = \mathbf{max}\{-\mathbf{l_i}, \mathbf{u_i}\} \qquad j \in \{0, \ldots, |\mathbf{X}_i|\}$$



Sherlock

Output Range Analysis

use **local search** to speed up the MILP solver

$\mathbf{l_j \leq x_{0,j} \leq u_j}$

$$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j}$$

$$0 \leq x_{i,j} \leq \mathbf{M_{i,j}} \cdot \delta_{i,j}$$

$$\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - \mathbf{M_{i,j}} \cdot (1 - \delta_{i,j})$$

$$\mathbf{M_{i,j}} = \mathbf{max}\{-\mathbf{l_i}, \mathbf{u_i}\}$$

$$\mathbf{x_N} < \hat{\mathbf{L}}$$

find another input $\hat{\mathbf{X}}$ such that $\hat{\mathbf{L}} \leq \mathbf{x_N}$

S. Dutta et al. - Output Range Analysis for Deep Feedforward Neural Networks (NFM 2018)

MILP-Based Methods

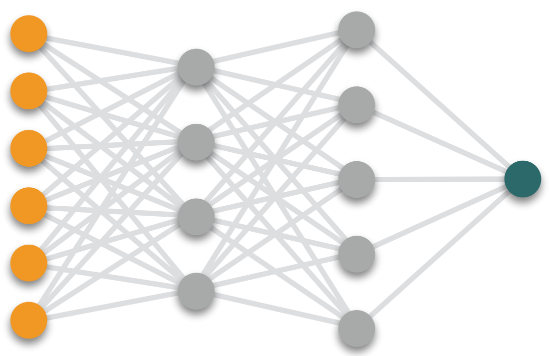
Bounded Encoding with Asymmetric Bounds

$$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j} \qquad i \in \{0,\ldots,n-1\}$$

$$0 \leq x_{i,j} \leq \mathbf{u_{i,j}} \cdot \delta_{i,j} \qquad \delta_{\mathbf{i,j}} \in \{\mathbf{0,1}\}$$

$$\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - \mathbf{l_{i,j}} \cdot (1 - \delta_{i,j}) \qquad i \in \{1,\ldots,n-1\}$$

$$j \in \{0,\ldots,|\mathbf{X}_i|\}$$



MIPVerify

Finding Nearest Adversarial Example

$\min_{\mathbf{X}'} \mathbf{d}(\mathbf{X}, \mathbf{X}')$

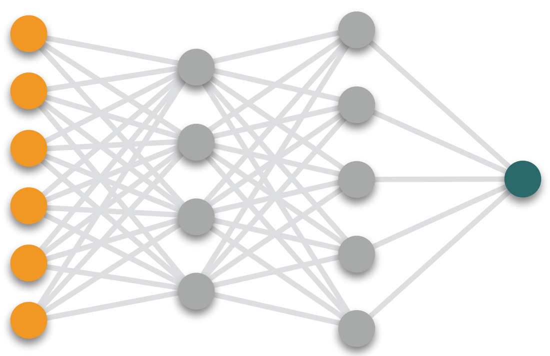
$$\hat{x}_{i+1,j} = \sum_{k=0}^{|\mathbf{X}_i|} w^i_{j,k} \cdot x_{i,k} + b_{i,j} \qquad i \in \{0, \ldots, n-1\}$$

$$\delta_{\mathbf{i,j}} \in \{\mathbf{0}, \mathbf{1}\}$$

$$0 \leq x_{i,j} \leq \mathbf{u_{i,j}} \cdot \delta_{i,j} \qquad i \in \{1, \ldots, n-1\}$$

$$\hat{x}_{i,j} \leq x_{i,j} \leq \hat{x}_{i,j} - \mathbf{l_{i,j}} \cdot (1 - \delta_{i,j}) \qquad j \in \{0, \ldots, |\mathbf{X}_i|\}$$

$$\mathbf{x_N} \neq \mathbf{O}$$



V. Tjeng et al. - Evaluating Robustness of Neural Networks with Mixed Integer Programming (ICLR 2019)

Other MILP-Based Methods

- **R. Bunel, I. Turkaslan, P. H. S. Torr, P. Kohli, and M. P. Kumar**. *A Unified View of Piecewise Linear Neural Network Verification*. In NeurIPS, 2018.
  a unifying verification framework for piecewise-linear ReLU neural networks

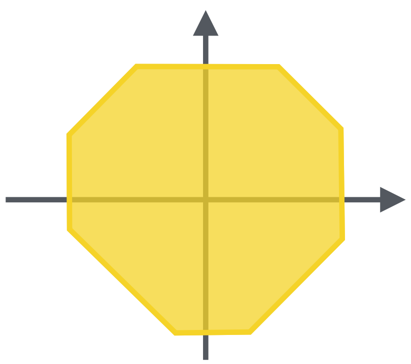
- **C.-H. Cheng, G. Nührenberg, and H. Ruess**. *Maximum Resilience of Artificial Neural Networks*. In ATVA, 2017.
  an approach for finding a lower bound on robustness to adversarial perturbations

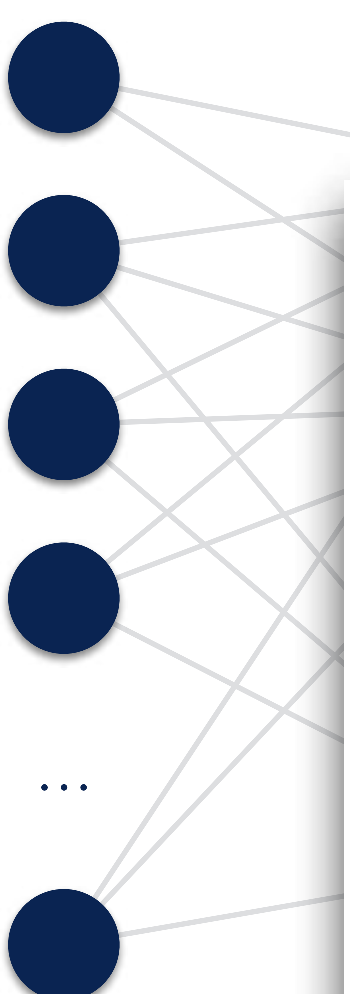
- **M. Fischetti and J. Jo**. *Deep Neural Networks and Mixed Integer Linear Optimization*. 2018.
  an approach for feature visualization and building adversarial examples

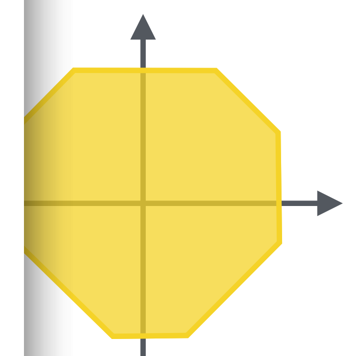
Static Analysis Methods

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 $\mathbf{I}$

Safety

Input-Output Properties

$\mathbf{I}$: input specification

$\mathbf{O}$: output specification

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

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

$\mathsf{SAFE}^{\mathbf{I}}_{\mathbf{O}}([\![M]\!]) \stackrel{\text{def}}{=} \forall t \in [\![M]\!]: t_0 \models \mathbf{I} \Rightarrow t_\omega \models \mathbf{O}$

Theorem

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

Corollary

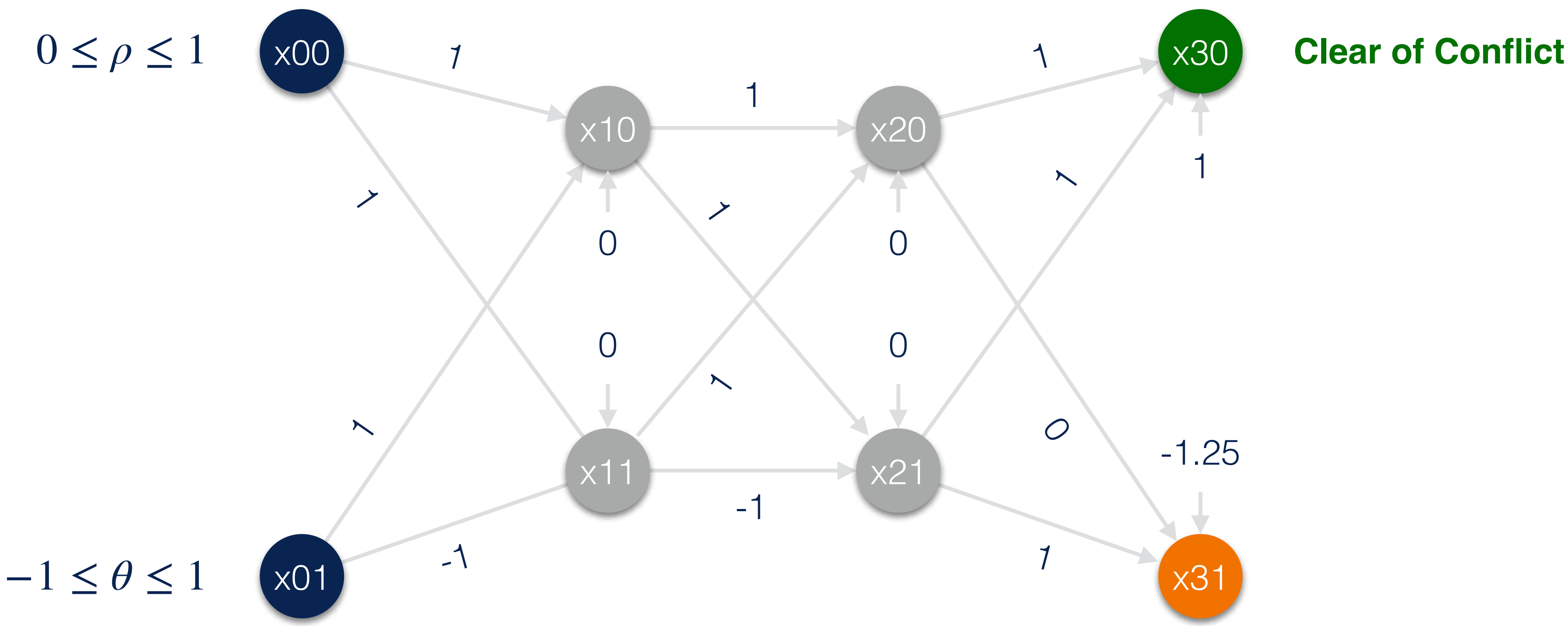
$M \models \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Leftrightarrow [\![M]\!] \subseteq \bigcup \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$

66

Theorem

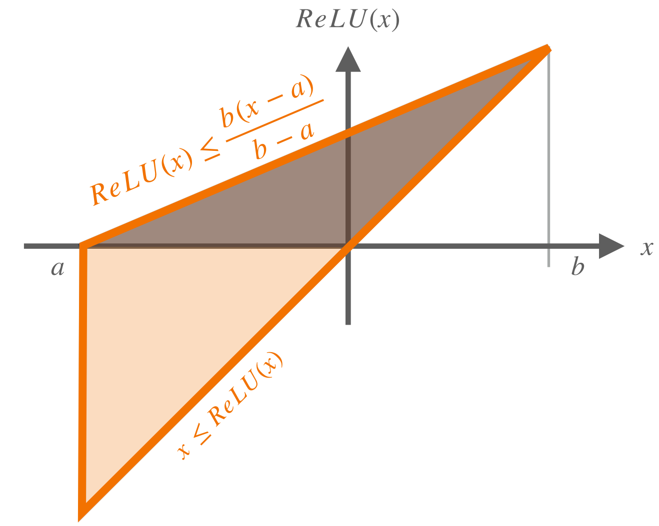
$$[\![M]\!] \subseteq [\![M]\!]^\natural \subseteq \bigcup \mathcal{S}^{\mathbf{I}}_{\mathbf{O}} \Rightarrow M \models \mathcal{S}^{\mathbf{I}}_{\mathbf{O}}$$

Example



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

DeepPoly[Singh19]



ReLU

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

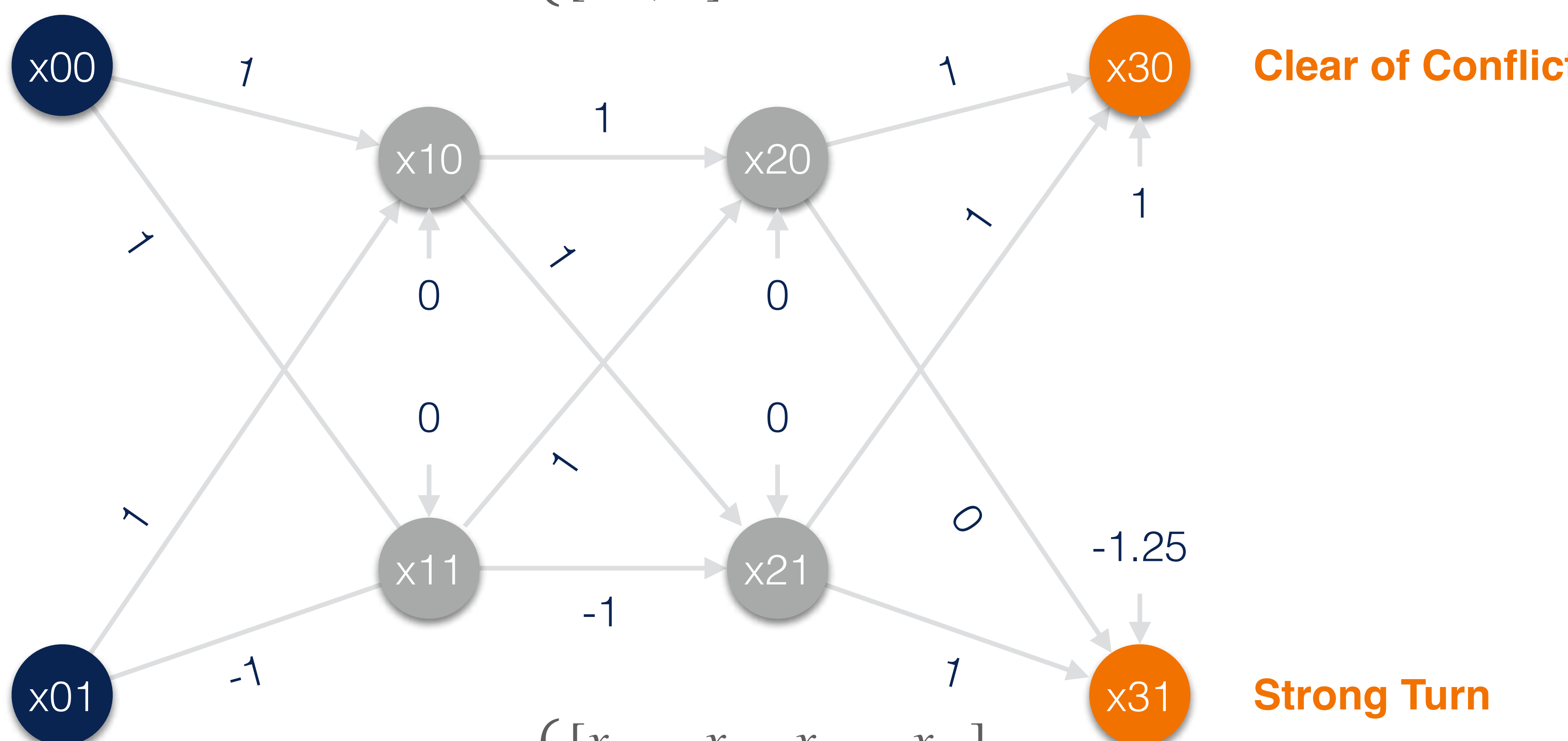
$$x_{10} \mapsto \begin{cases} [x_{00} + x_{01},\ x_{00} + x_{01}] \\ [\mathbf{-1},\ \mathbf{2}] \end{cases}$$

$0 \leq \rho \leq 1$

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

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

$-1 \leq \theta \leq 1$

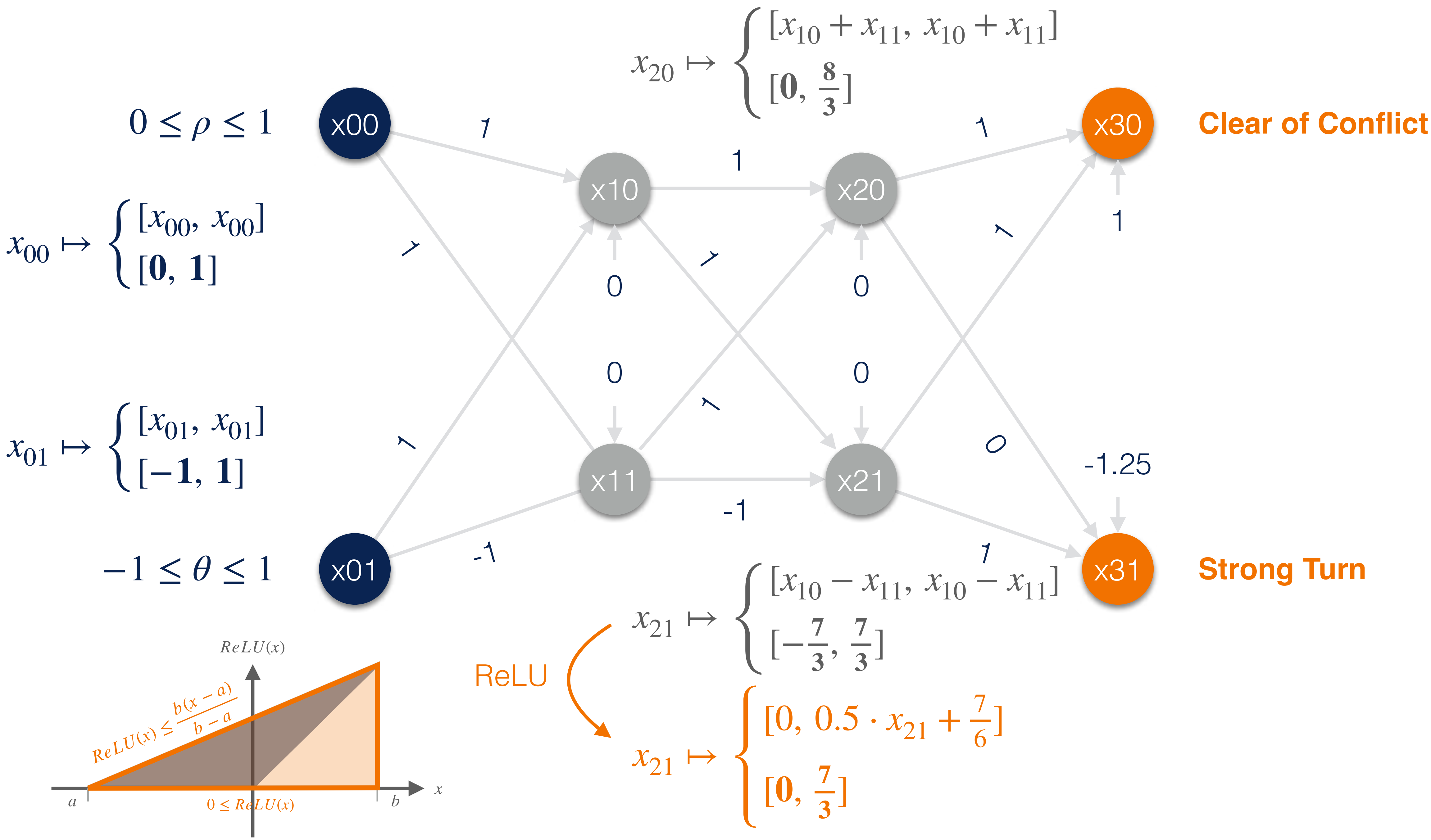


$$x_{11} \mapsto \begin{cases} [x_{00} - x_{01},\ x_{00} - x_{01}] \\ [\mathbf{-1},\ \mathbf{2}] \end{cases}$$

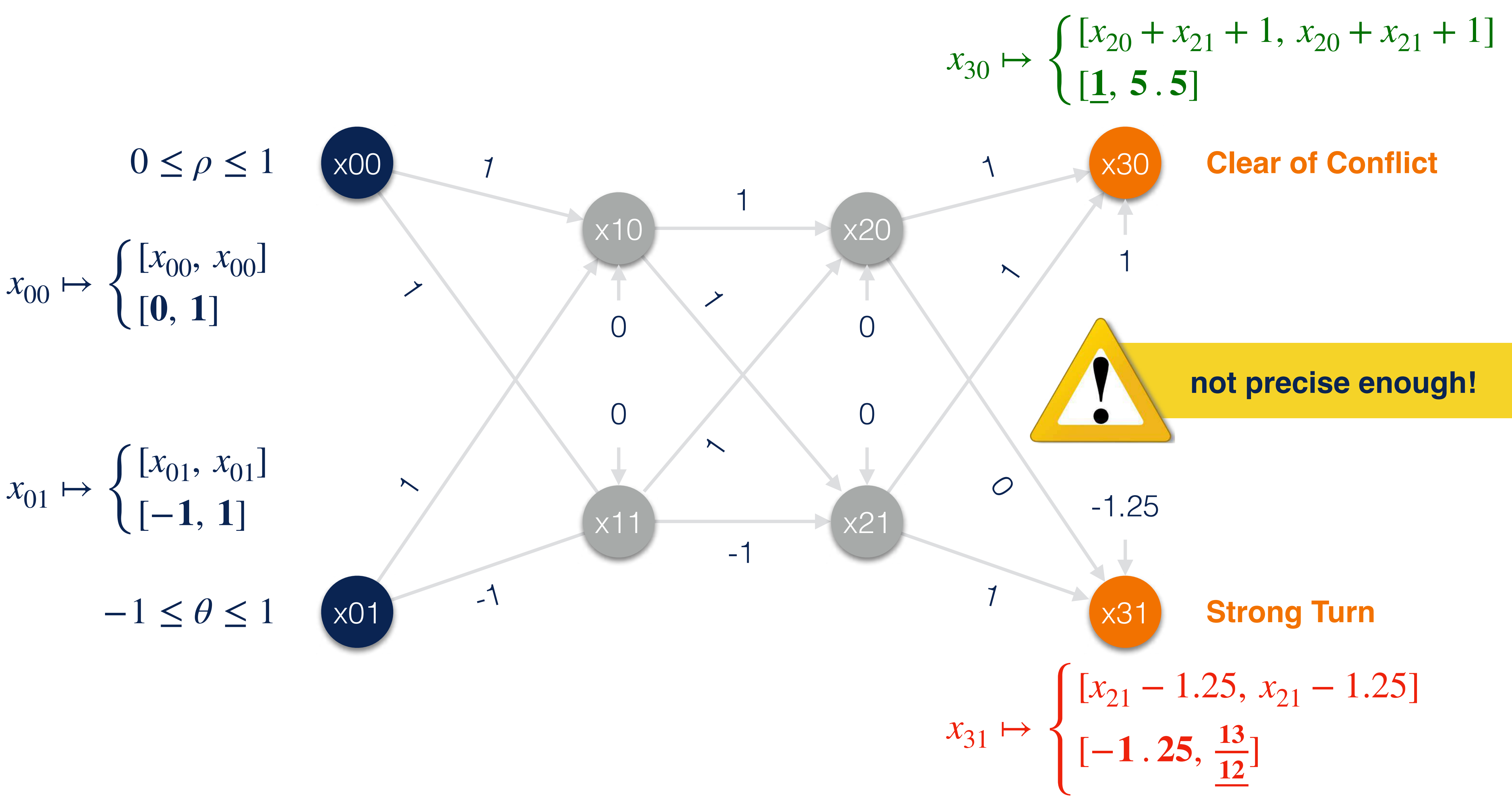
ReLU

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

DeepPoly[Singh19]

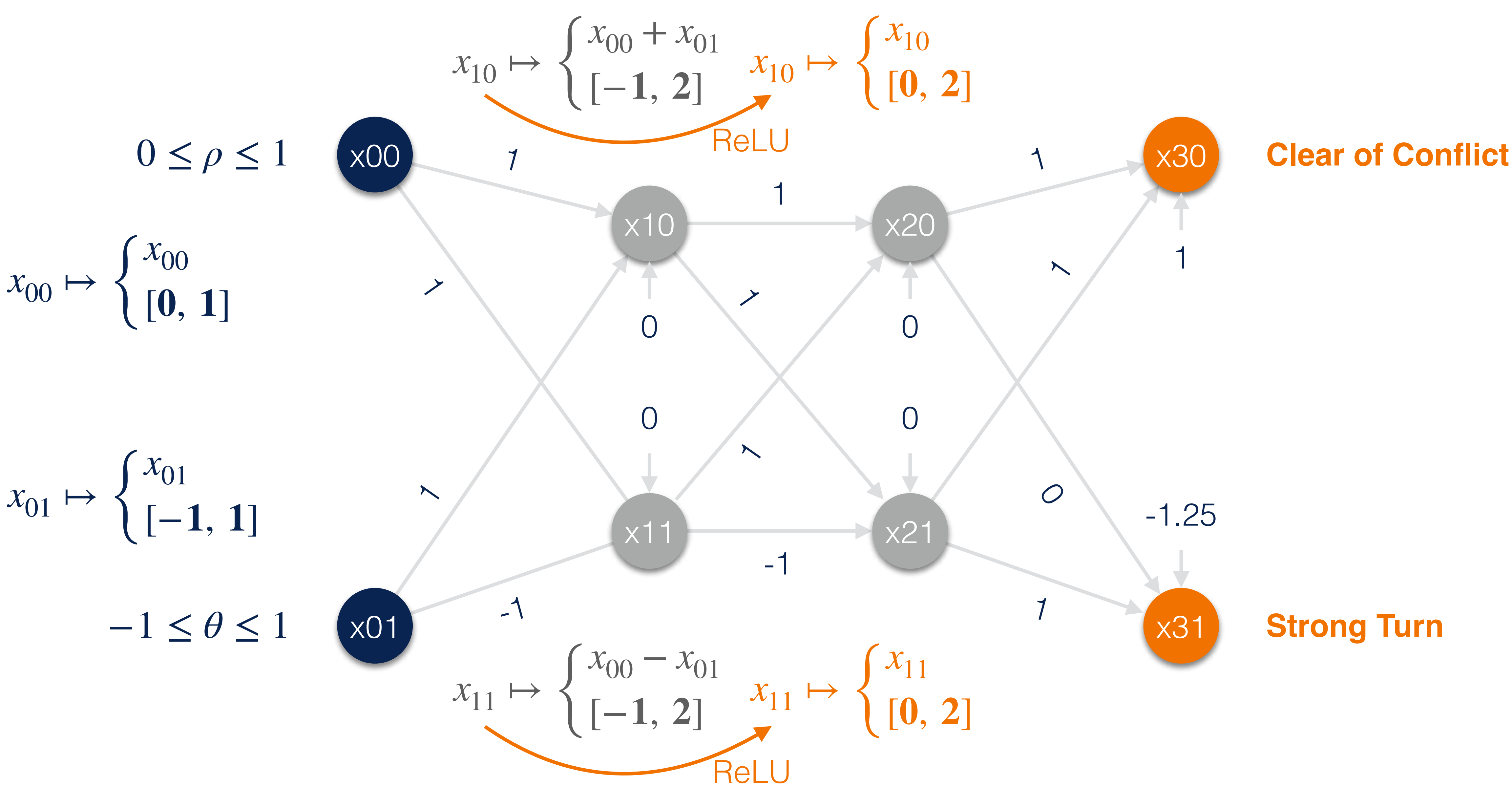


DeepPoly[Singh19]



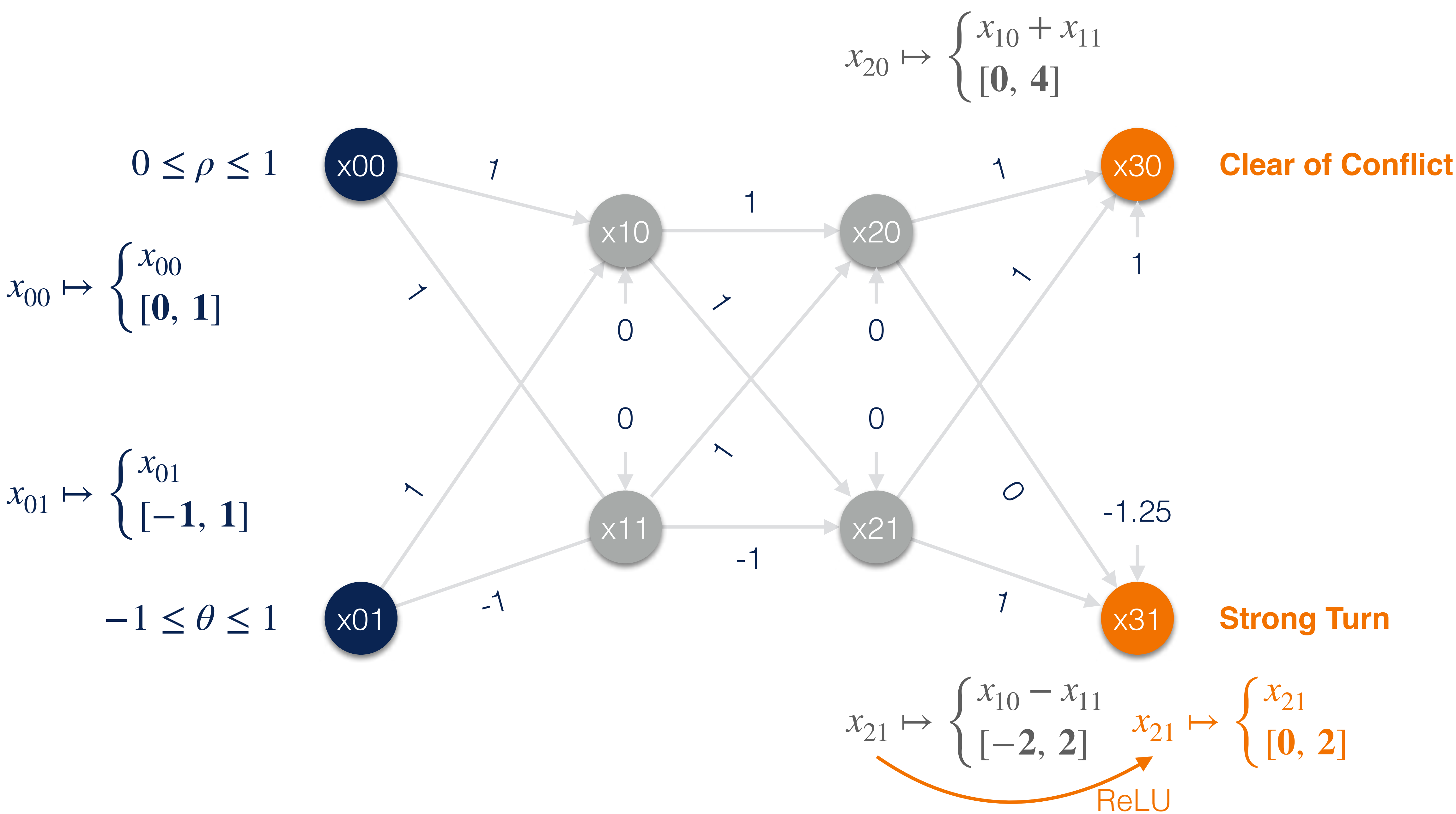
Interval Abstraction

with Symbolic Constant Propagation [Li19]



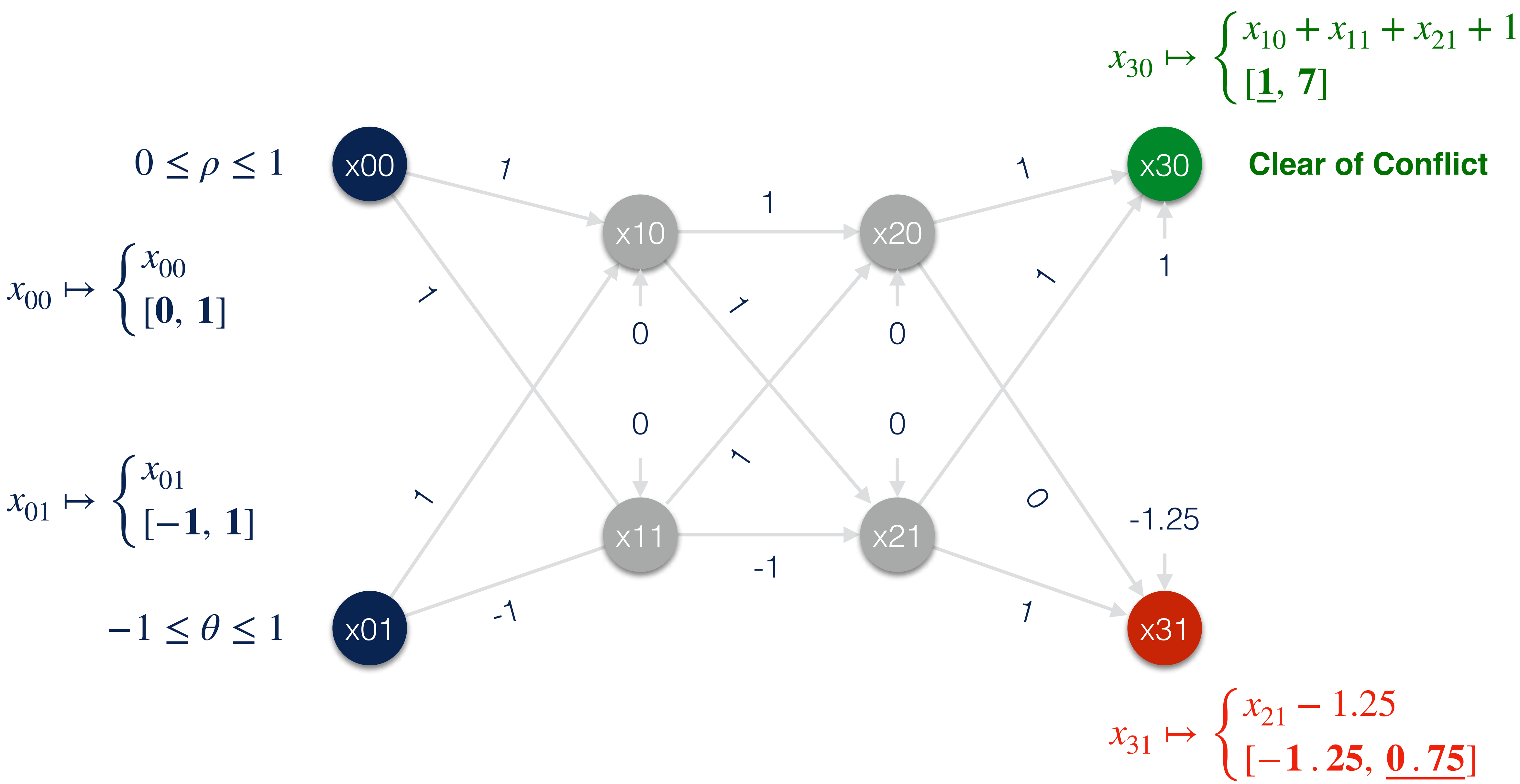
Interval Abstraction

with Symbolic Constant Propagation [Li19]



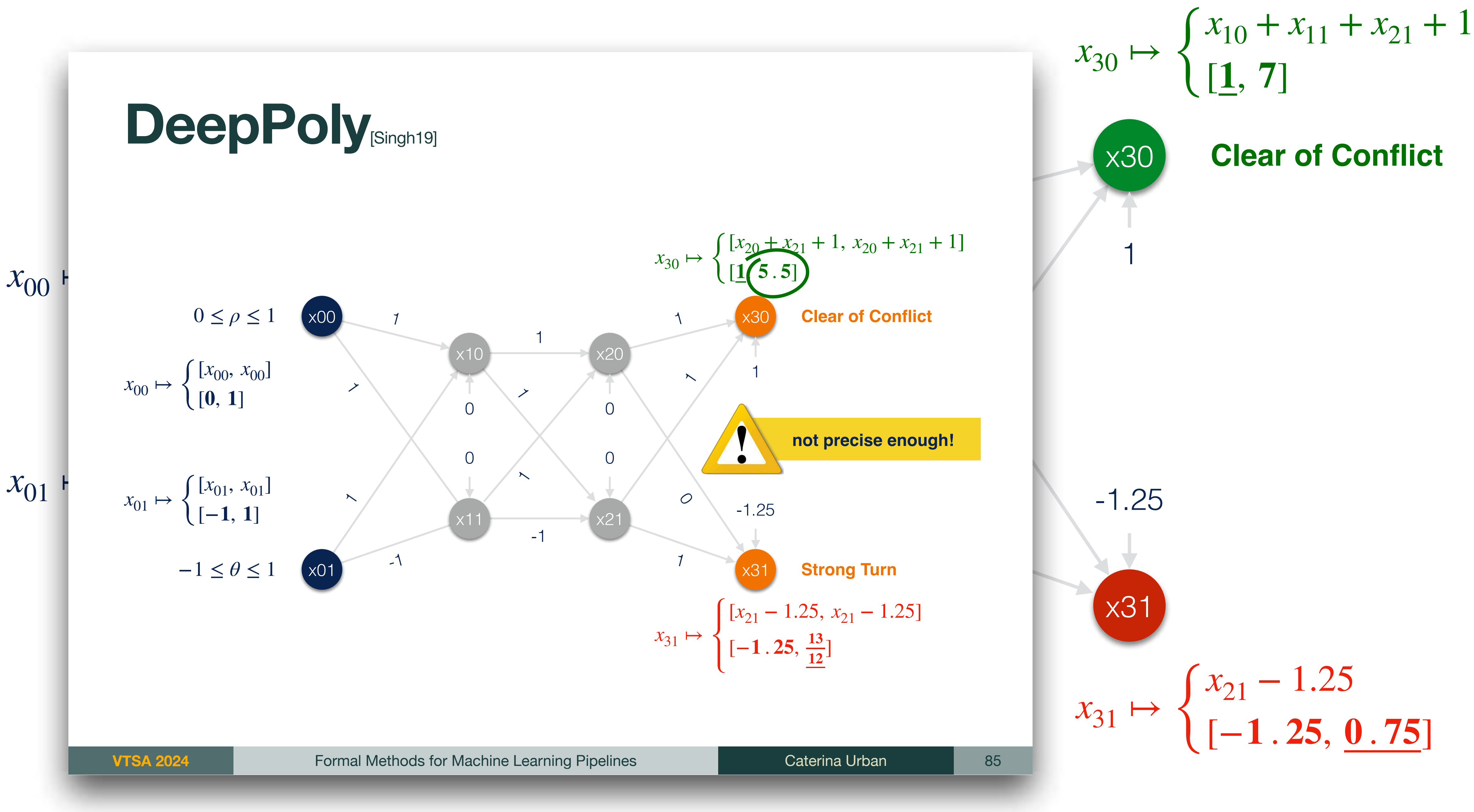
Interval Abstraction

with Symbolic Constant Propagation [Li19]



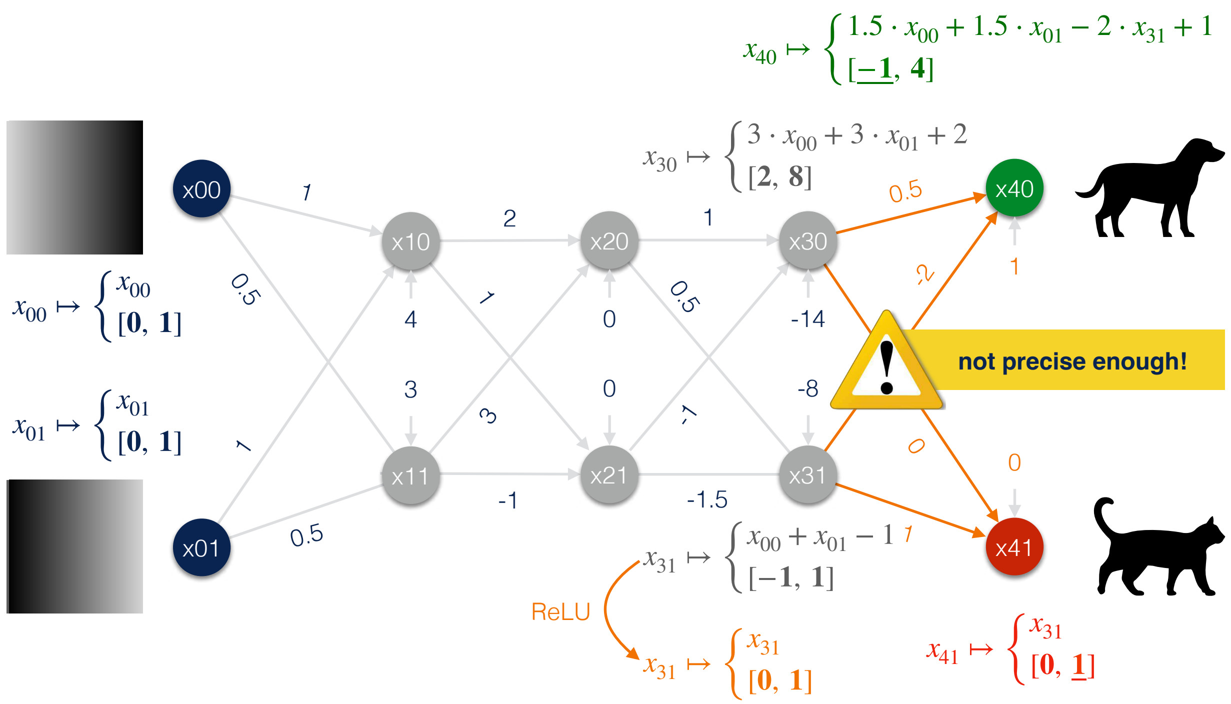
Interval Abstraction

with Symbolic Constant Propagation [Li19]

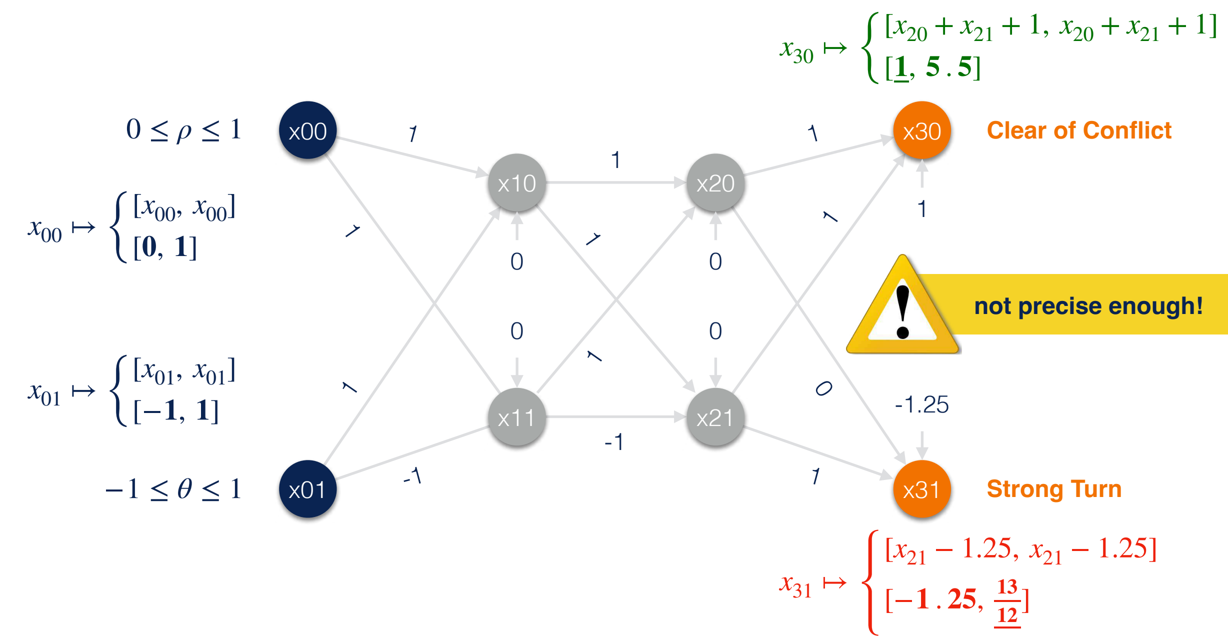


Interval Abstraction

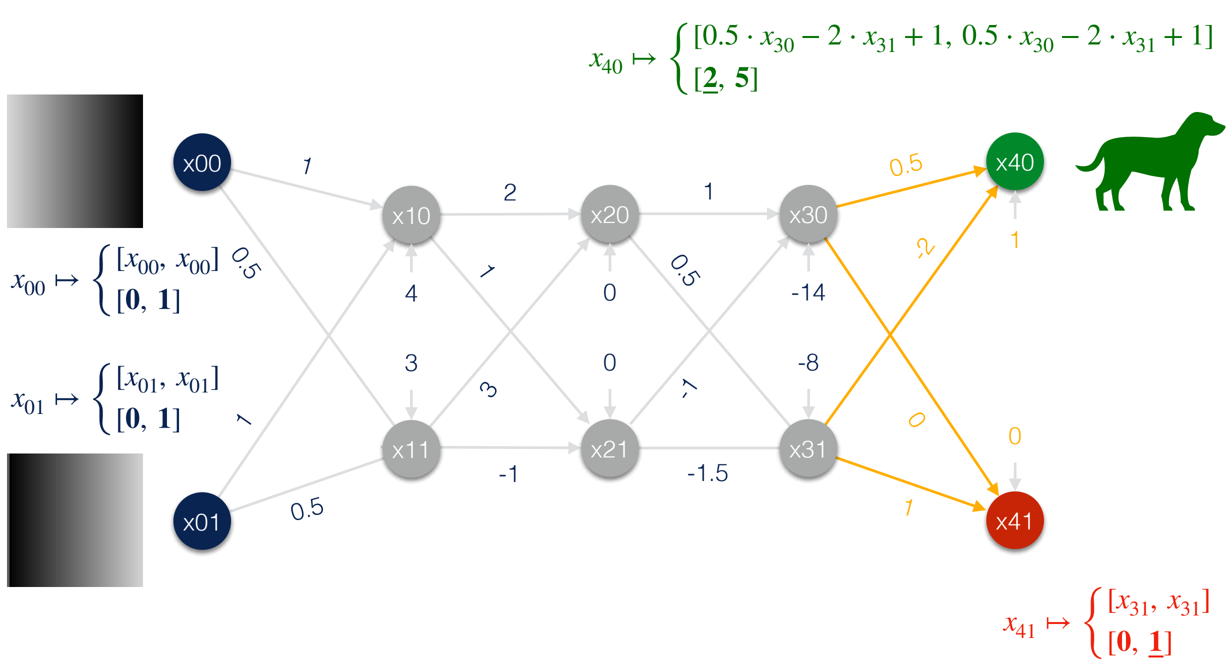
with Symbolic Constant Propagation [Li19]



DeepPoly [Singh19]

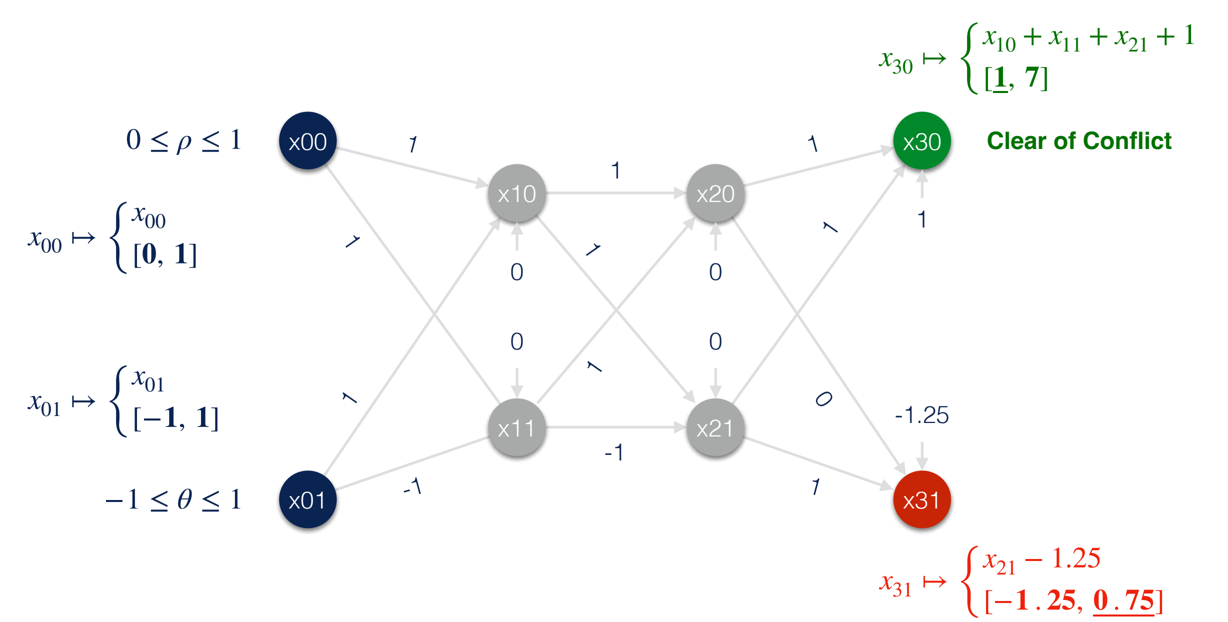


DeepPoly [Singh19]



Interval Abstraction

with Symbolic Constant Propagation [Li19]



Product Domain [Mazzucato21]

DeepPoly with Symbolic Constant Propagation

$$\begin{tikzcd}
\textbf{Symbolic} \arrow[rr, bend right, "{[\mathbf{l}, \mathbf{u}]}"'] & & \textbf{DeepPoly} \arrow[ll, bend right, "{[\mathbf{l}, \mathbf{u}]}"']
\end{tikzcd}$$

$[\mathbf{max}(\mathbf{l}, \mathbf{l}), \mathbf{min}(\mathbf{u}, \mathbf{u})]$

$[\mathbf{max}(\mathbf{l}, \mathbf{l}), \mathbf{min}(\mathbf{u}, \mathbf{u})]$

Product Domain [Mazzucato21]

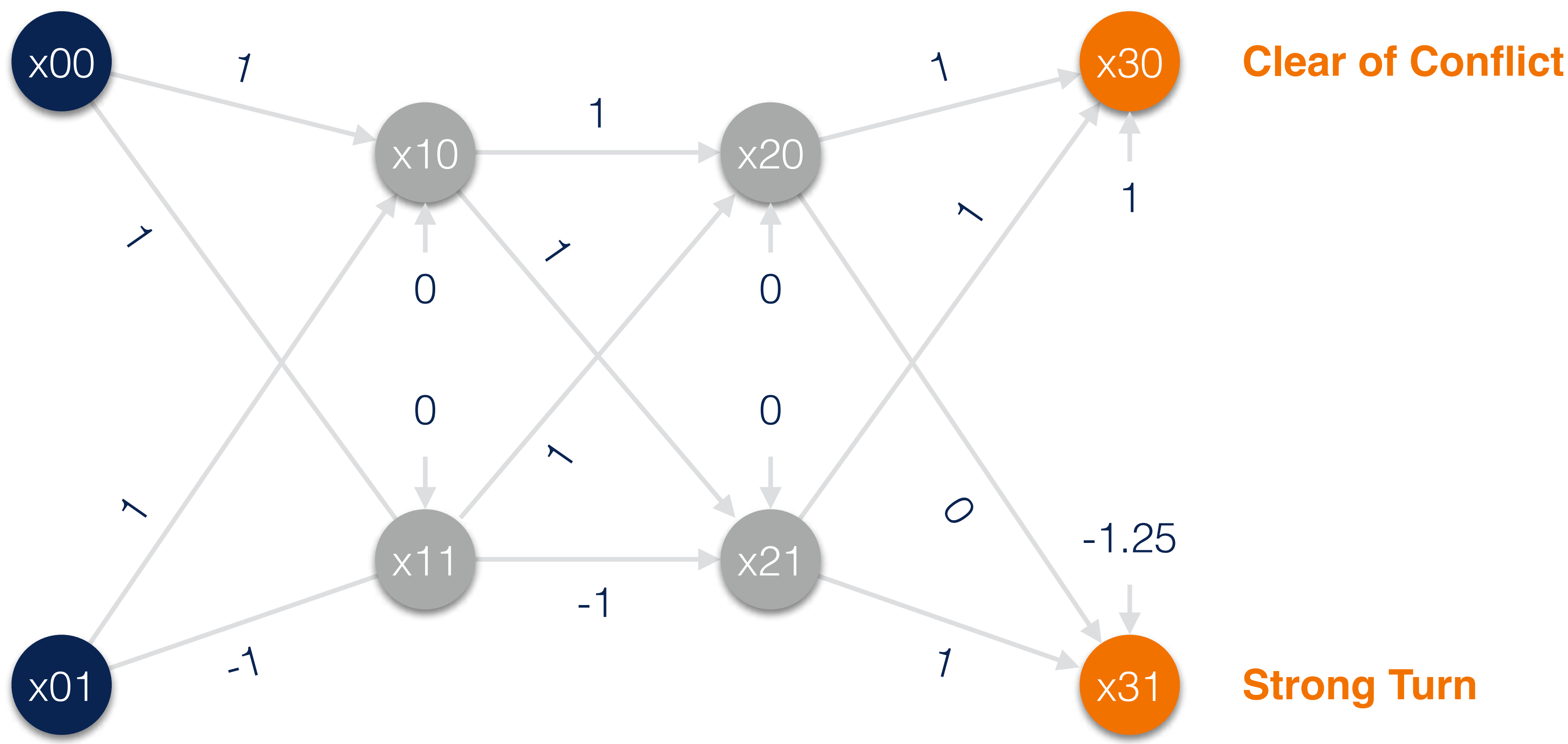
DeepPoly with Symbolic Constant Propagation

$0 \leq \rho \leq 1$

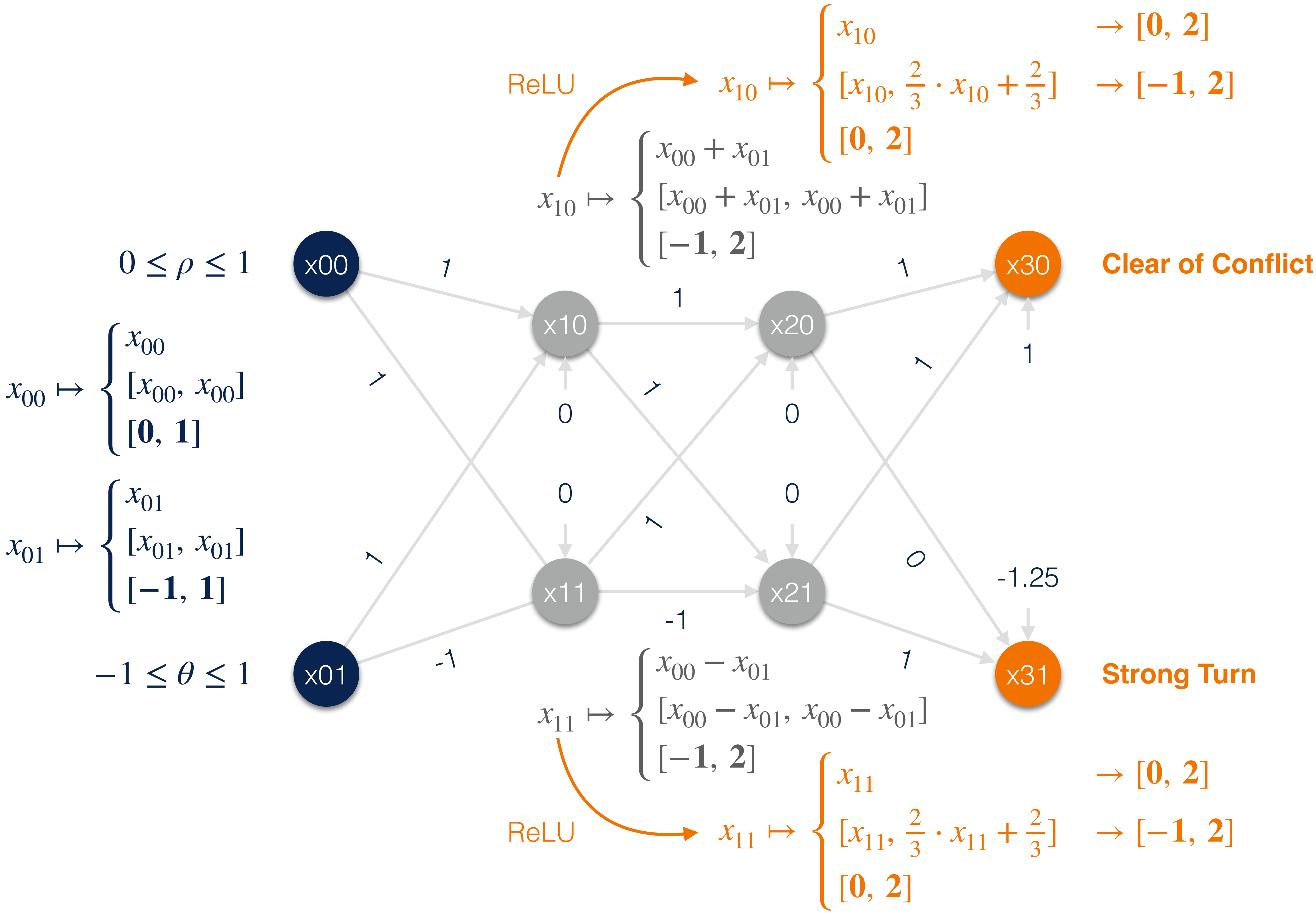
$$x_{00} \mapsto \begin{cases} x_{00} \\ [x_{00}, x_{00}] \\ [\mathbf{0}, \mathbf{1}] \end{cases}$$

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

$-1 \leq \theta \leq 1$



Product Domain [Mazzucato21]



Product Domain [Mazzucato21]

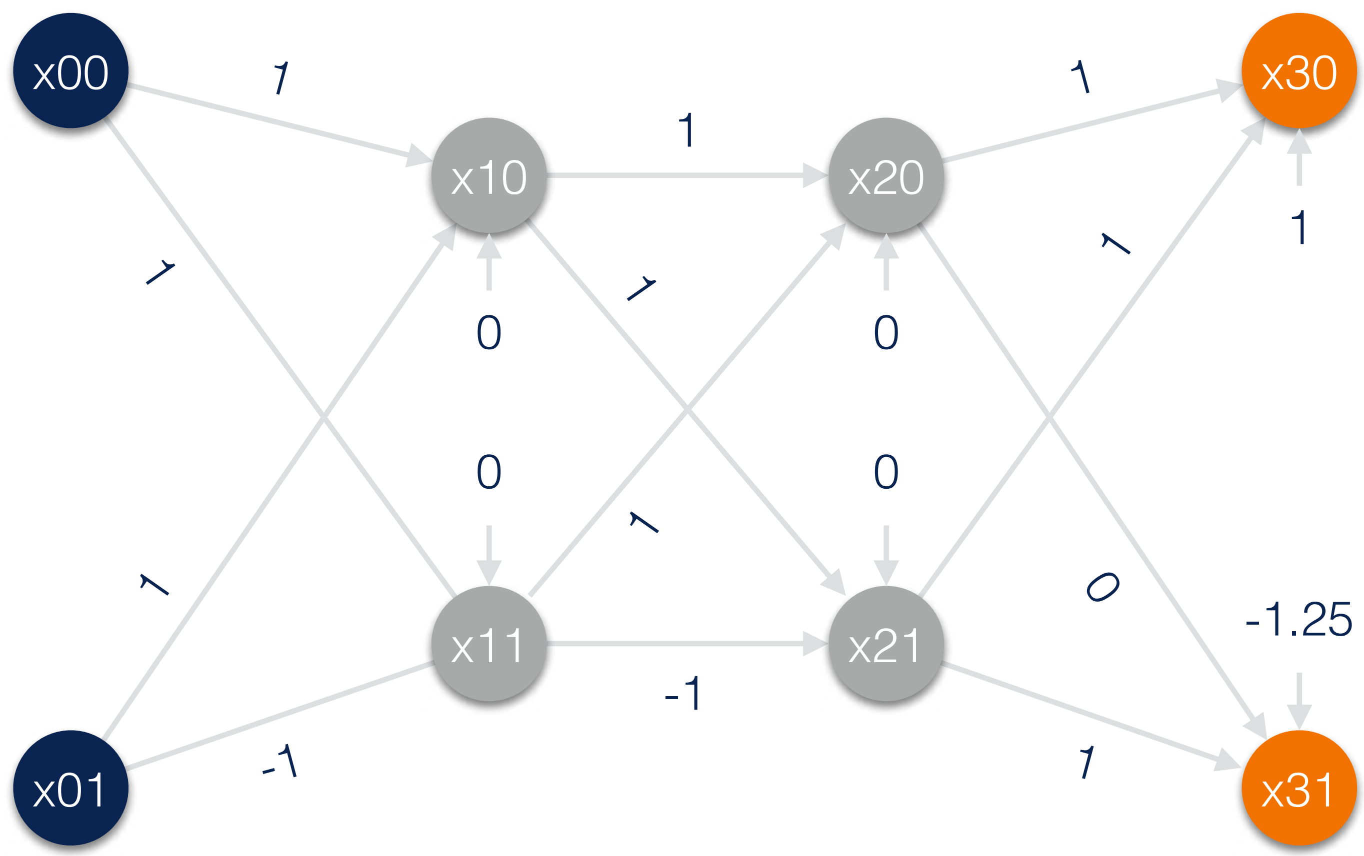
$$x_{20} \mapsto \begin{cases} x_{10} + x_{11} & \rightarrow [\mathbf{0},\ \mathbf{4}] \\ [x_{10} + x_{11},\ x_{10} + x_{11}] & \rightarrow [\mathbf{0},\ \tfrac{\mathbf{8}}{\mathbf{3}}] \\ [\mathbf{0},\ \tfrac{\mathbf{8}}{\mathbf{3}}] & \end{cases}$$

$0 \leq \rho \leq 1$

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

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

$-1 \leq \theta \leq 1$



Clear of Conflict

Strong Turn

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

ReLU

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

Product Domain [Mazzucato21]

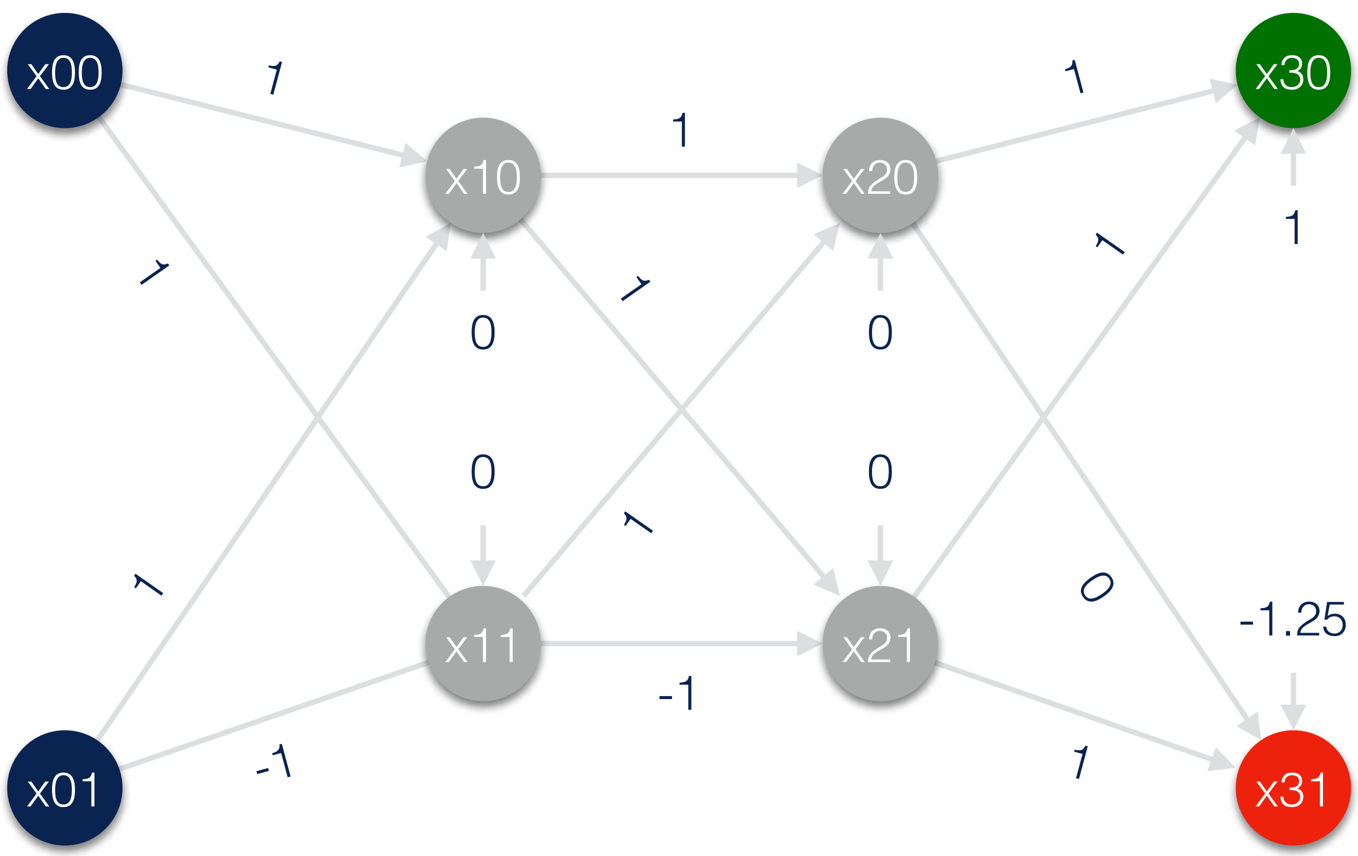
$$x_{30} \mapsto \begin{cases} x_{10} + x_{11} + x_{21} + 1 & \to \mathbf{[1,\ \tfrac{20}{3}]} \\ [x_{20} + x_{21} + 1,\ x_{20} + x_{21} + 1] & \to \mathbf{[1,\ 4.5]} \\ \mathbf{[\underline{1},\ 4.5]} \end{cases}$$

$0 \leq \rho \leq 1$

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

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

$-1 \leq \theta \leq 1$



Clear of Conflict

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

Other Complete Methods

Star Sets

Exact Static Analysis Method

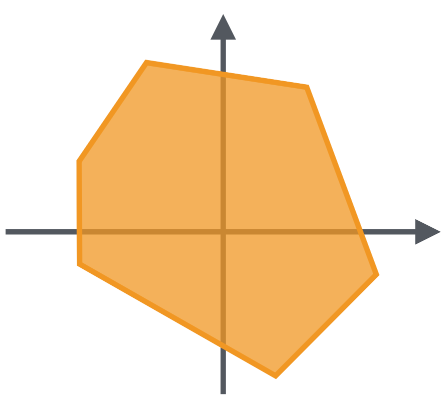
use union of **efficient representations** of bounded convex polyhedra

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

$c \in \mathcal{R}^n$: center

$V = \{v_1, \ldots, v_m\}$: basis vectors in $\mathcal{R}^n$

$P : \mathcal{R}^m \to \{ \perp , \top \}$: 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)

Star Sets

Exact Static Analysis Method

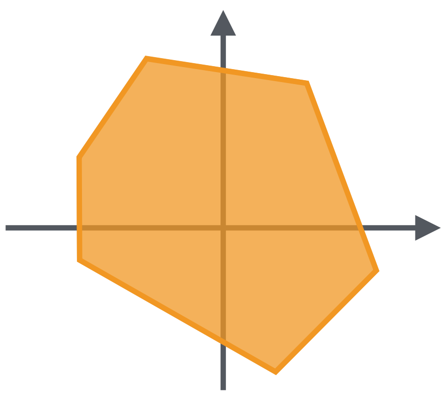
us
efficient
of bounded

Follow-up Work

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

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

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



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

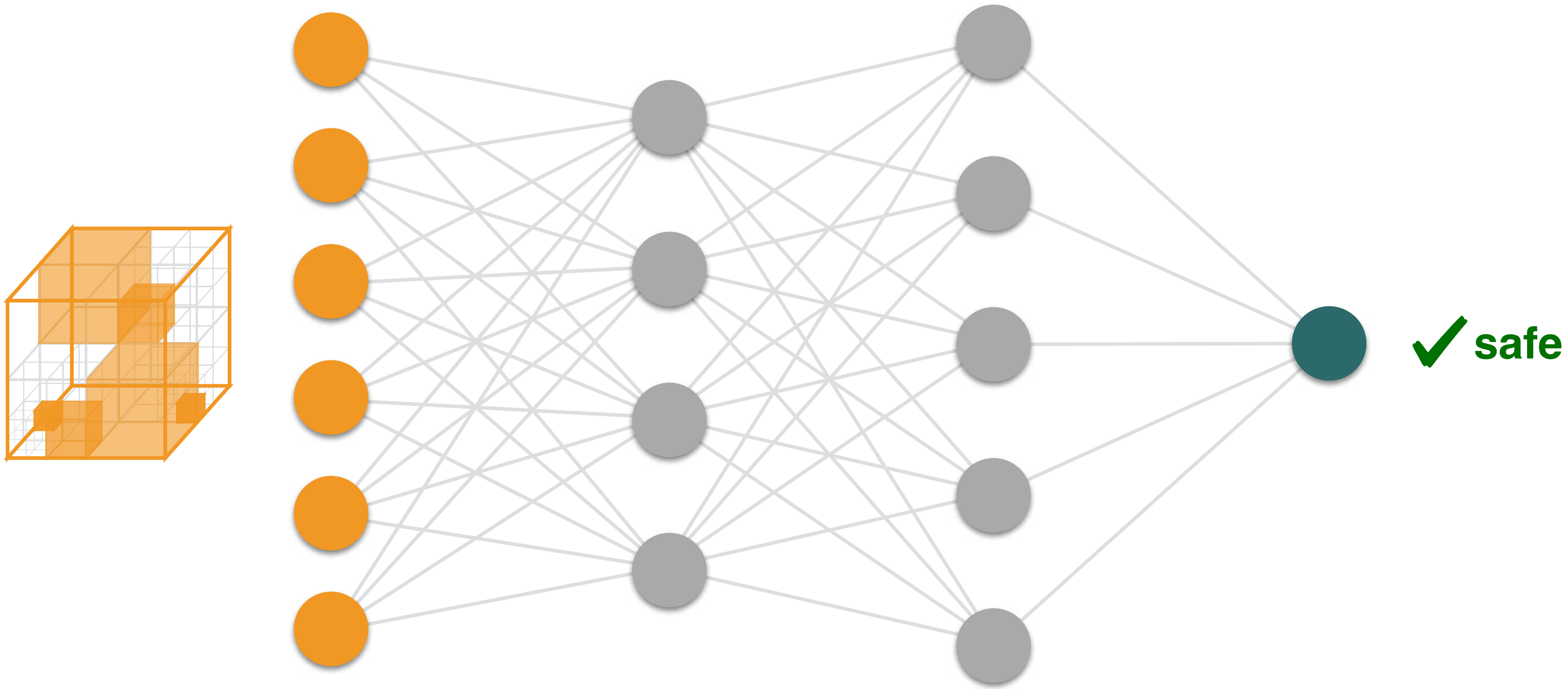
- fast and cheap **affine mapping operations** → neural network layers
- inexpensive **intersections with half-spaces** → ReLU activations

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

ReluVal

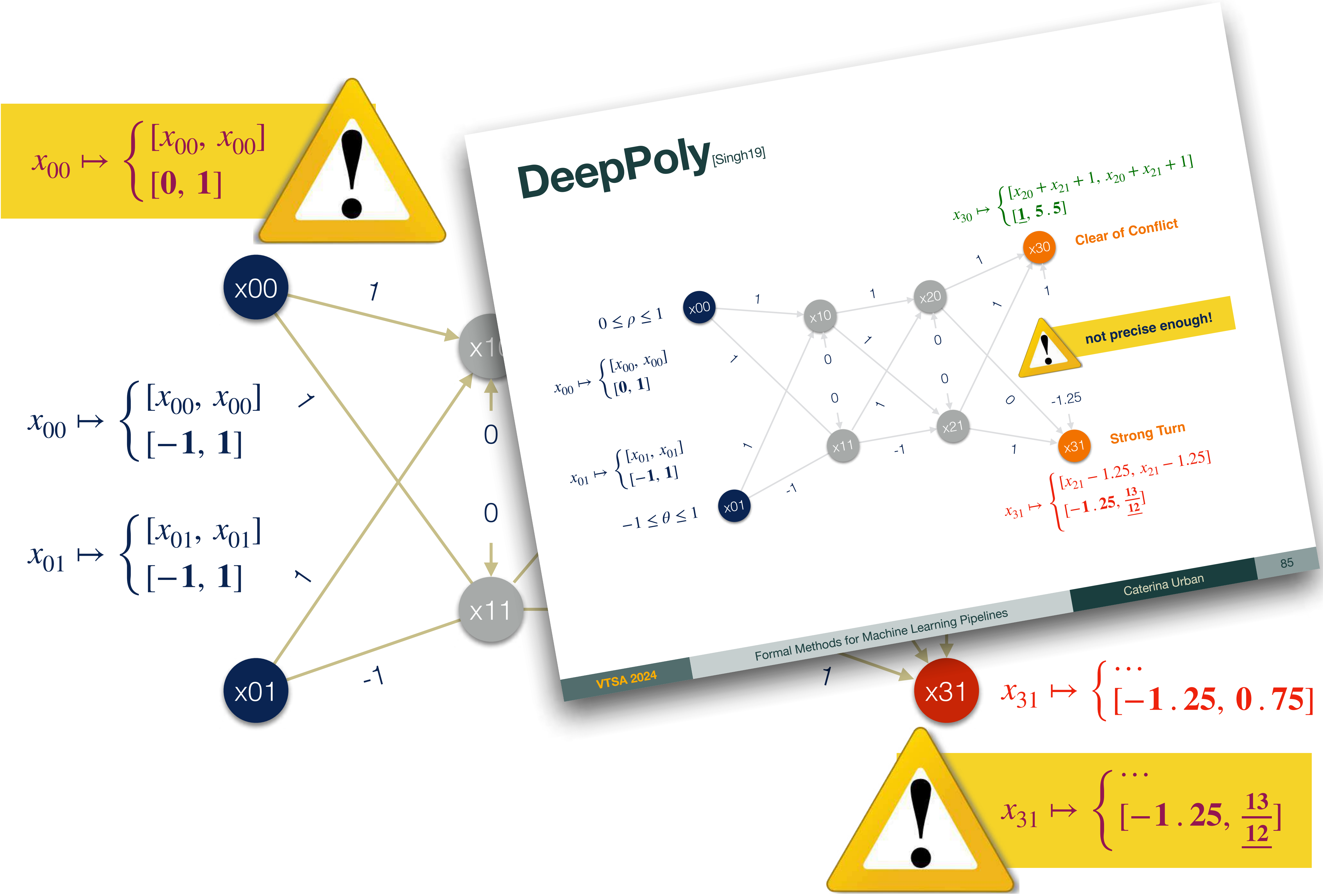
Asymptotically Complete Method

use symbolic propagation
+ iterative input refinement



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

DeepPoly + Input Refinement



Neurify

Asymptotically Complete Method

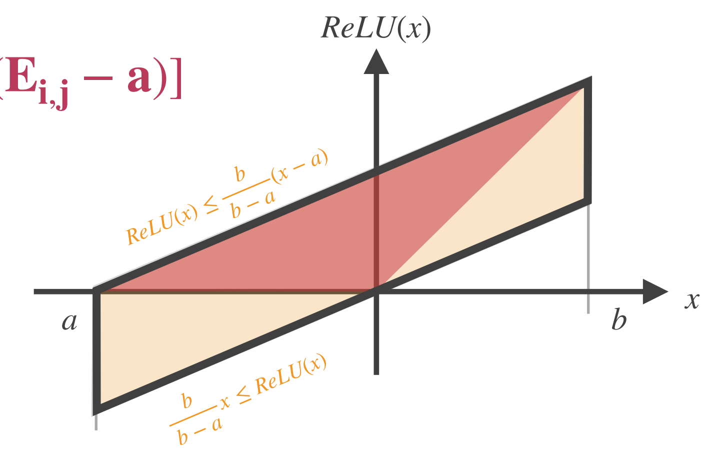
use symbolic propagation + convex ReLU approximation + iterative input/ReLU refinement

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

$$x_{i,j} \mapsto \begin{cases} [\mathbf{E_{i,j}}, \mathbf{E_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases} \xrightarrow{\text{ReLU}} x_{i,j} \mapsto \begin{cases} [\mathbf{E_{i,j}}, \mathbf{E_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases} \qquad 0 \leq a$$

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

$$x_{i,j} \mapsto \begin{cases} [\mathbf{E_{i,j}}, \mathbf{E_{i,j}}] \\ [\mathbf{a}, \mathbf{b}] \end{cases} \xrightarrow{\text{ReLU}} x_{i,j} \mapsto \begin{cases} [\mathbf{0}, \mathbf{0}] \\ [\mathbf{0}, \mathbf{0}] \end{cases} \qquad b \leq 0$$



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

Further Complete Methods

- **W. Ruan, X. Huang, and M. Kwiatkowska**. *Reachability Analysis of Deep Neural Networks with Provable Guarantees*. In IJCAI, 2018.
  a global optimization-based approach for verifying Lipschitz continuous neural networks

- **G. Singh, T. Gehr, M. Püschel, and M. Vechev**. *Boosting Robustness Certification of Neural Networks*. In ICLR, 2019.
  an approach combining abstract interpretation and (mixed integer) linear programming

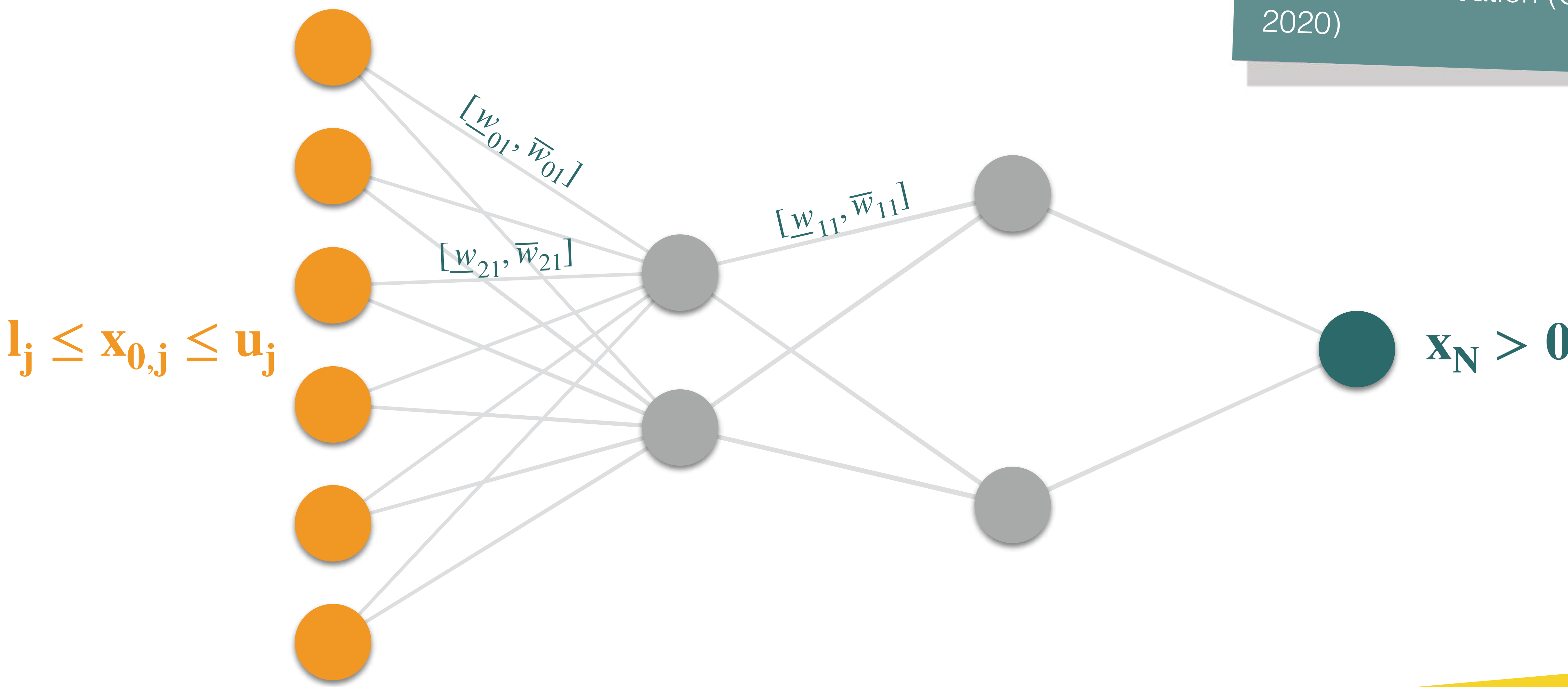
Other Incomplete Methods

Interval Neural Networks

Abstraction-Based Method

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)

Further Incomplete Methods

- **W. Xiang, H.-D. Tran, and T. T. Johnson**. *Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks*. 2018.
  an approach combining simulation and linear programming

- **K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli**. *A Dual Approach to Scalable Verification of Deep Networks*. In UAI, 2018.
  an approach based on duality for verifying neural networks

Further Incomplete Methods

- **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.
  T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon. *Towards Fast Computation of Certified Robustness for ReLU Networks*. In ICML, 2018.
  H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. Efficient *Neural Network Robustness Certification with General Activation Functions*. In NeurIPS, 2018.
  approaches for finding a lower bound on robustness to adversarial perturbations

Further 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

- **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**

suffer from **false positives**

Disadvantages

able to scale to large models

sound often also with respect to
floating-point arithmetic

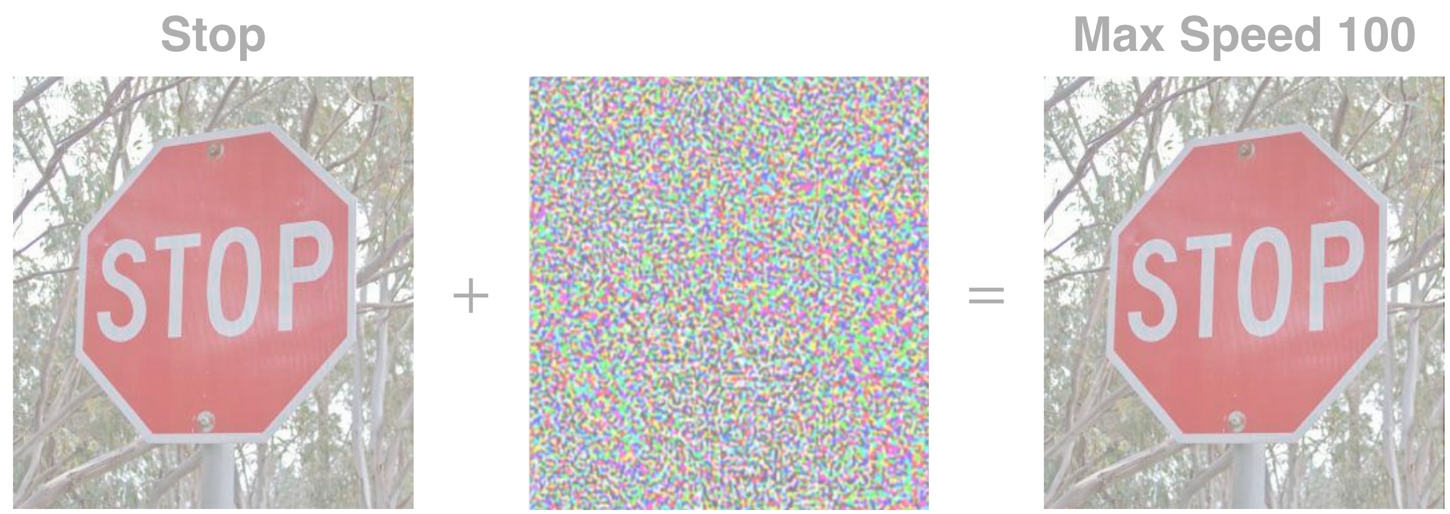
less limited to certain
model **architectures**

Advantages

Incomplete Methods

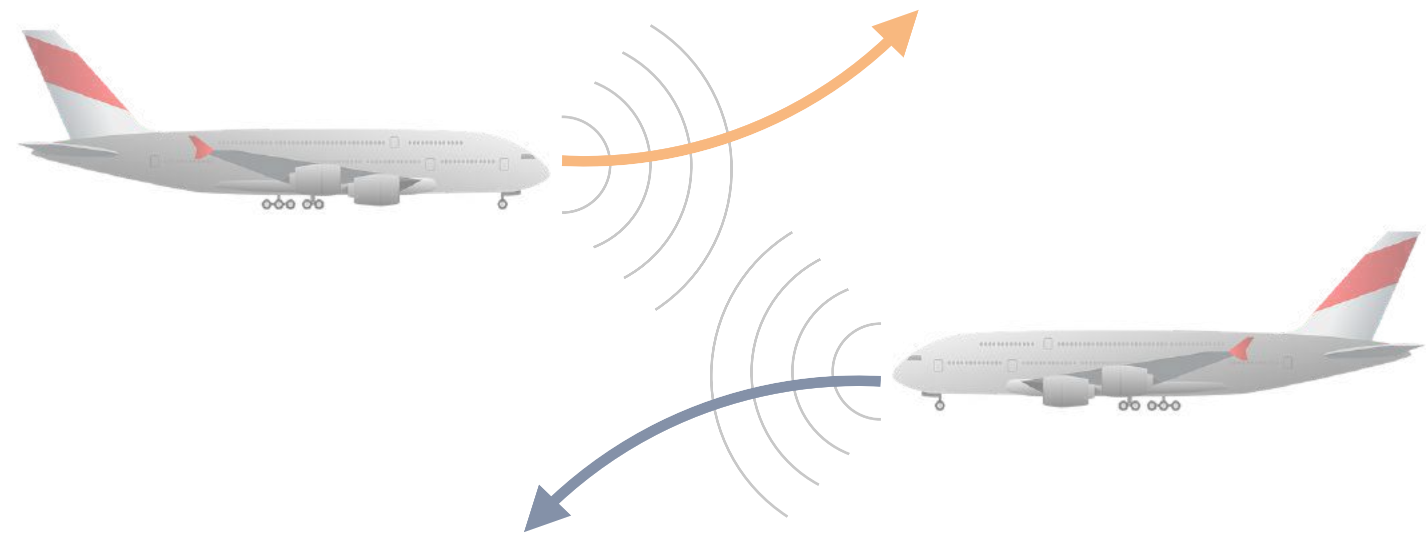
Stability

Goal G3 in [Kurd03]

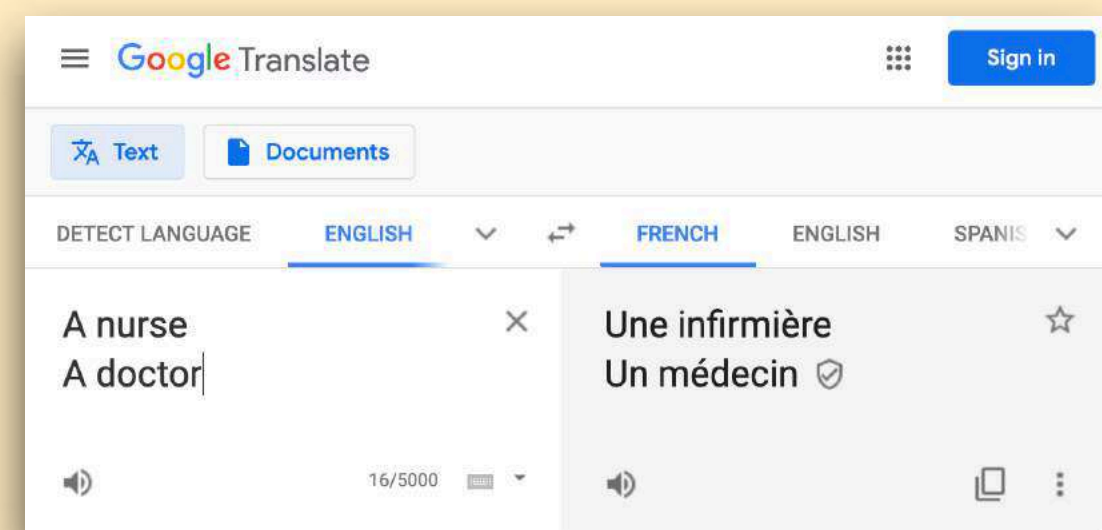


Safety

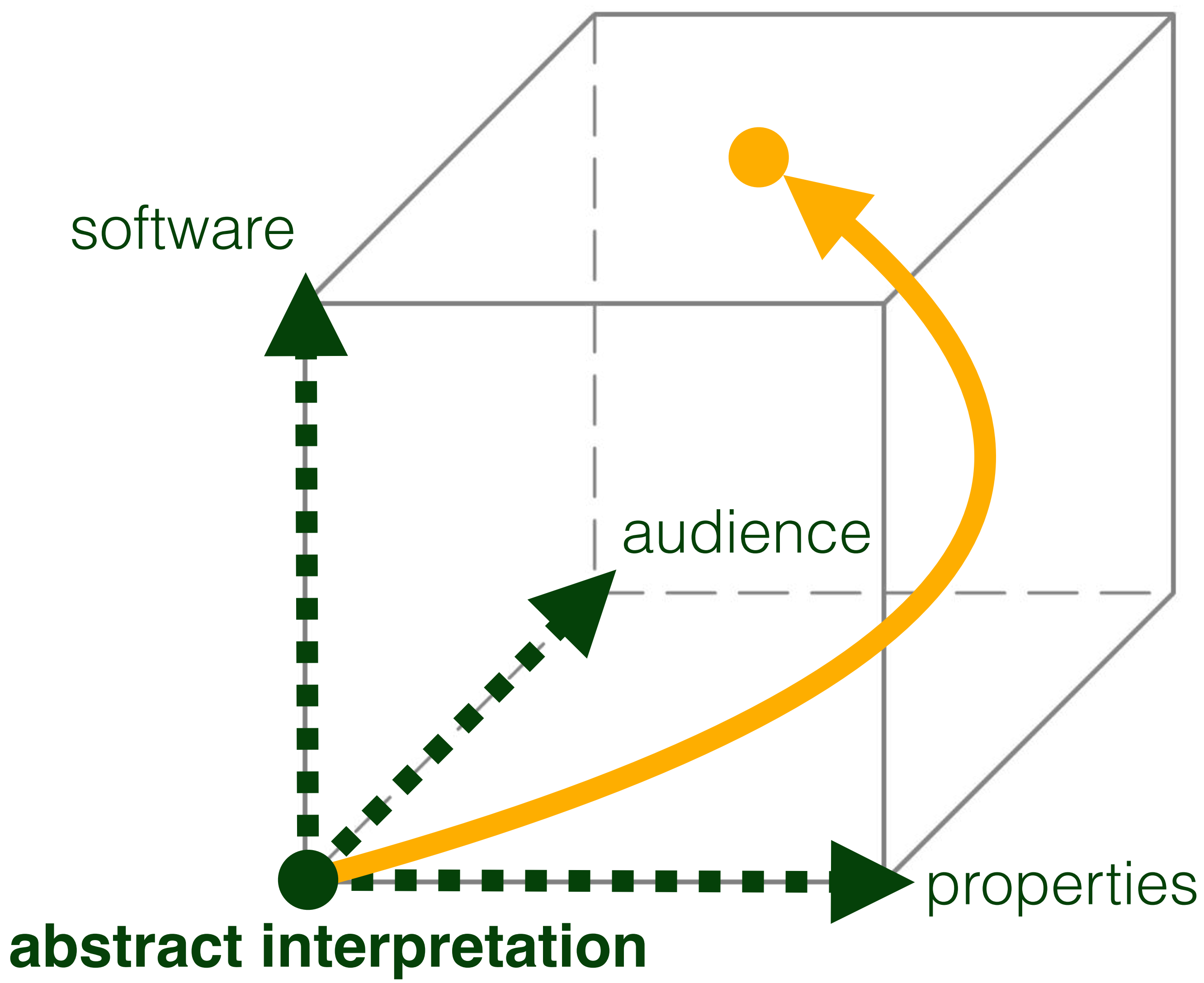
Goal G4 in [Kurd03]



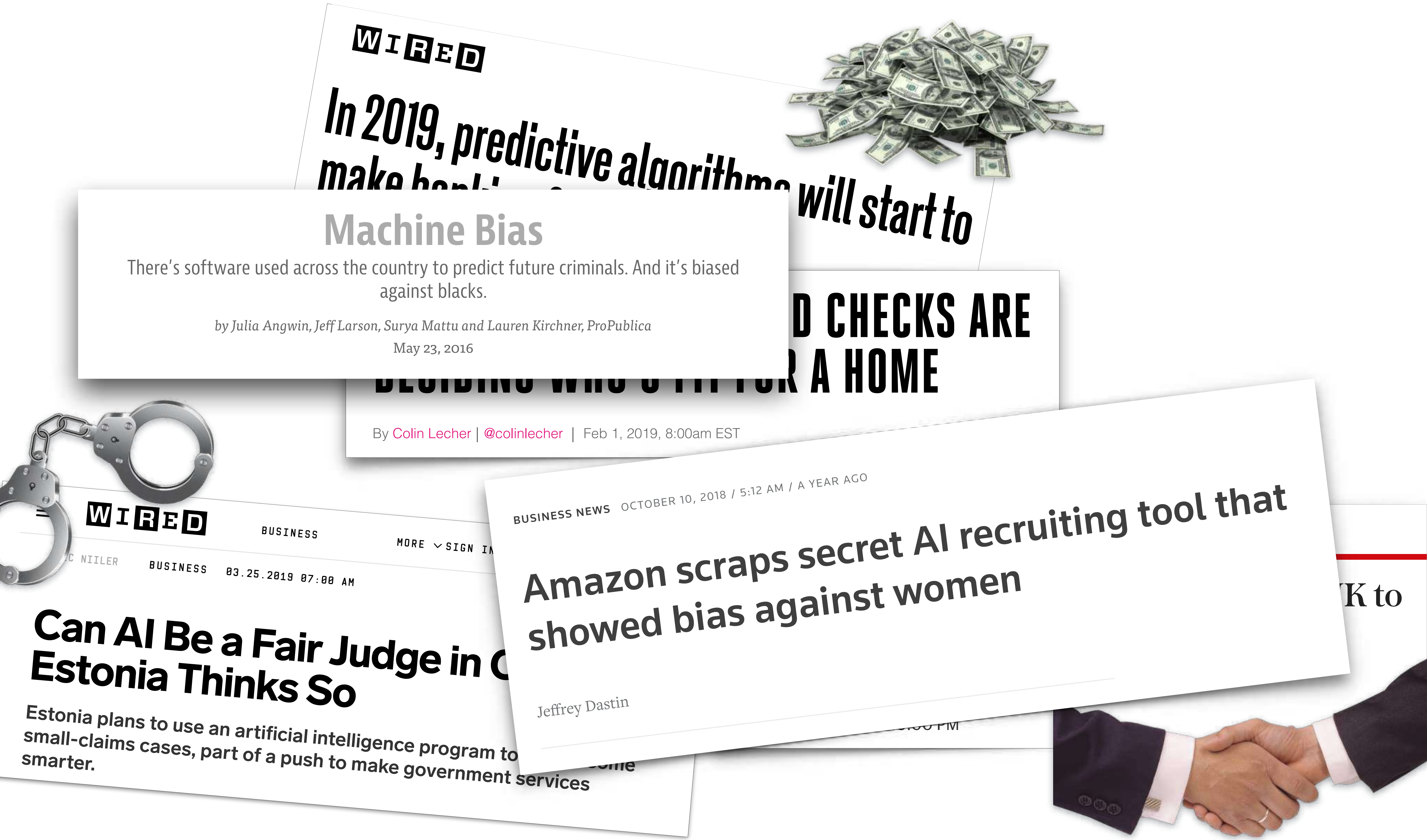
Fairness

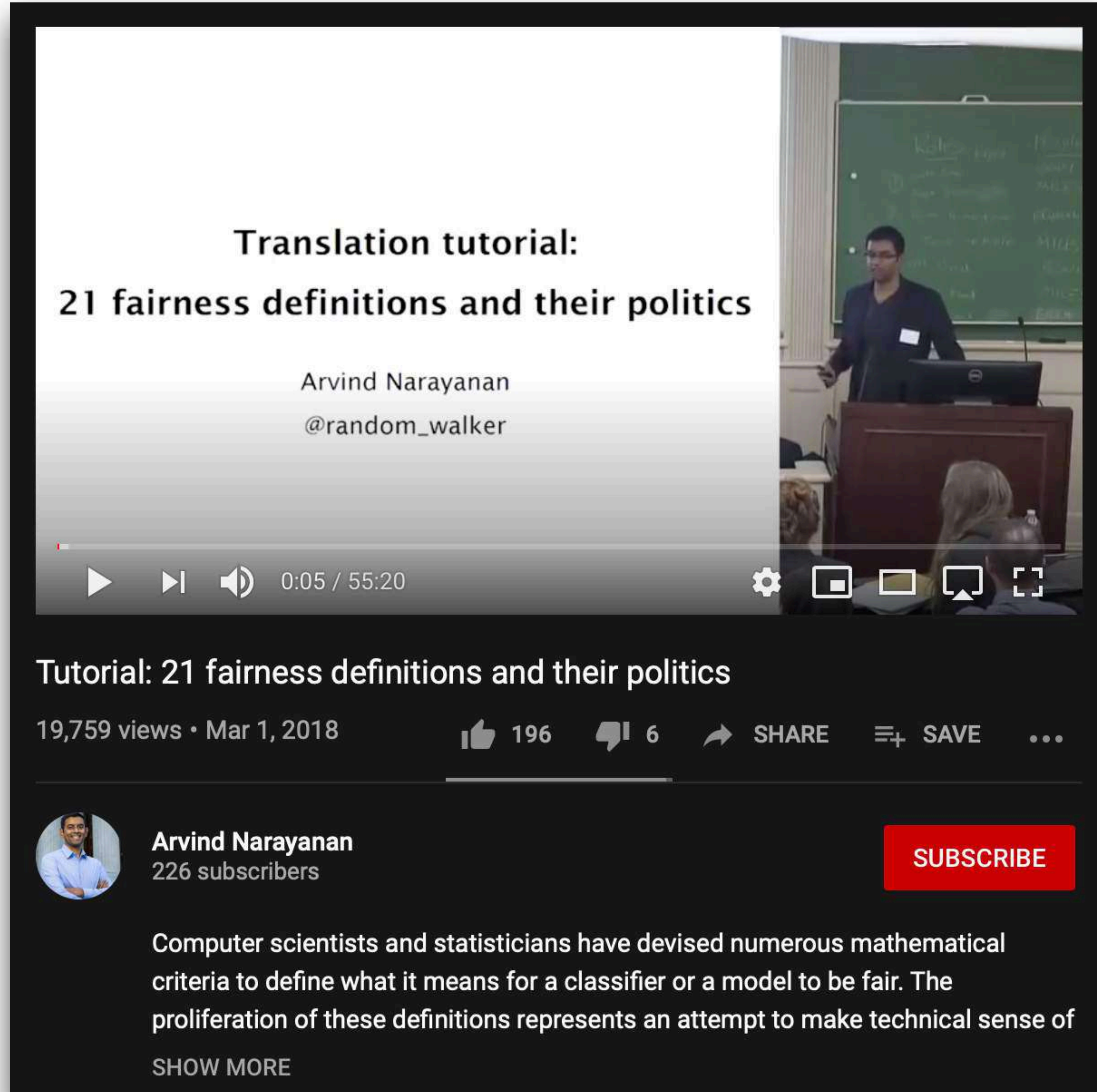


Fairness Verification



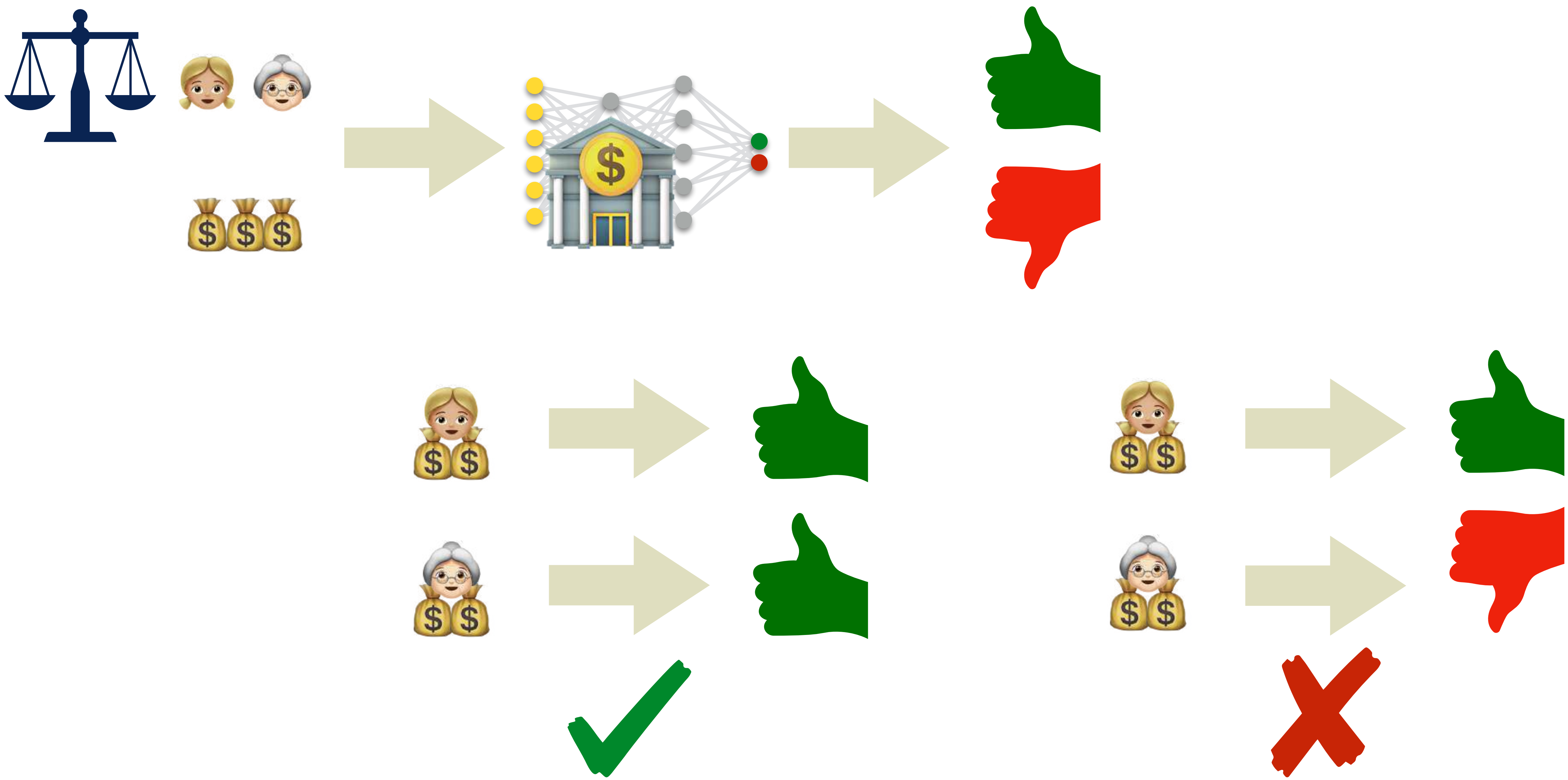
ML Impacts Our Society




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

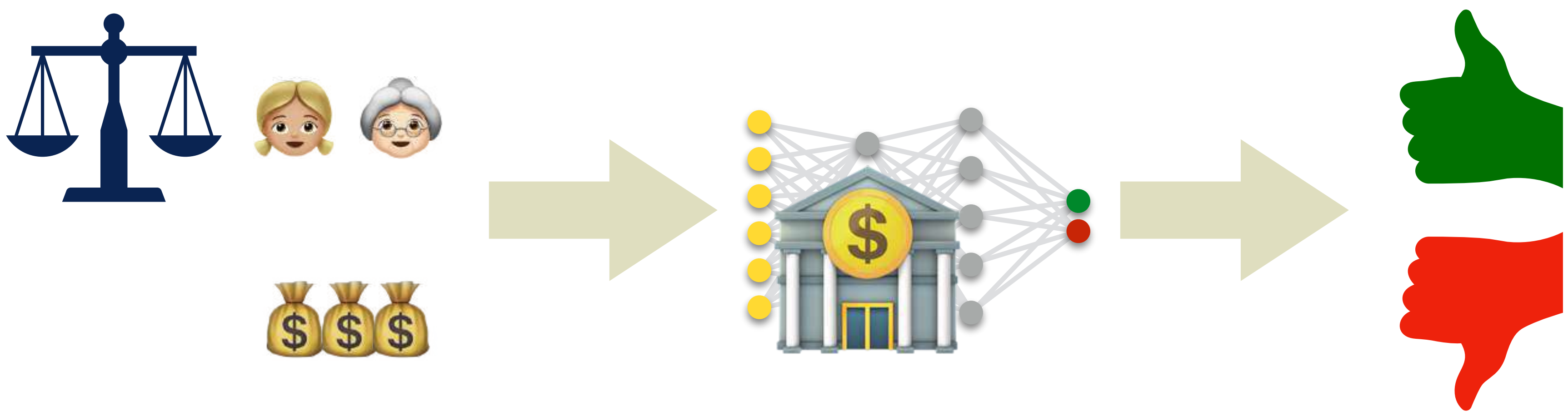
$$\mathscr{F}_i \stackrel{\text{def}}{=} \{ [\![M]\!] \mid \text{UNUSED}_i([\![M]\!]) \}$$

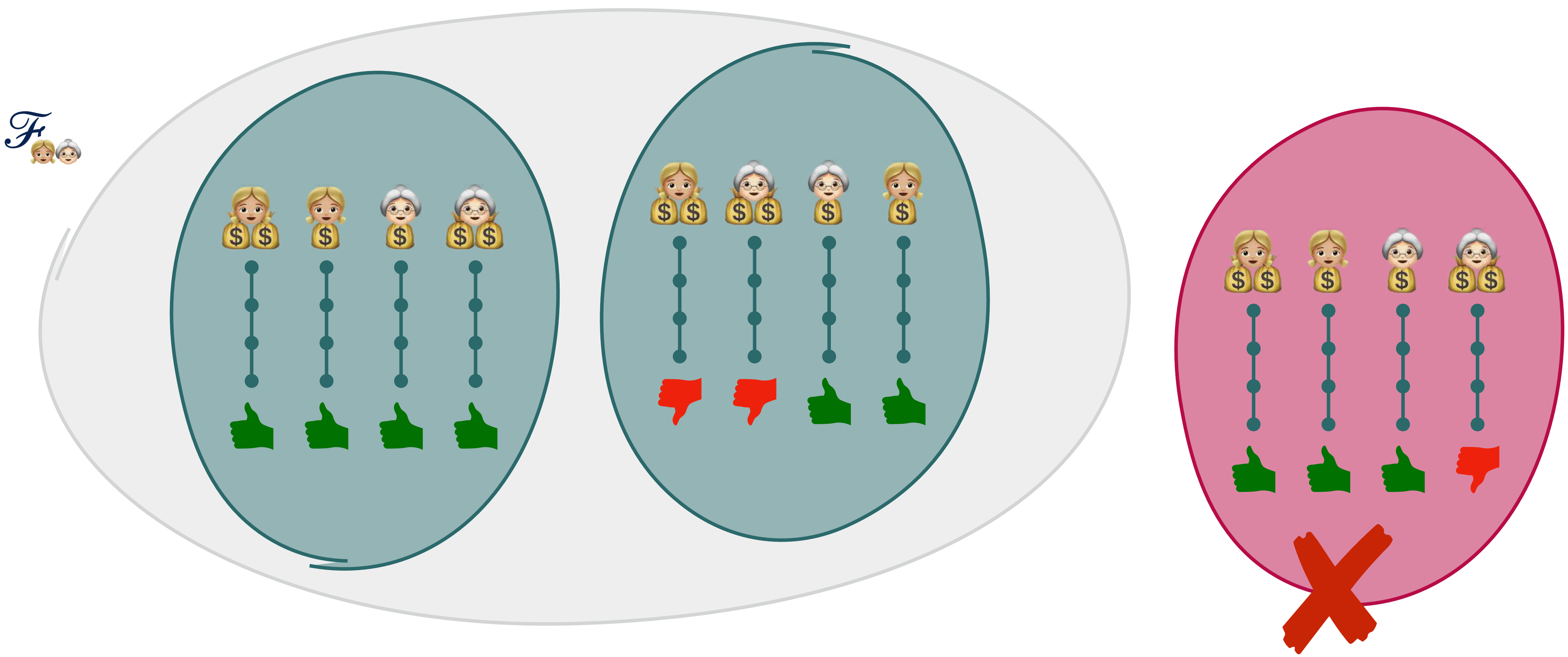
$\mathscr{F}_i$ is the set of all neural networks M (or, rather, their semantics $[\![M]\!]$) that **do not use** the value of the sensitive input node $x_{0,i}$ for classification

$$\begin{aligned}\text{UNUSED}_i(T) \stackrel{\text{def}}{=} \; & \forall t, t' \in T\colon \; t_0(x_{0,i}) \neq t'_0(x_{0,i}) \land \\ & (\forall 0 \leq j \leq |L_0|\colon j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j})) \\ & \Rightarrow t_\omega = t'_\omega\end{aligned}$$

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

$$\mathcal{F}_i \overset{\text{def}}{=} \{\llbracket M \rrbracket \mid \text{UNUSED}_i(\llbracket M \rrbracket)\}$$

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

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

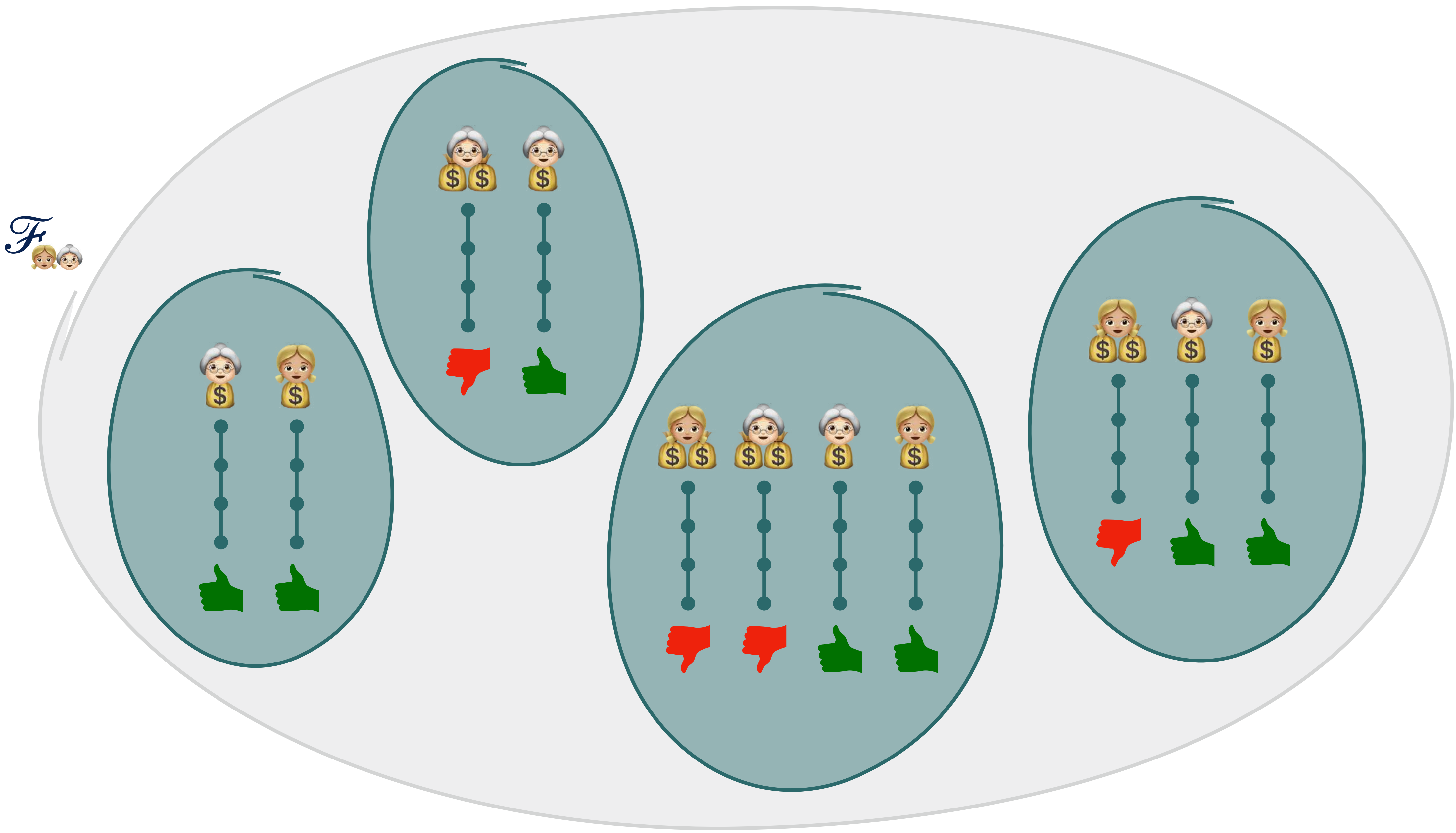
Intuitively: inputs differing only on the value of the sensitive input node $x_{0,i}$ should lead to the same **classification outcome**

Theorem

$$M \models \mathcal{F}_i \Leftrightarrow \{\llbracket M \rrbracket\} \subseteq \mathcal{F}_i$$

Dependency Fairness

Subset-Closed Property (*)



(*) ML Models are Deterministic

Dependency Fairness

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

$\mathcal{F}_i$ 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

$$\begin{aligned}\text{UNUSED}_i(T) \stackrel{\text{def}}{=} \; & \forall t, t' \in T \colon \; t_0(x_{0,i}) \neq t'_0(x_{0,i}) \wedge \\ & (\forall 0 \leq j \leq |L_0| \colon j \neq i \Rightarrow t_0(x_{0,j}) = t'_0(x_{0,j})) \\ & \Rightarrow t_\omega = t'_\omega\end{aligned}$$

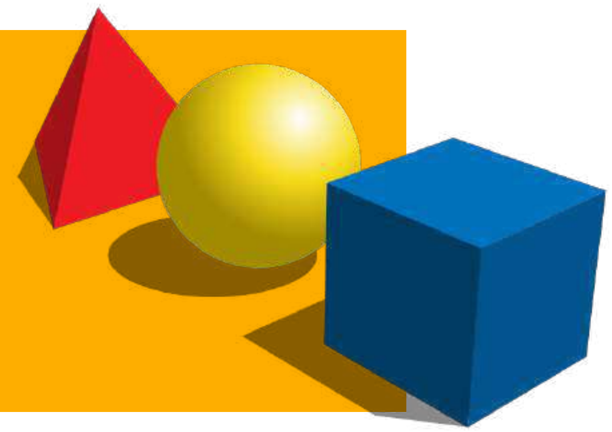
Intuitively: inputs differing only on the value of the sensitive input node $x_{0,i}$ should lead to the same **classification outcome**

Theorem	Corollary
$M \models \mathcal{F}_i \Leftrightarrow \{ [\![M]\!] \} \subseteq \mathcal{F}_i$	$M \models \mathcal{F}_i \Leftarrow [\![M]\!] \subseteq [\![M]\!]^\natural \in \mathcal{F}_i$

Abstract Interpretation Recipe

practical tools
targeting specific programs

algorithmic approaches
to decide program properties



mathematical models
of the program behavior

Abstract Interpretation Recipe

practical tools
targeting specific programs

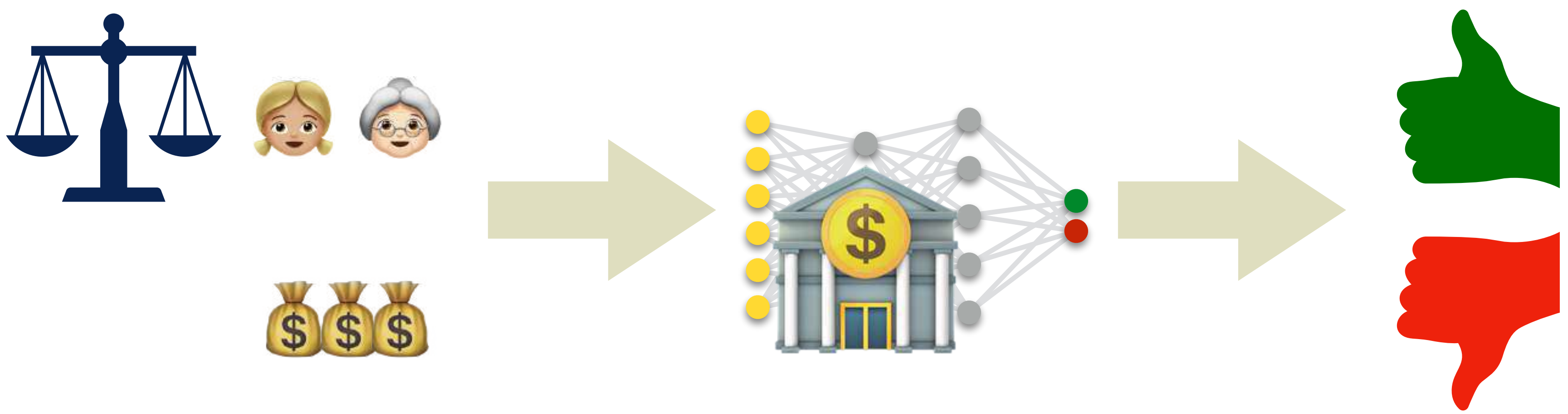
algorithmic approaches
to decide program properties

mathematical models
of the program behavior

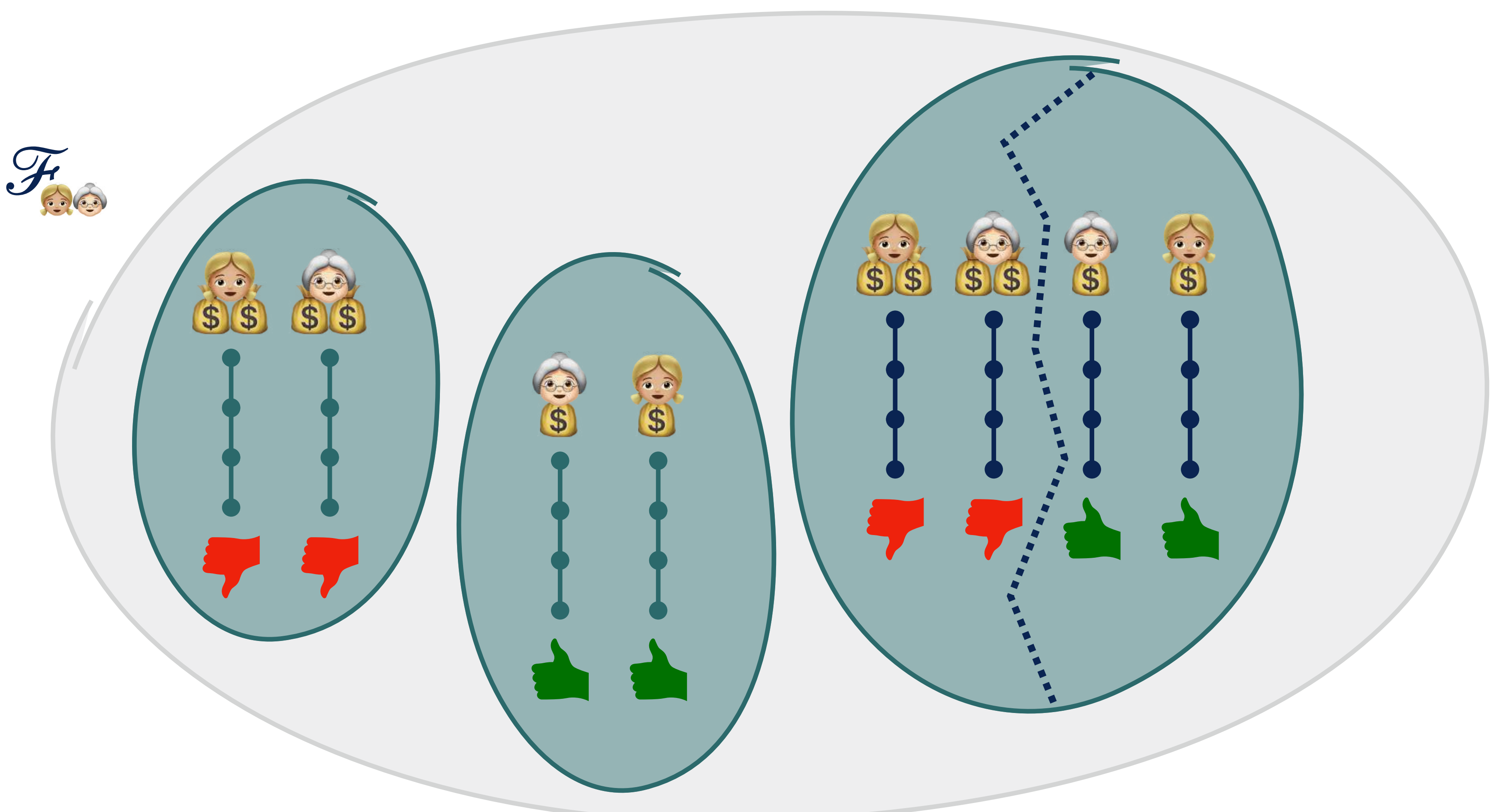
Hierarchy of Semantics

$$
\begin{tikzcd}
\text{parallel semantics} & \{[M]\}^{\parallel}_{\leadsto} & & \\
& & {[\![M]\!]_{\leadsto}} \arrow[ul, "\alpha_{\parallel}"'] & \text{dependency semantics} \\
& \{[M]\}^{\parallel}_{\bullet} \arrow[uu, "\alpha_{\leadsto}"] & & \\
& & {[\![M]\!]_{\bullet}} \arrow[ul, "\alpha_{\parallel}"'] \arrow[uu, "\alpha_{\leadsto}"] & \text{outcome semantics} \\
& \{[M]\}^{\parallel} \arrow[uu, "\alpha_{\bullet}"] & & \\
& & \{[\![M]\!]\} \arrow[ul, "\alpha_{\parallel}"'] \arrow[uu, "\alpha_{\bullet}"] & \text{collecting semantics}
\end{tikzcd}
$$

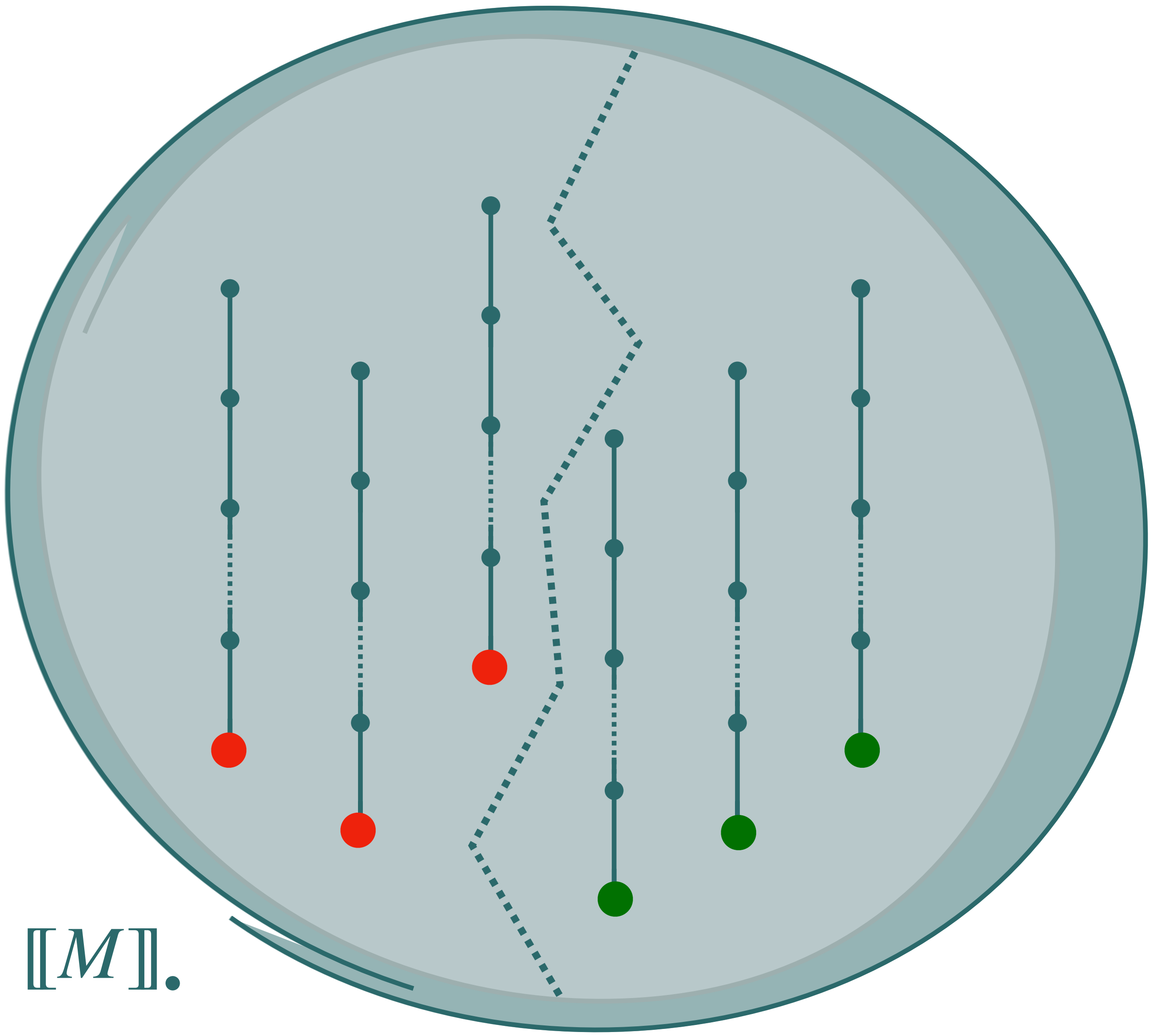
Outcome Semantics



💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the program outcome** yields sets of traces that also satisfy dependency fairness

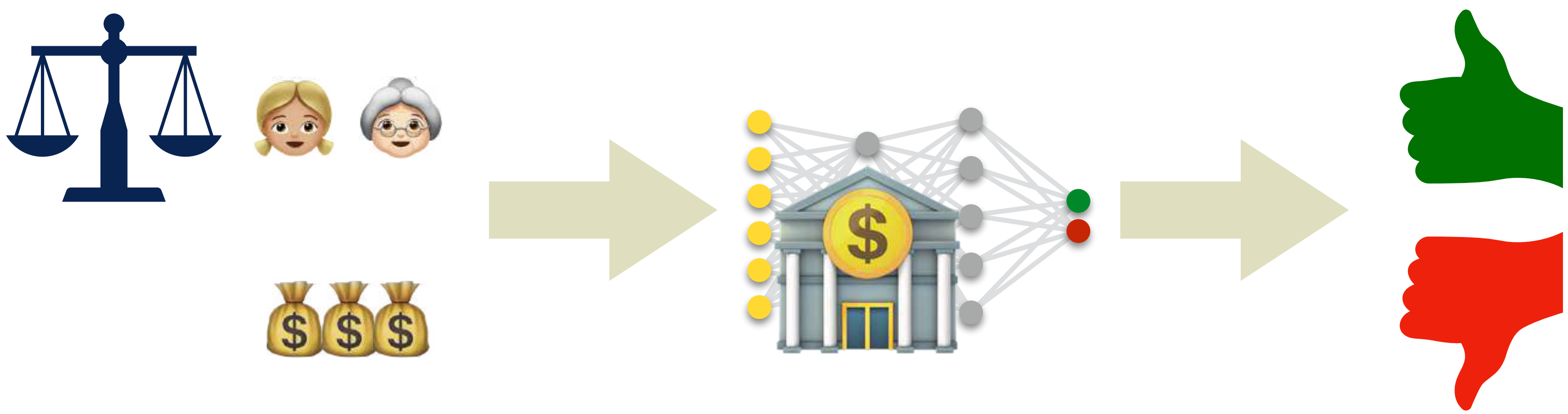


Outcome Semantics

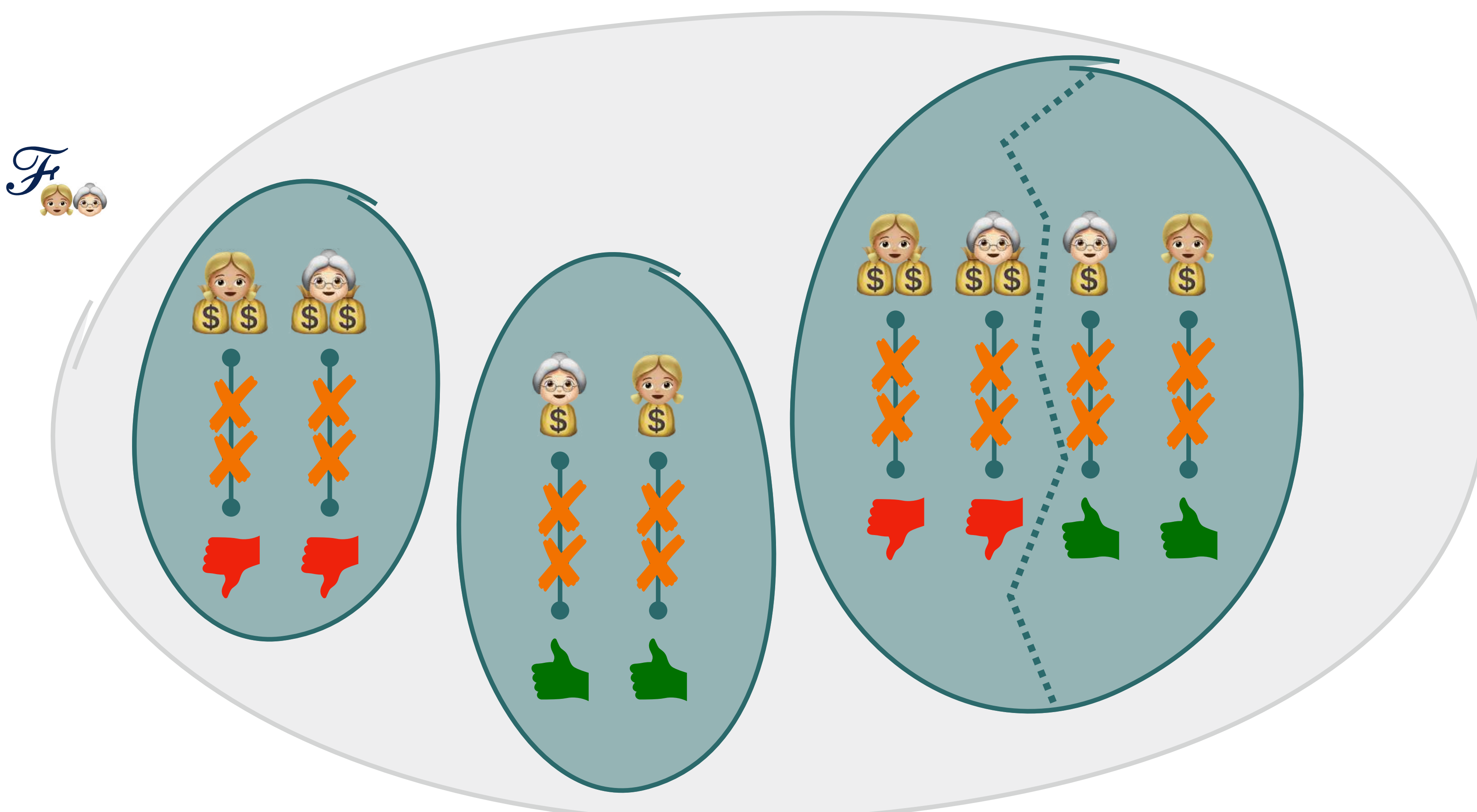


💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the program outcome** yields sets of traces that also satisfy dependency fairness

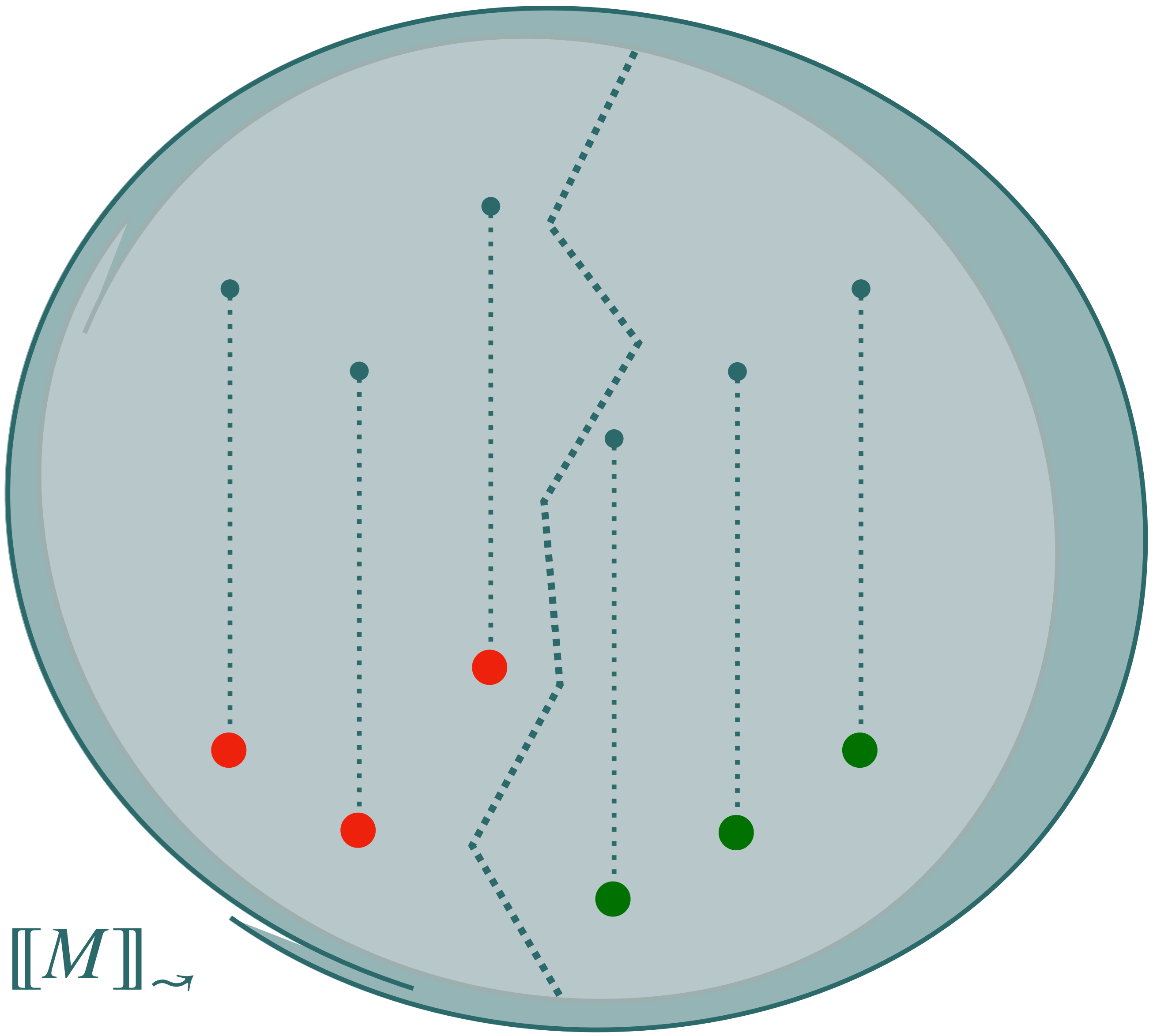
Dependency Semantics



💡 to reason about dependency fairness **we do not need to consider all intermediate computations** between the initial and final states of a trace (if any)



Dependency Semantics

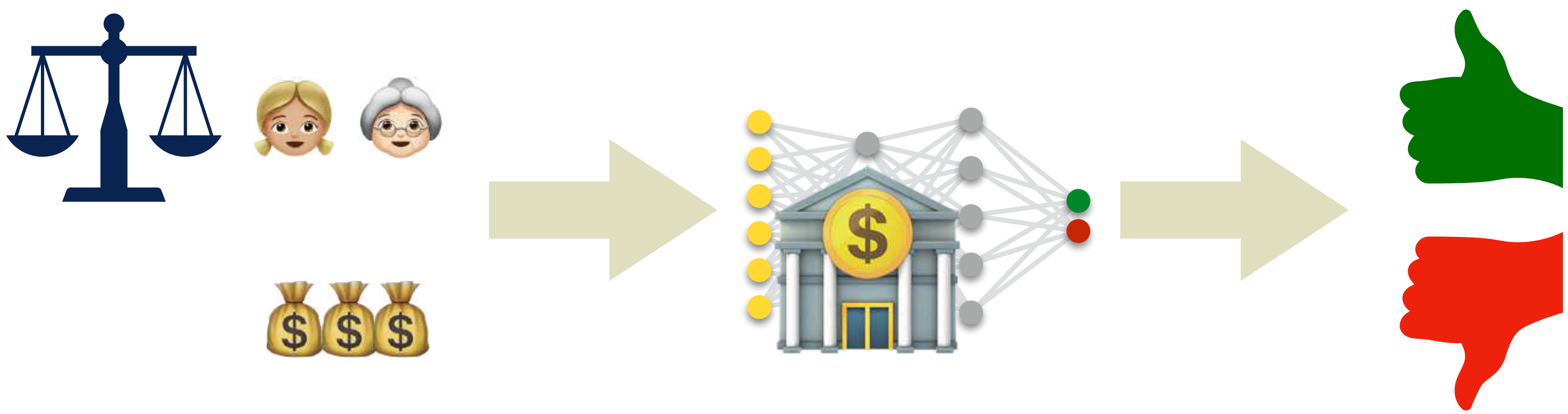


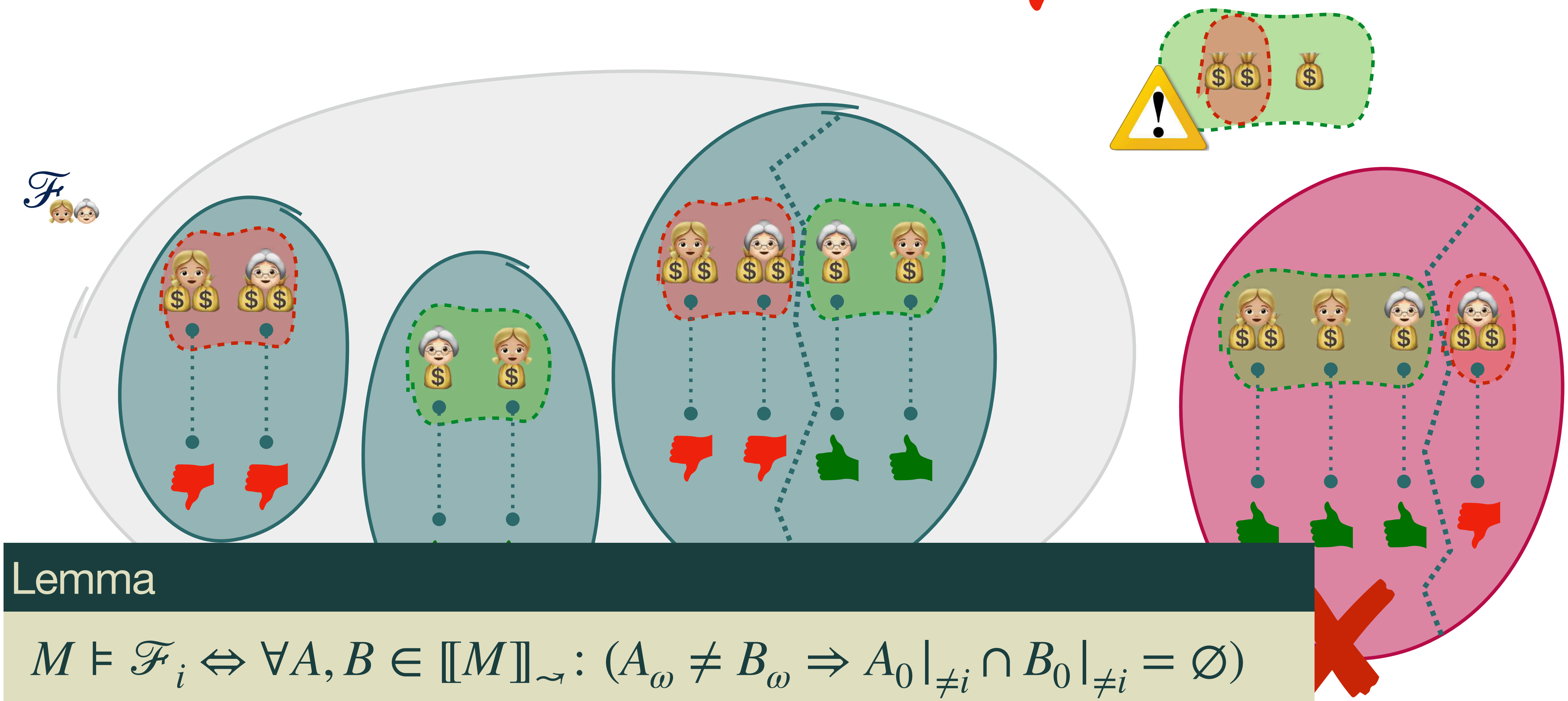
$[\![M]\!]_{\rightsquigarrow}$

💡 to reason about dependency fairness **we do not need to consider all intermediate computations** between the initial and final states of a trace (if any)

Dependency Semantics

💡 **partitioning with respect to the outcome** classification **induces a partition of the** space of **values** of the input nodes ***used*** **for classification**





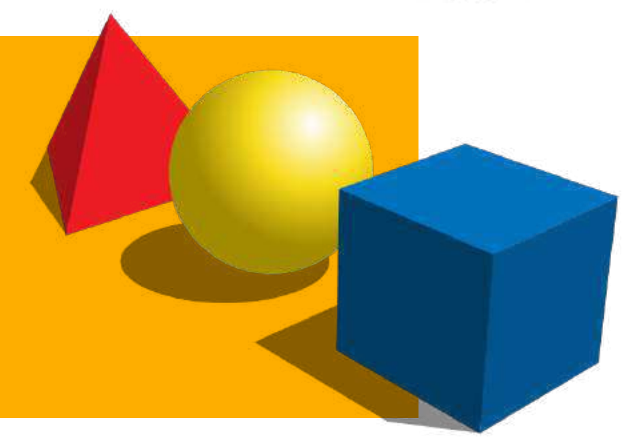
Lemma

$$M \models \mathscr{F}_i \Leftrightarrow \forall A, B \in [\![M]\!]_{\rightsquigarrow} : (A_\omega \neq B_\omega \Rightarrow A_0|_{\neq i} \cap B_0|_{\neq i} = \emptyset)$$

Abstract Interpretation Recipe

practical tools
targeting specific programs

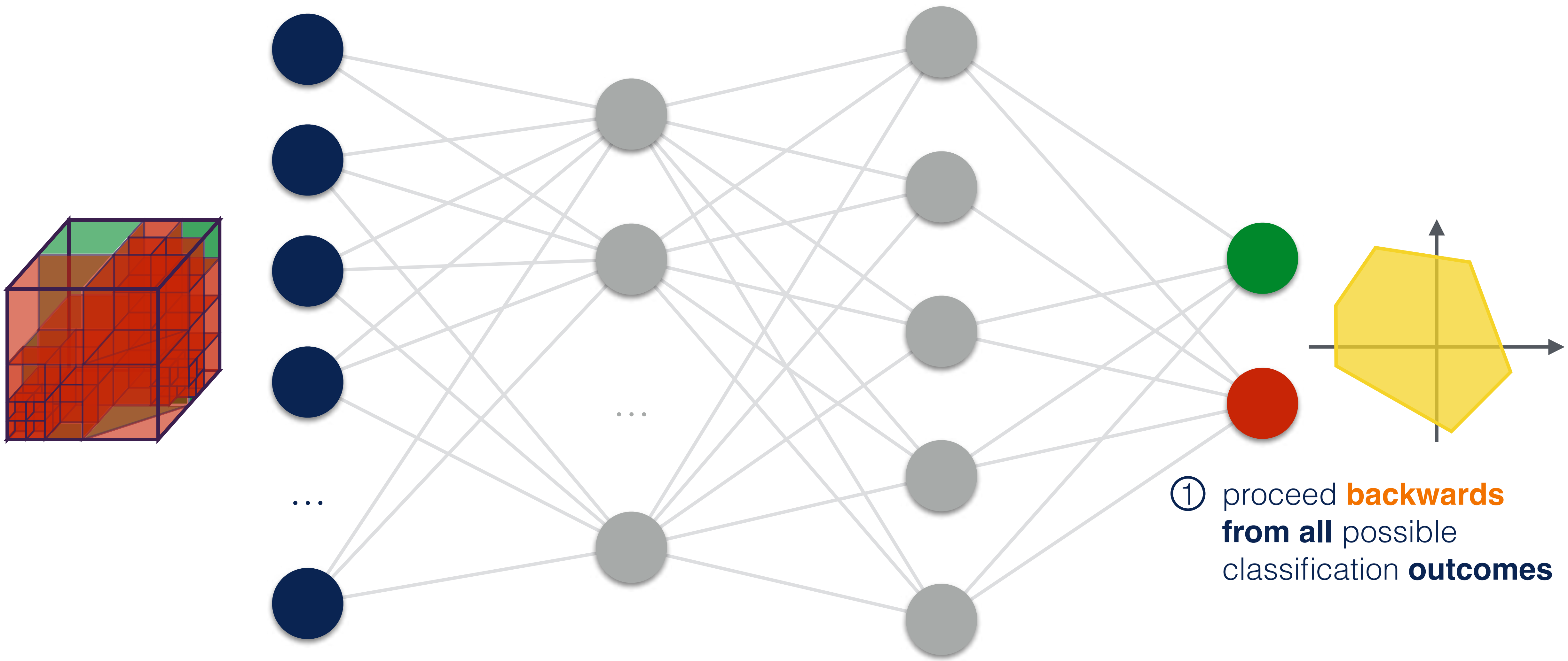
algorithmic approaches
to decide program properties



mathematical models
of the program behavior

Naïve Backward Analysis

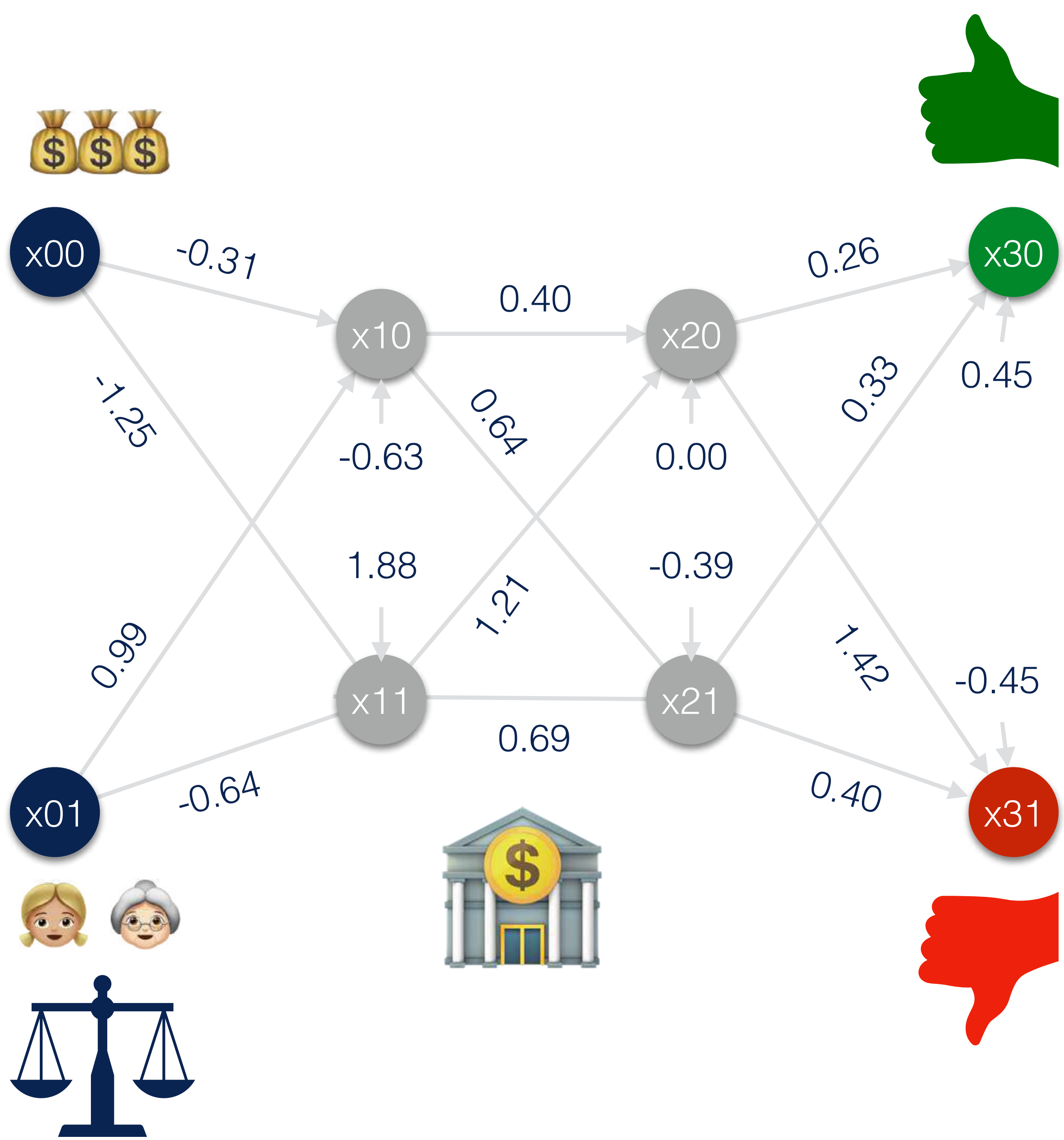
② **forget** the values of the **sensitive input** nodes



① proceed **backwards** **from all** possible classification **outcomes**

③ check for **intersection**:
empty → **fair**
otherwise → **alarm**

Naïve Backward Analysis




```
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 ' '
```

too many disjunctions!

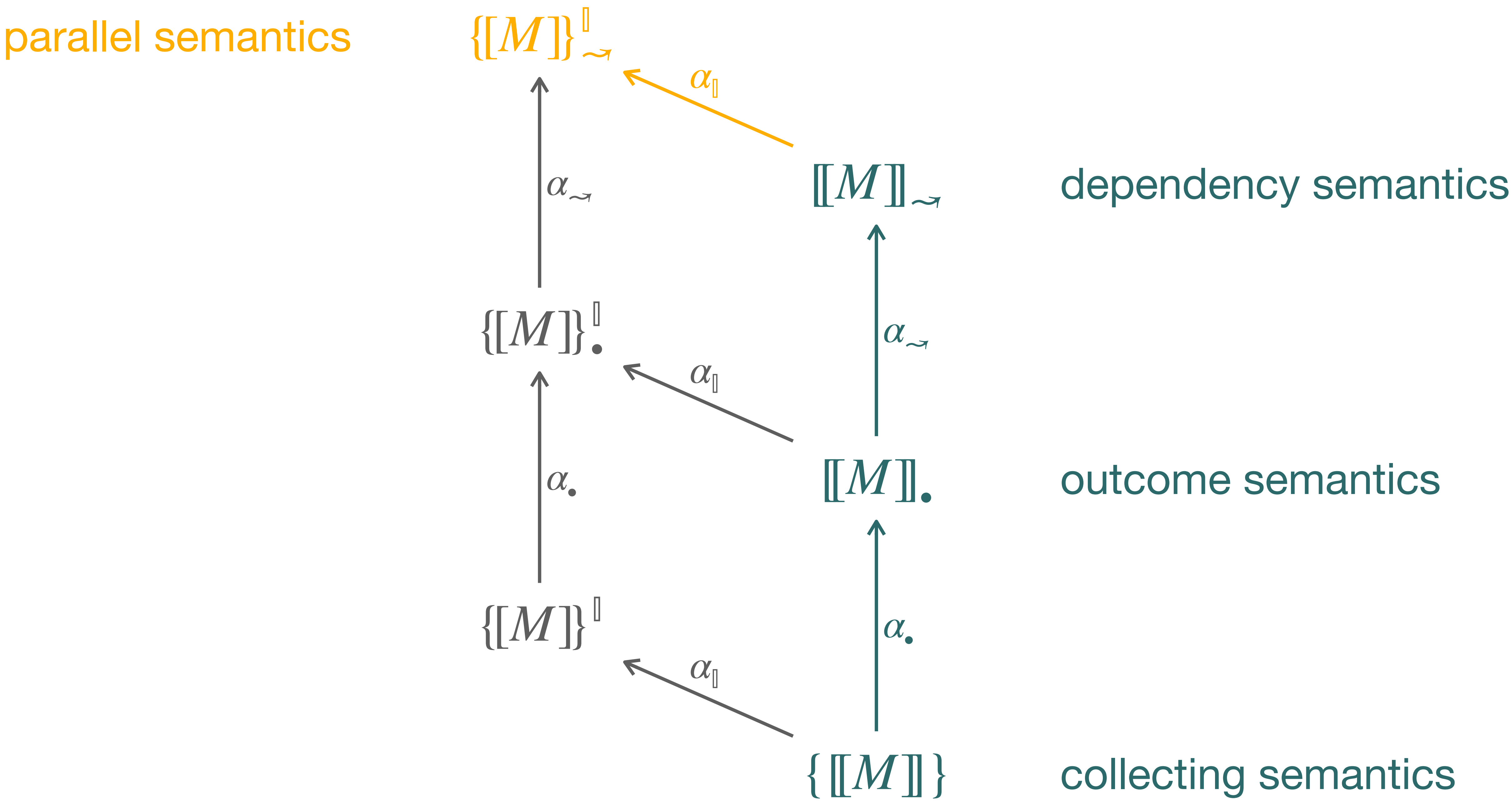
Abstract Interpretation Recipe

practical tools
targeting specific programs

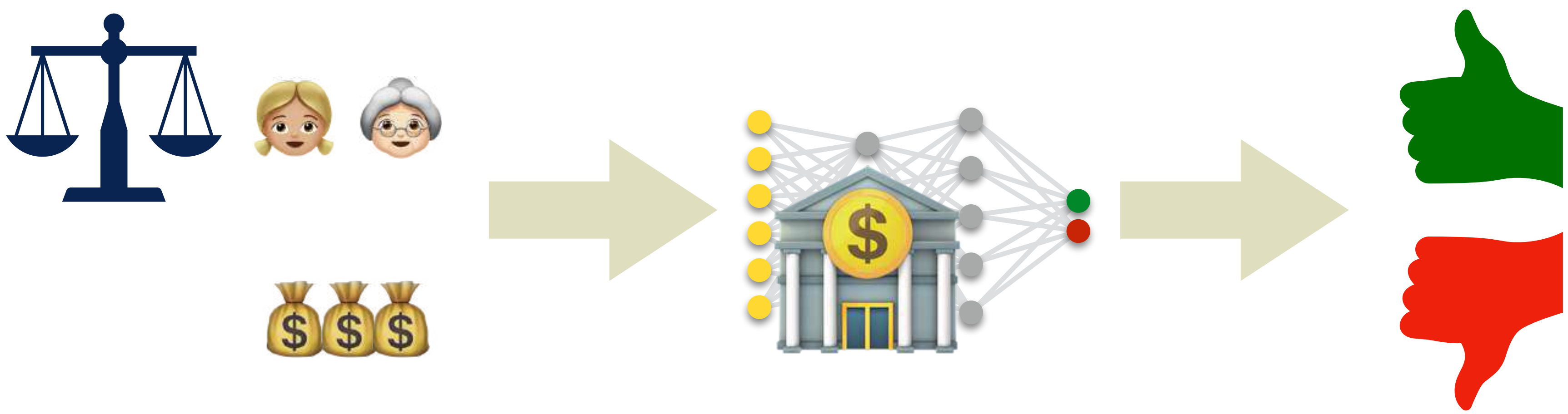
algorithmic approaches
to decide program properties

mathematical models
of the program behavior

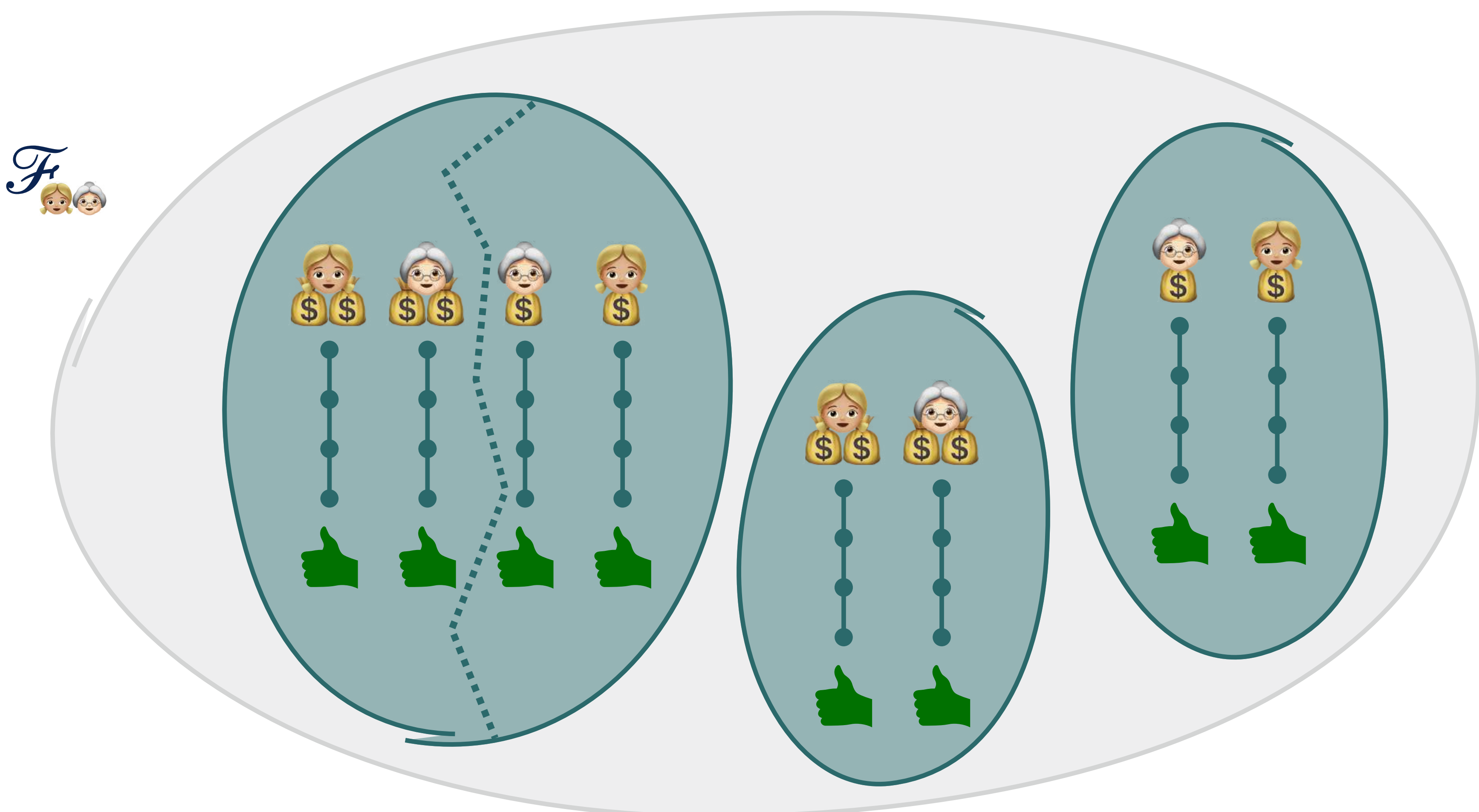
Hierarchy of Semantics



Parallel Semantics

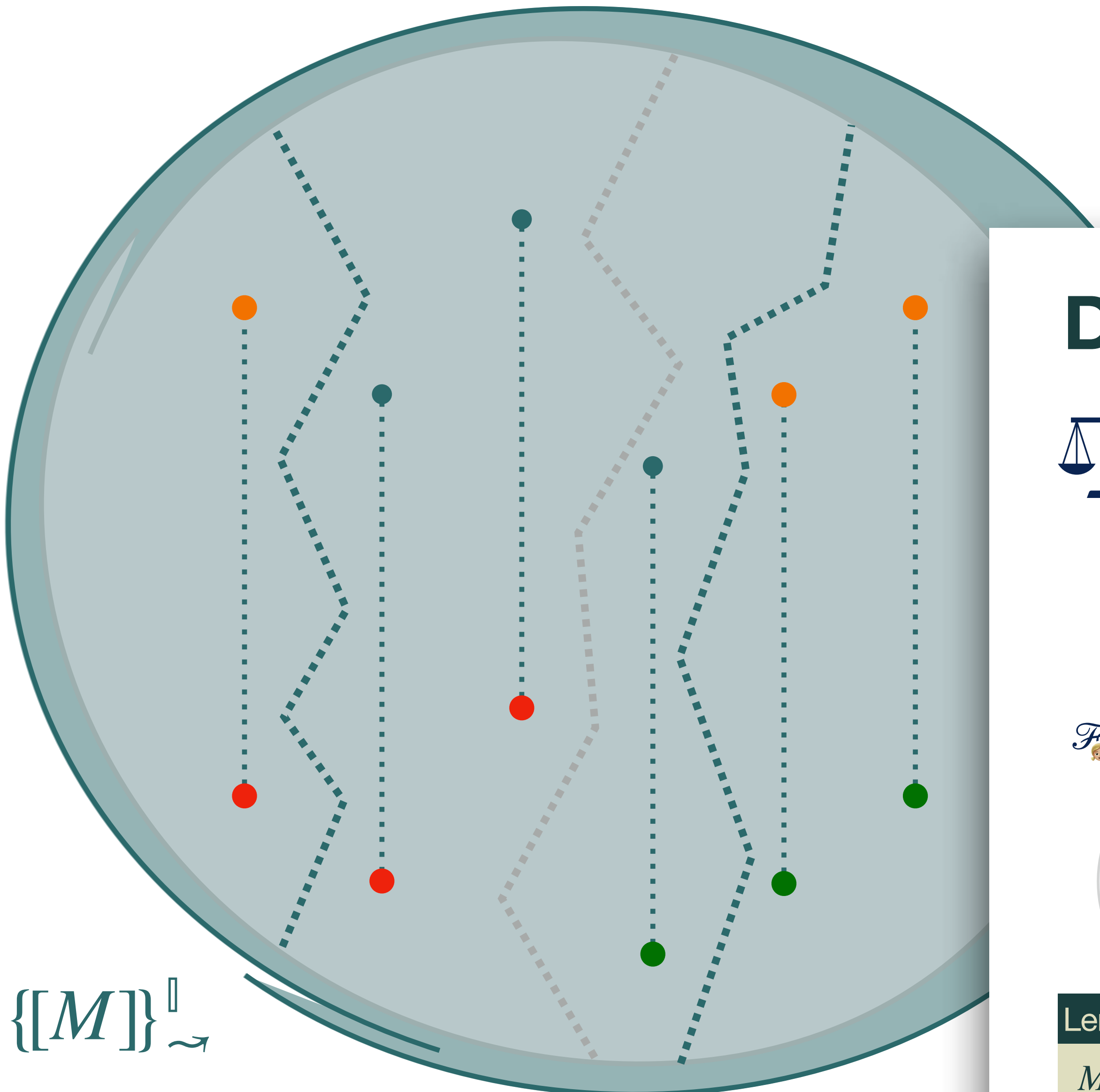


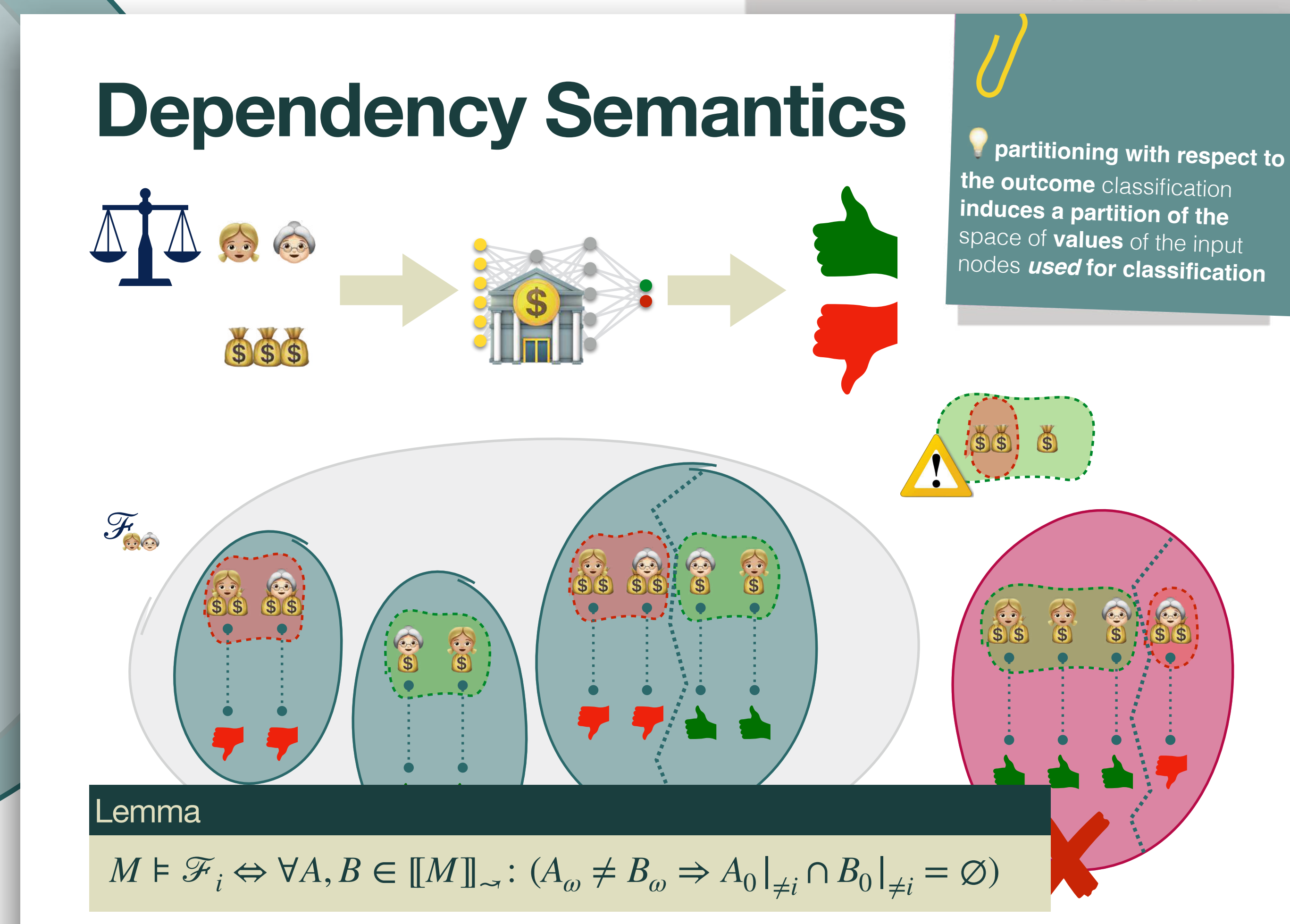
💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the non-sensitive inputs** yields sets of traces that also satisfy dependency fairness



Parallel Semantics

💡 **partitioning** a set of traces that satisfies dependency fairness **with respect to the non-sensitive inputs** yields sets of traces that also satisfy dependency fairness





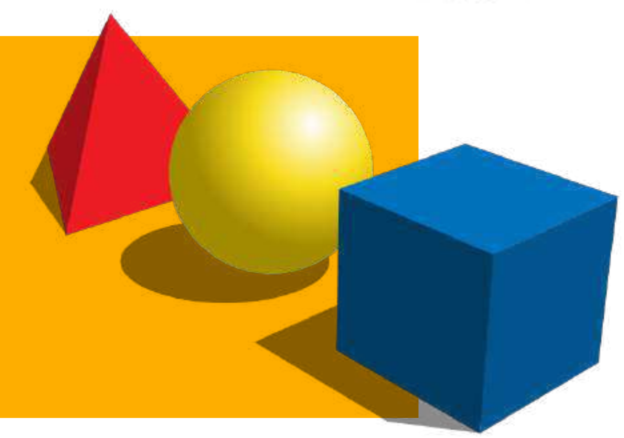
Lemma

$$M \models \mathcal{F}_i \Leftrightarrow \forall I \in \mathbb{I} : \forall A, B \in \{[M]\}^{\mathbb{I}}_{\rightsquigarrow} : (A^I_\omega \neq B^I_\omega \Rightarrow A^I_0|_{\neq i} \cap B^I_0|_{\neq i} = \emptyset)$$

Abstract Interpretation Recipe

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

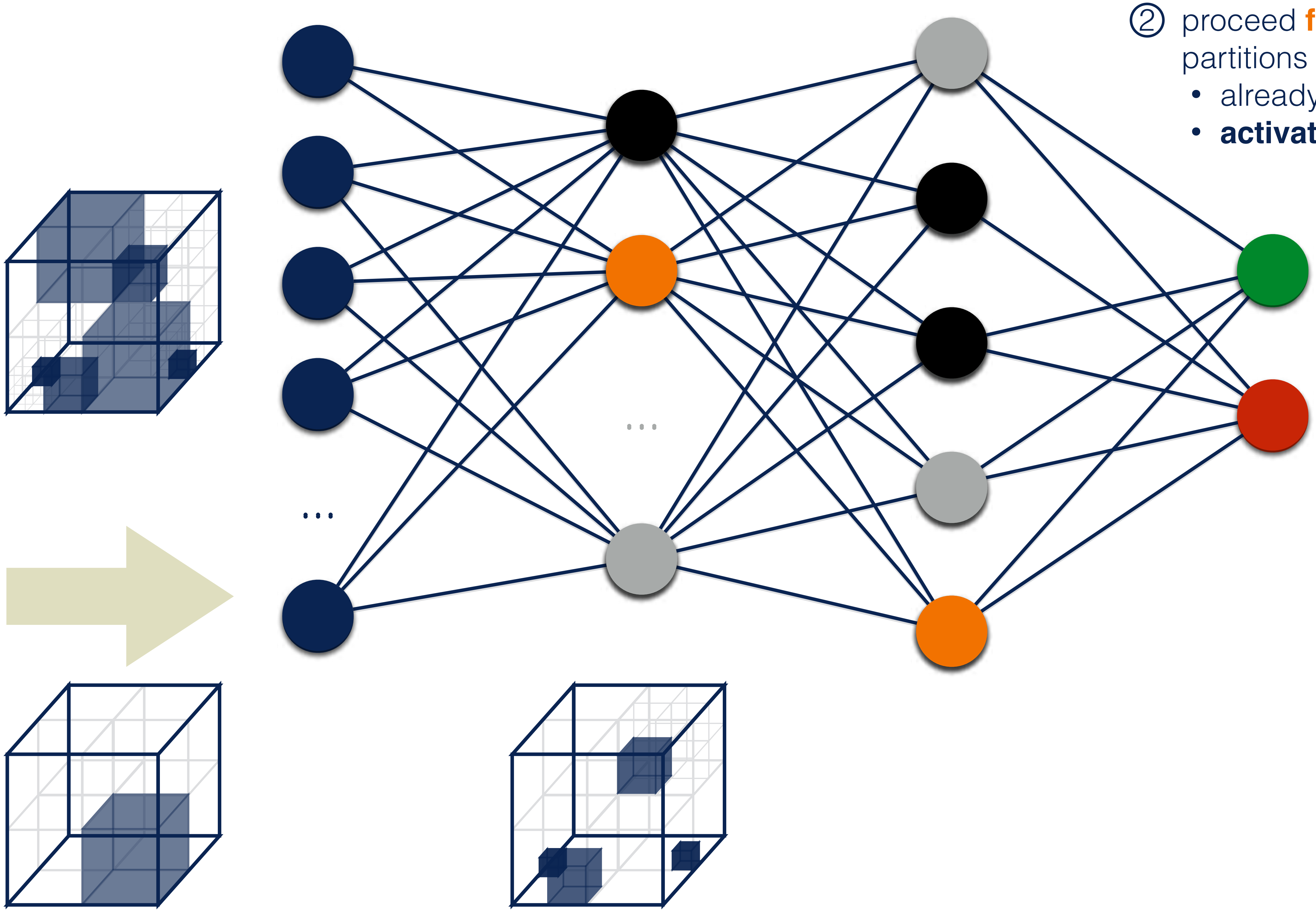


mathematical models
of the program behavior

Forward and Backward Analysis

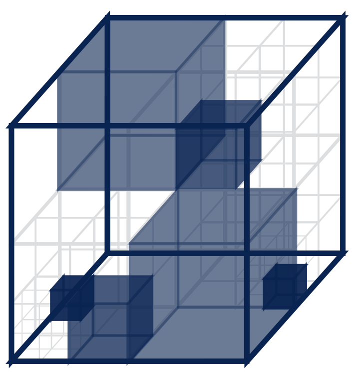
① **partition** the space of values of the **non-sensitive input** nodes

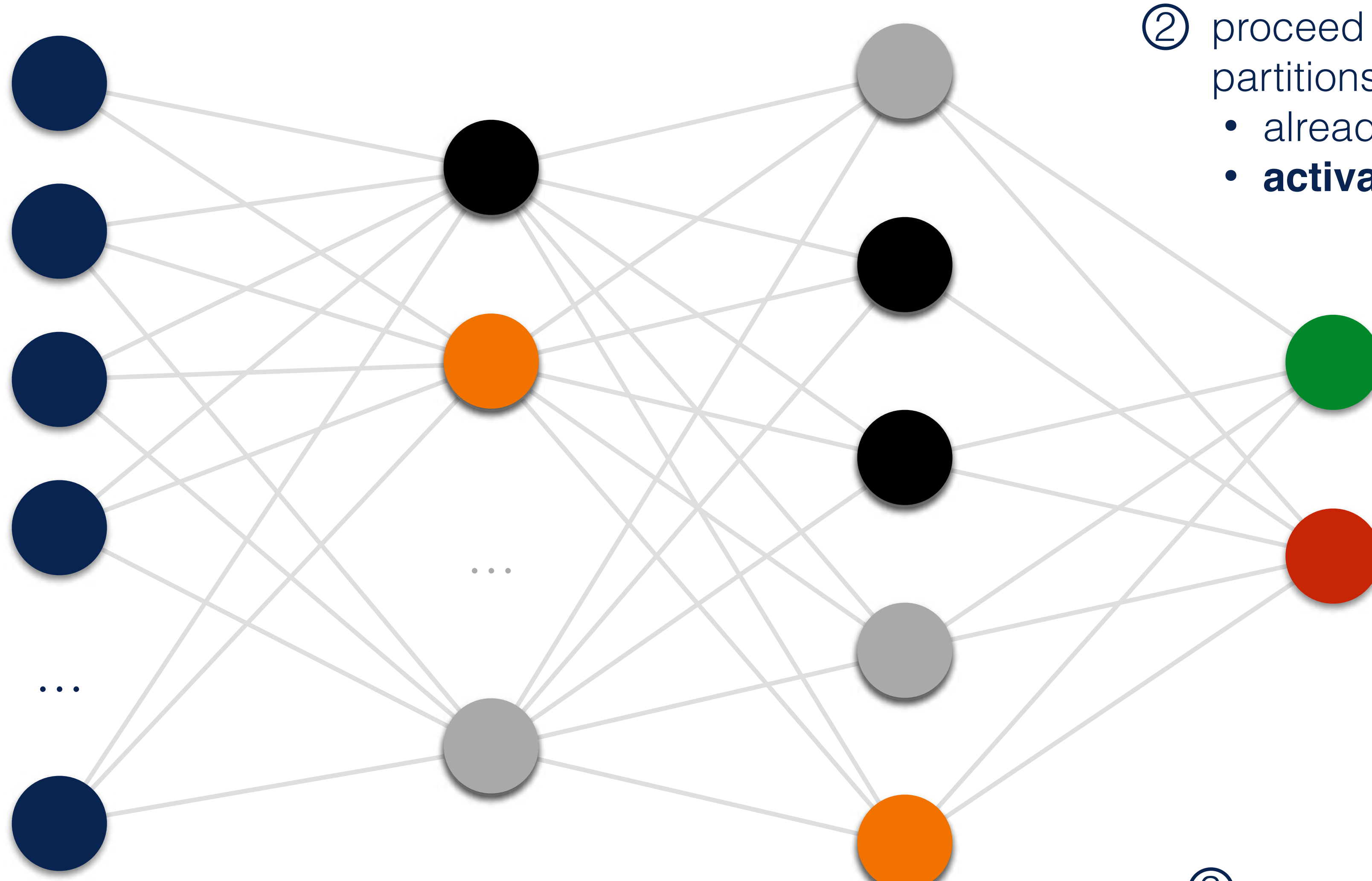
② proceed **forwards** from all partitions to find:
- already ✓ **fair** partitions
- **activation patterns**



Forward and Backward Analysis

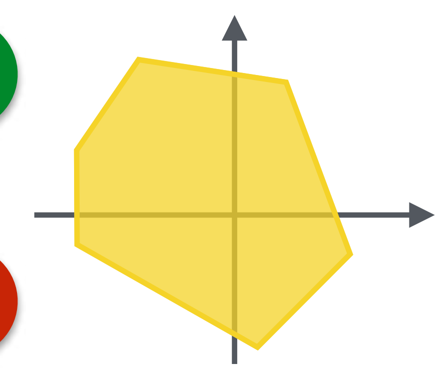
① **partition** the space of values of the **non-sensitive input** nodes



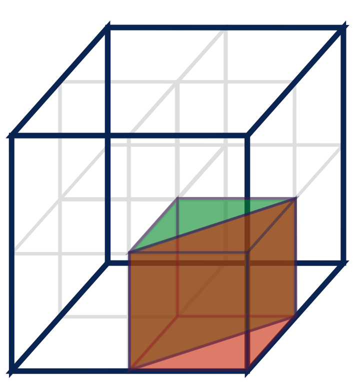


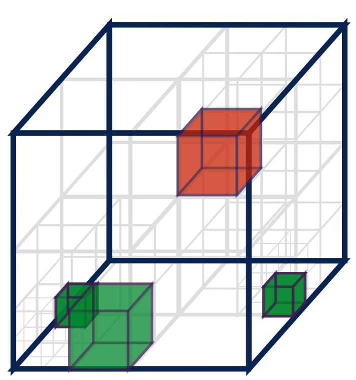
② proceed **forwards** from all partitions to find:

- already ✓ **fair** partitions
- **activation patterns**



③ proceed **backwards** for each activation pattern





Abstract Interpretation Recipe

practical tools
targeting specific programs

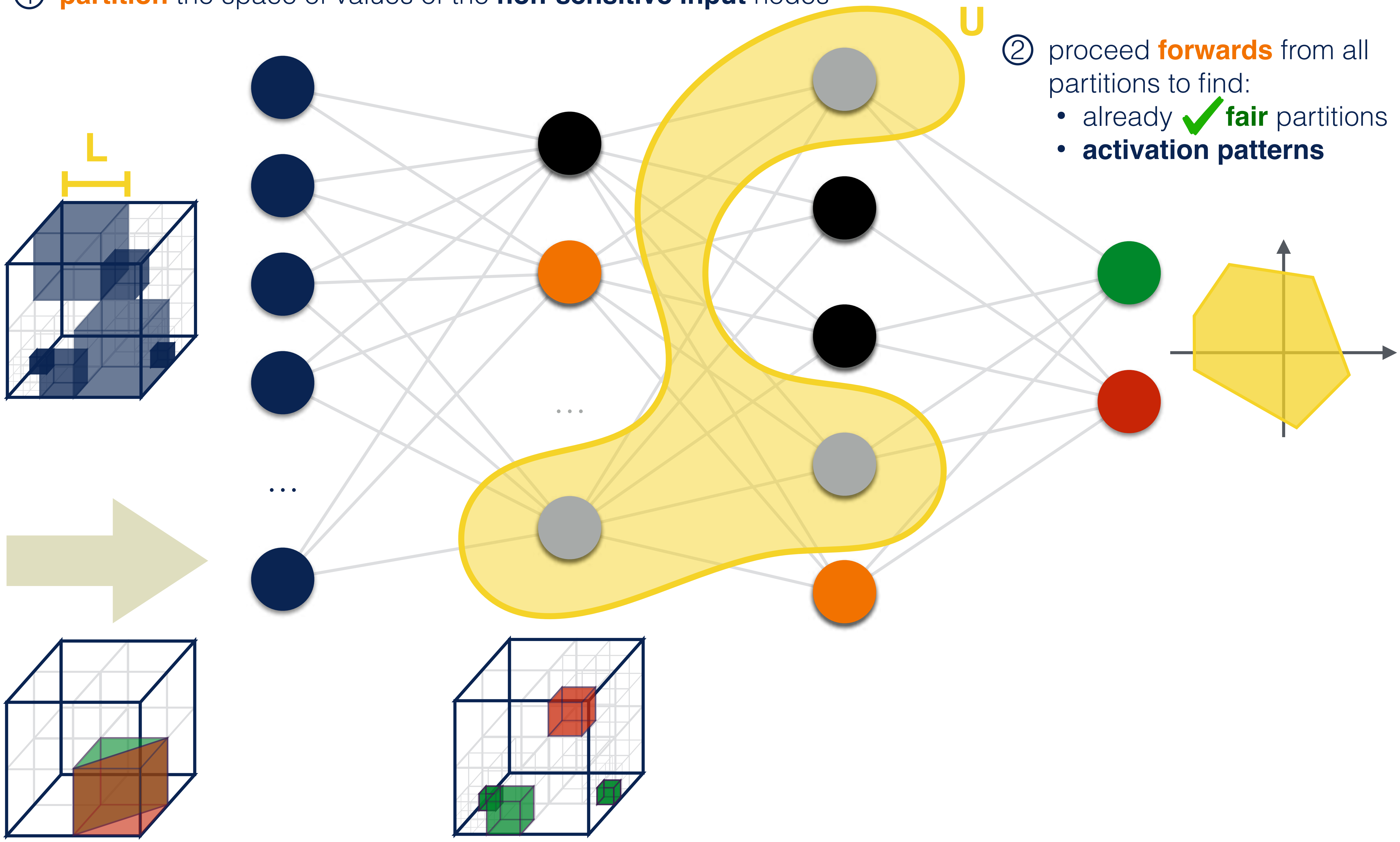
algorithmic approaches
to decide program properties

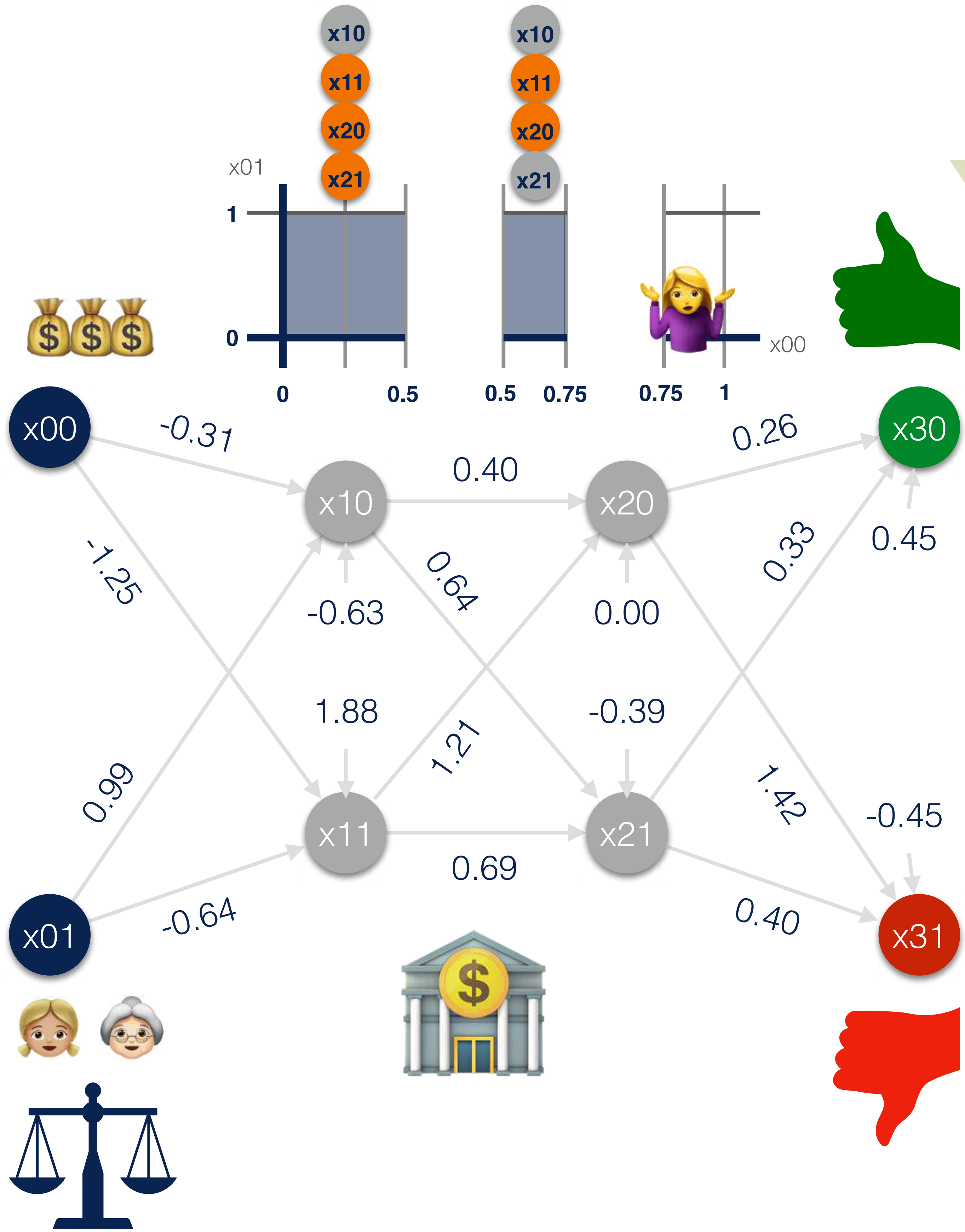
mathematical models
of the program behavior

Iterative Forward Analysis

① **partition** the space of values of the **non-sensitive input** nodes

② proceed **forwards** from all partitions to find:
- already ✓ **fair** partitions
- **activation patterns**

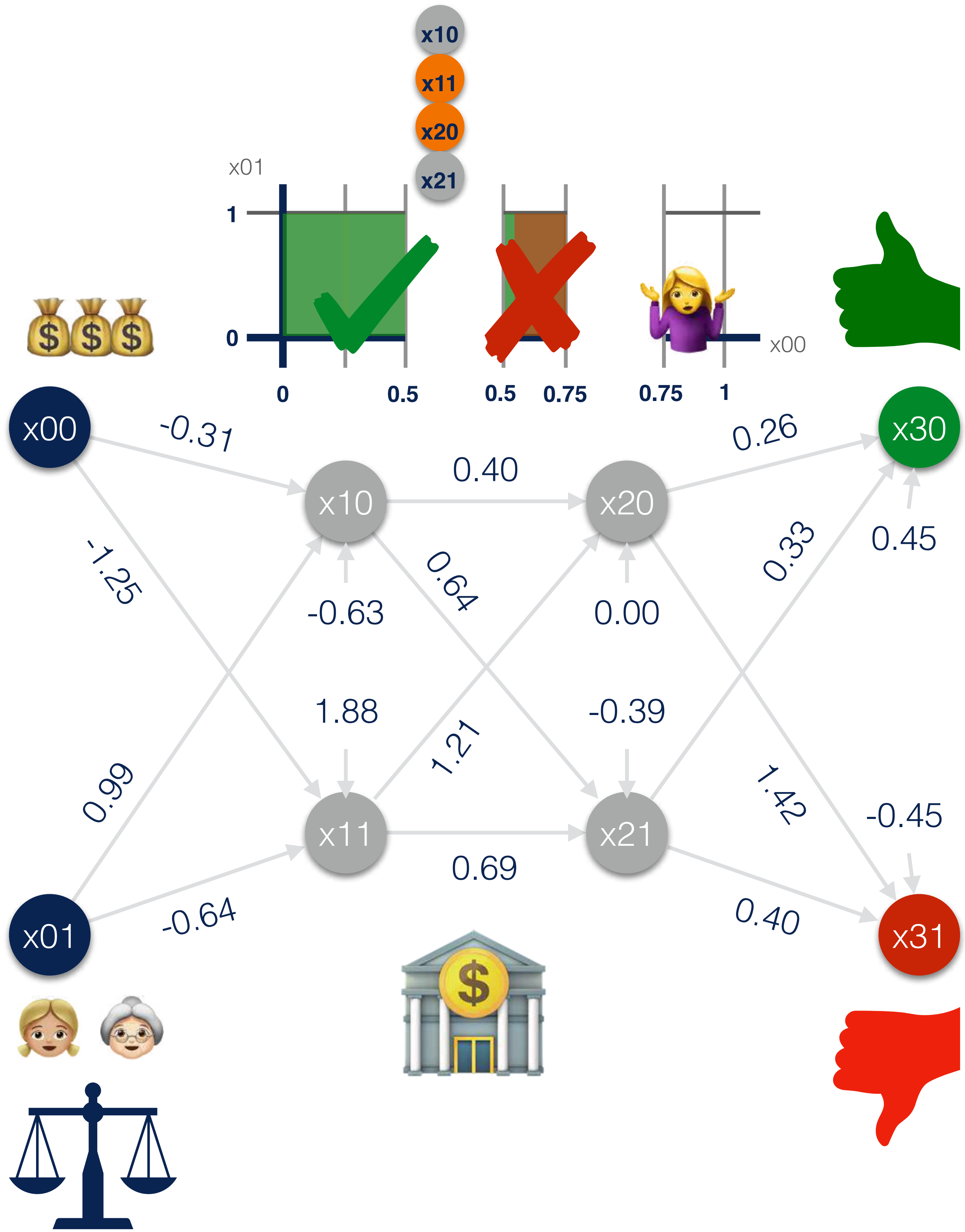




L = 0.25

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 '👎'
```

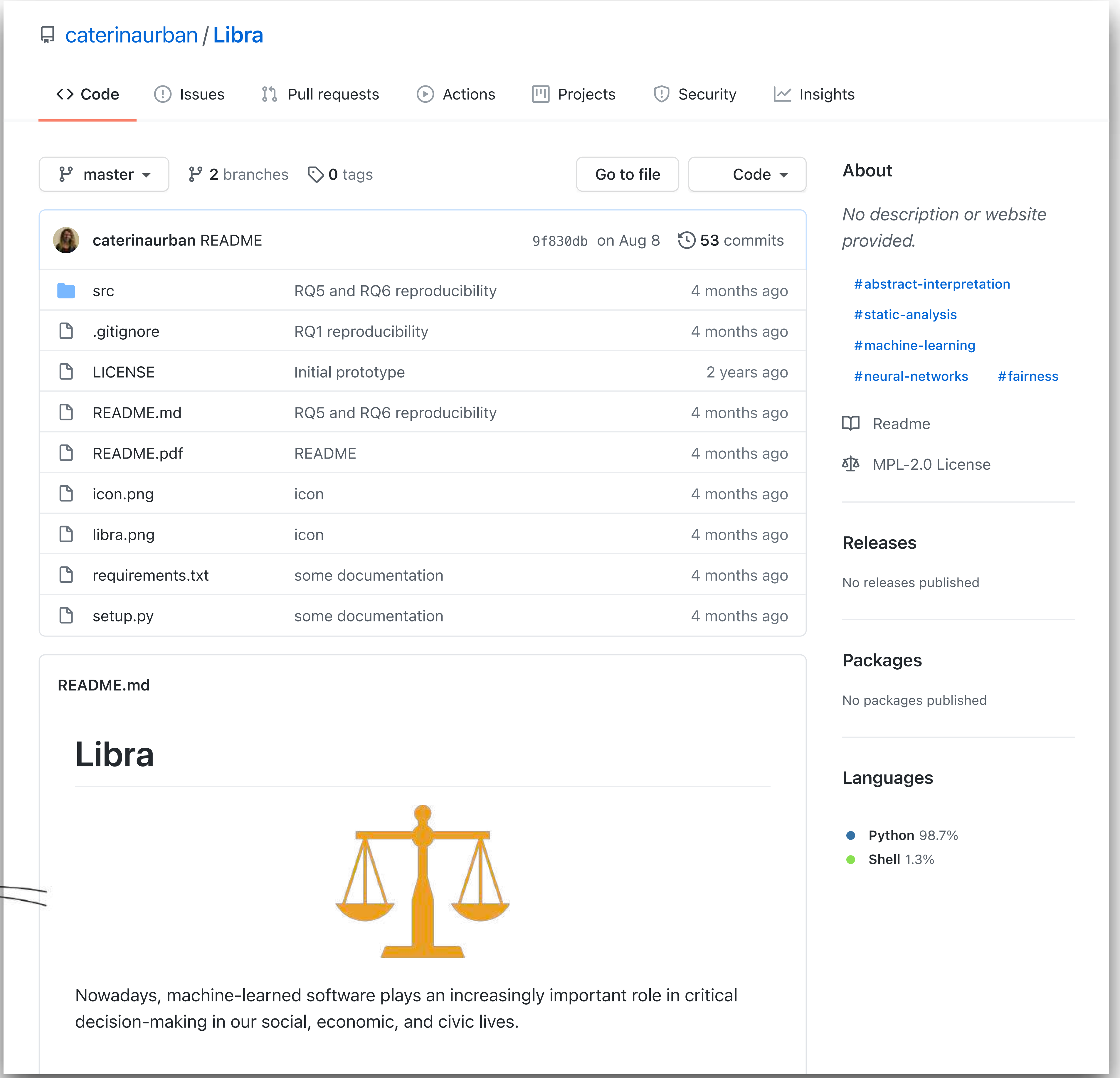


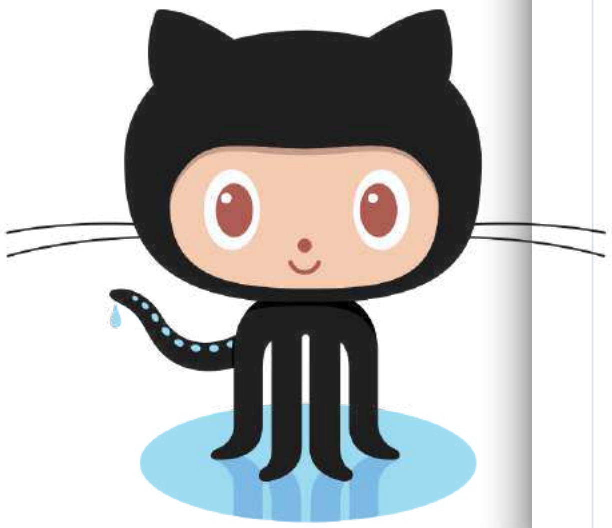
L = 0.25

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 ' '
```

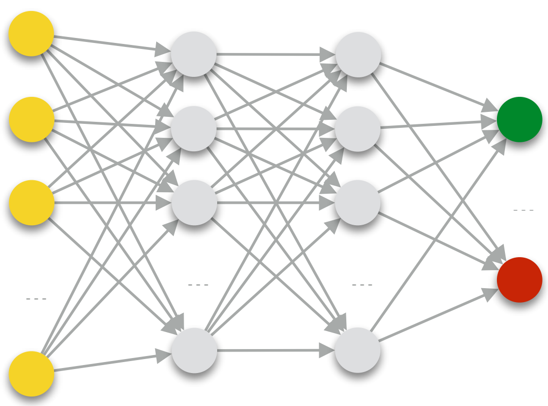
Libra


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

Japanese Credit Screening Dataset

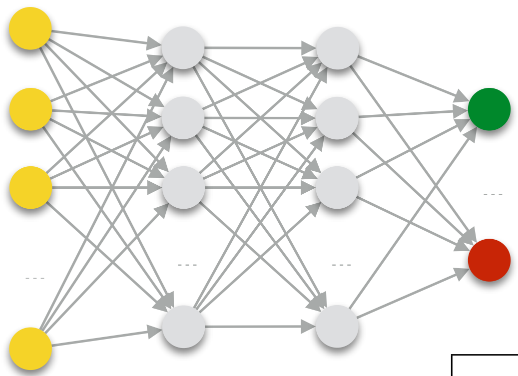


17 inputs
4 HL * 5 N
2 classes
86% accuracy

L	U	⬟ BOXES INPUT	|C|	|F|		TIME	▲ SYMBOLIC INPUT	|C|	|F|		TIME	⋆ DEEPPOLY 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

Seeded Bias

German Credit Dataset (L = 0)



17 inputs
4 HL * 5 N
2 classes
71% accuracy

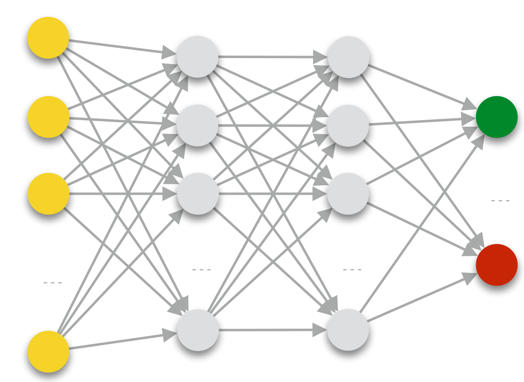
17 inputs
4 HL * 5 N
2 classes
65% accuracy

CREDIT	DEEPPOLY											
	FAIR DATA						BIASED DATA					
	U	BIAS	|C|	|F|		TIME	U	BIAS	|C|	|F|		TIME
≤ 1000	8	0.33%	170	21	25	3m 40s	**8**	**0.79%**	**260**	**42**	**53**	**5m 42s**
	6	**0.17%**	**211**	**10**	**10**	**4m 5s**	4	0.31%	218	9	20	1m 6s
	2	**0.09%**	**176**	4	5	**14s**	12	0.82%	271	53	61	18m 18s
	7	**0.15%**	**212**	9	9	**1m 31s**	4	**0.42%**	**242**	21	28	**1m 36s**
	3	**0.23%**	**217**	8	15	**32s**	**10**	**0.95%**	**260**	42	67	**3m 2s**
	12	0.30%	213	17	23	5m 45s	**2**	**0.41%**	**226**	**20**	**26**	**1m 56s**
	6	**0.20%**	**193**	11	11	**52s**	**3**	**0.48%**	**228**	**19**	**34**	**39s**
	5	**0.16%**	**193**	9	10	**10s**	**1**	**0.09%**	**206**	**5**	**5**	**51s**
MIN		0.09%				10s		0.09%				39s
MEDIAN		0.19%				1m 12s		0.45%				1m 46s
MAX		0.33%				5m 45s		0.95%				18m 18s
> 1000	**10**	12.08%	321	85	150	**10m 30s**	**11**	27.59%	498	234	333	**1h 16m 41s**
	11	7.43%	329	75	125	**22m 33s**	7	30.77%	394	70	228	**6m 34s**
	2	2.21%	217	15	16	39s	7	33.17%	435	185	327	**6h 51m 50s**
	10	4.29%	239	24	33	**4m 4s**	6	16.45%	448	162	260	**18m 25s**
	4	9.73%	268	29	87	4m 0s	**13**	30.17%	418	141	332	**43m 12s**
	14	14.96%	403	116	231	**1h 9m 45s**	5	17.24%	460	91	217	**12m 53s**
	7	5.83%	313	92	115	**4m 17s**	8	19.23%	363	79	189	7m 24s
	9	4.61%	264	50	74	**5m 38s**	2	4.52%	331	45	95	4m 44s
MIN		2.21%				39s		4.52%				4m 44s
MEDIAN		6.63%				4m 58s		23.41%				15m 39s
MAX		14.96%				1h 9m 45s		31.17%				6h 51m 50s

Bias Queries

ProPublica COMPAS Dataset (L = 0)

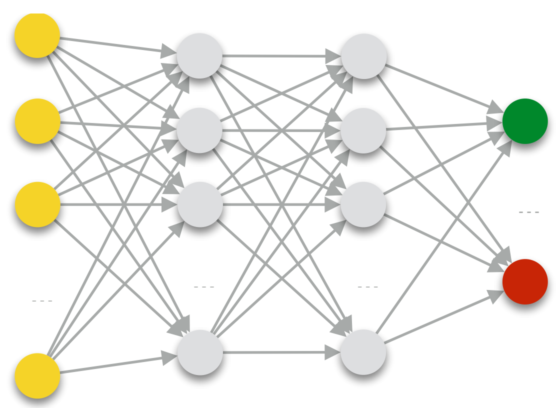
QUERY	DEEPPOLY											
	FAIR DATA						BIASED DATA					
	U	BIAS	|C|	|F|		TIME	U	BIAS	|C|	|F|		TIME
AGE < 25 RACE BIAS?	**10**	**0.23%**	**71**	**18**	**20**	**1h 11m 43s**	10	0.83%	43	15	33	2h 5m 5s
	10	**0.75%**	**33**	**14**	**16**	**10m 33s**	**10**	**6.48%**	**63**	**25**	**34**	**8m 46s**
	10	0.22%	34	17	22	52m 29s	**10**	**1.15%**	**33**	**10**	**14**	**11m 58s**
	10	**0.24%**	**118**	**28**	**29**	**42m 2s**	10	0.42%	31	13	30	10m 51s
	10	0.31%	117	49	54	1h 0m 2s	**10**	**0.12%**	**37**	**11**	**16**	**18m 18s**
	10	0.33%	59	18	21	53m 29s	**10**	**2.27%**	**33**	**16**	**24**	**1h 4m 35s**
	10	**1.19%**	**39**	**17**	**23**	**9m 39s**	**10**	**3.41%**	**133**	**92**	**102**	**33m 43s**
	10	**2.12%**	**33**	**17**	**31**	**5m 18s**	**10**	**0.18%**	**33**	**12**	**17**	**14m 58s**
MIN		0.22%				5m 18s		0.12%				8m 46s
MEDIAN		0.32%				47m 16s		0.99%				16m 38s
MAX		2.12%				1h 11m 43s		6.48%				2h 5m 5s
MALE AGE BIAS?	10	3.86%	242	96	180	2h 30m 23s	**10**	**5.22%**	**204**	**65**	**180**	**3h 25m 21s**
	10	**8.84%**	**100**	**45**	**77**	**19m 47s**	**10**	**12.38%**	**387**	**152**	**318**	**40m 49s**
	10	8.14%	204	47	143	28m 12s	**10**	**7.10%**	**181**	**63**	**142**	**20m 51s**
	10	**2.70%**	**563**	**168**	**232**	**1h 49m 9s**	10	6.90%	96	23	95	1h 21m 37s
	10	**4.65%**	**545**	**280**	**415**	**1h 33m 36s**	**10**	**6.14%**	**157**	**62**	**110**	**27m 43s**
	10	5.77%	217	68	154	1h 35m 25s	**10**	**8.10%**	**345**	**61**	**284**	**47m 9s**
	10	**7.76%**	**252**	**62**	**226**	**23m 10s**	**10**	**6.78%**	**251**	**141**	**223**	**50m 13s**
	10	8.70%	267	90	266	53m 26s	**10**	**12.88%**	**257**	**124**	**228**	**47m 46s**
MIN		2.70%				19m 47s		5.22%				20m 51s
MEDIAN		6.77%				1h 13m 31s		7.00%				47m 28s
MAX		8.84%				2h 20m 23s		12.88%				3h 25m 21s
CAUCASIAN PRIORS BIAS?	**11**	**2.18%**	**106**	**21**	**53**	**2h 32m 44s**	**11**	**2.92%**	**86**	**26**	**69**	**2h 26m 20s**
	7	**3.66%**	**105**	**38**	**55**	**18m 26s**	**11**	**6.95%**	**108**	**33**	**71**	**15m 29s**
	11	**2.73%**	**100**	**32**	**57**	**39m 5s**	14	4.43%	69	12	51	1h 47m 5s
	17	**2.19%**	**101**	**28**	**57**	**16h 19m 14s**	**7**	**3.40%**	**83**	**21**	**82**	**20m 1s**
	19	3.17%	86	30	53	52h 10m 2s	**13**	**3.09%**	**96**	**24**	**58**	**1h 8m 4s**
	11	**2.45%**	**94**	**26**	**52**	**2h 18m 42s**	**14**	**5.79%**	**99**	**45**	**87**	**1h 51m 2s**
	15	3.94%	87	29	52	2h 39m 18s	**17**	**5.10%**	**110**	**73**	**94**	**17h 48m 22s**
	15	**5.36%**	**90**	**35**	**89**	**3h 41m 16s**	**14**	**3.99%**	**97**	**38**	**65**	**1h 21m 8s**
MIN		2.18%				18m 26s		2.92%				15m 29s
MEDIAN		2.95%				2h 36m 1s		4.21%				1h 34m 7s
MAX		5.36%				52h 10m 2s		6.95%				17h 48m 22s



19 inputs
4 HL * 5 N
3 classes
55% | 56% accuracy

Scalability wrt Model Size

Adult Census Dataset (L = 0.5)



23 inputs	23 inputs	23 inputs	23 inputs	23 inputs
2 HL * 5 N	4 HL * 3 N	4 HL * 5 N	4 HL * 10 N	9 HL * 5 N
2 classes	2 classes	2 classes	2 classes	2 classes

|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

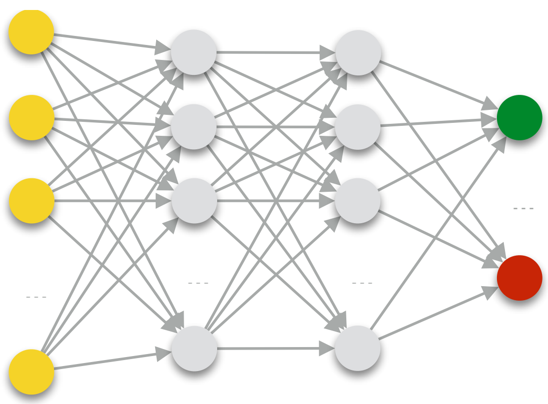
Scalability wrt Input Space Size

Adult Census Dataset (L = 0.25, U = 0.1 * |M|)

|M|	QUERY	BOXES INPUT	BOXES |C|	BOXES |F|		BOXES TIME	SYMBOLIC INPUT	SYMBOLIC |C|	SYMBOLIC |F|		SYMBOLIC TIME	DEEPPOLY INPUT	DEEPPOLY |C|	DEEPPOLY |F|		DEEPPOLY 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

Scalability-vs-Precision Tradeoff

Product Domain / Adult Census Dataset



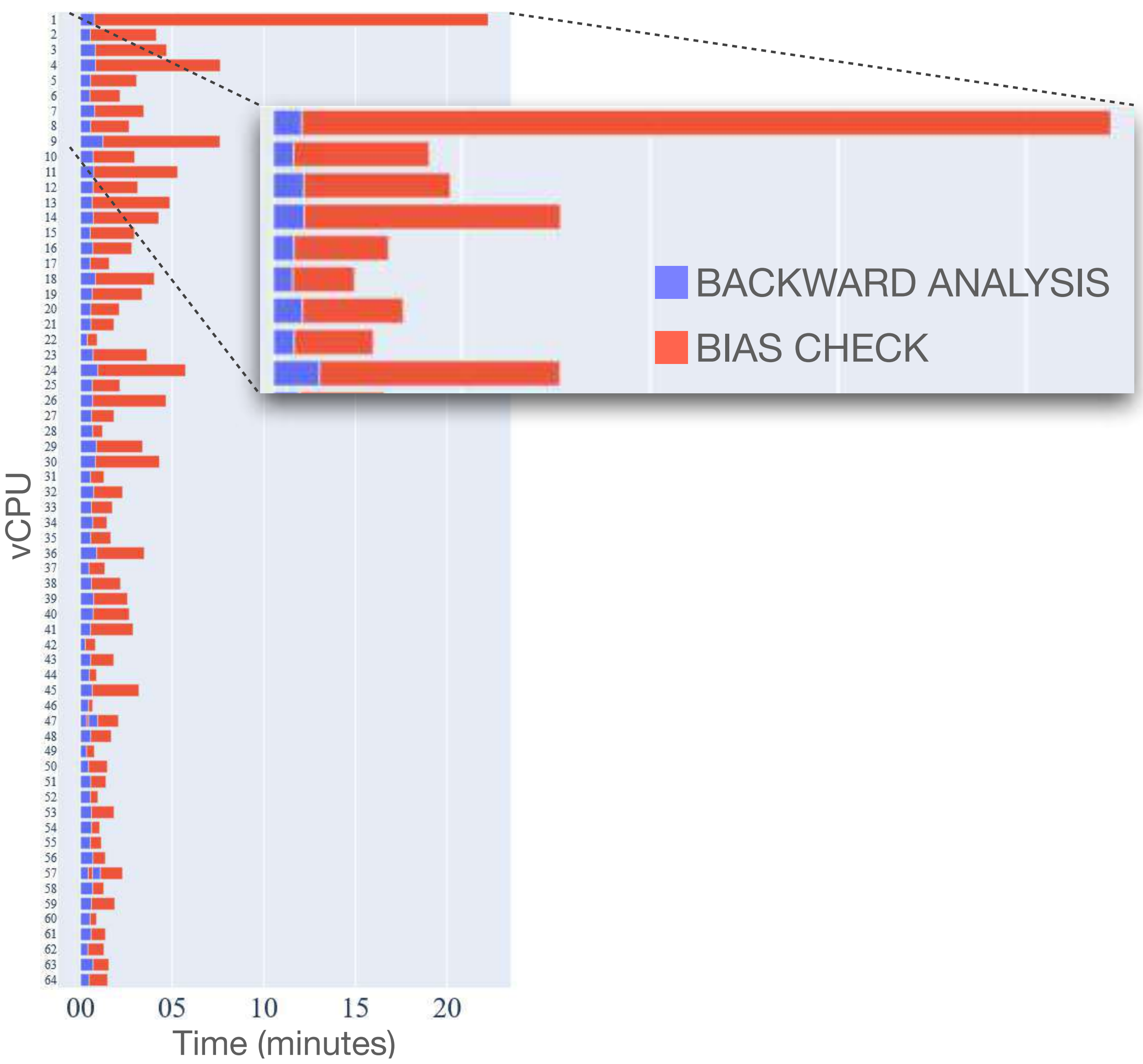
23 inputs
4 HL * 5 N
2 classes

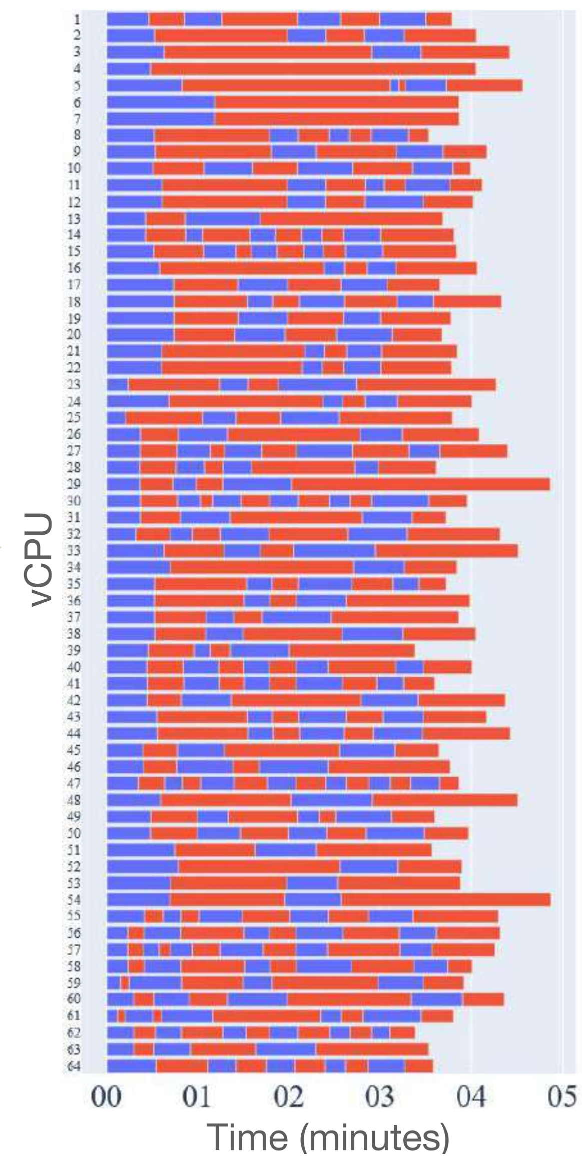
L	U	Intervals	Symbolic	DeepPoly	Neurify	**Product**	
0.5	3	37,9 %	48,8 %	48,9 %	46,5 %	**59,2 %**	**+ 10,3%**
	5	41,0 %	56,1 %	56,3 %	53,1 %	**68,2 %**	**+ 11,9%**
0.25	3	70,6 %	83,6 %	81,8 %	81,4 %	**87,0 %**	**+ 3,4%**
	5	83,1 %	91,7 %	91,6 %	92,3 %	**95,5 %**	**+ 3,2%**

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

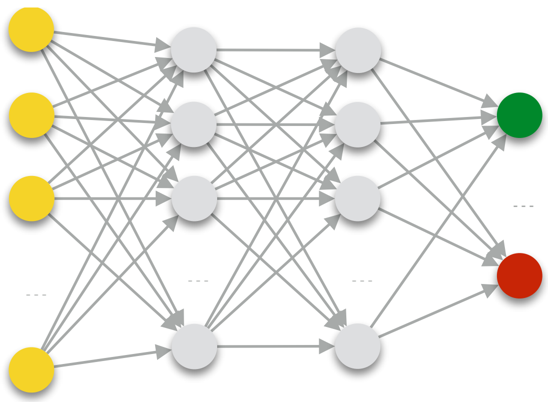
Perfect Parallelization





Scalability-vs-Precision Tradeoff

Perfect Parallelization / Adult Census Dataset



23 inputs
4 HL * 5 N
2 classes

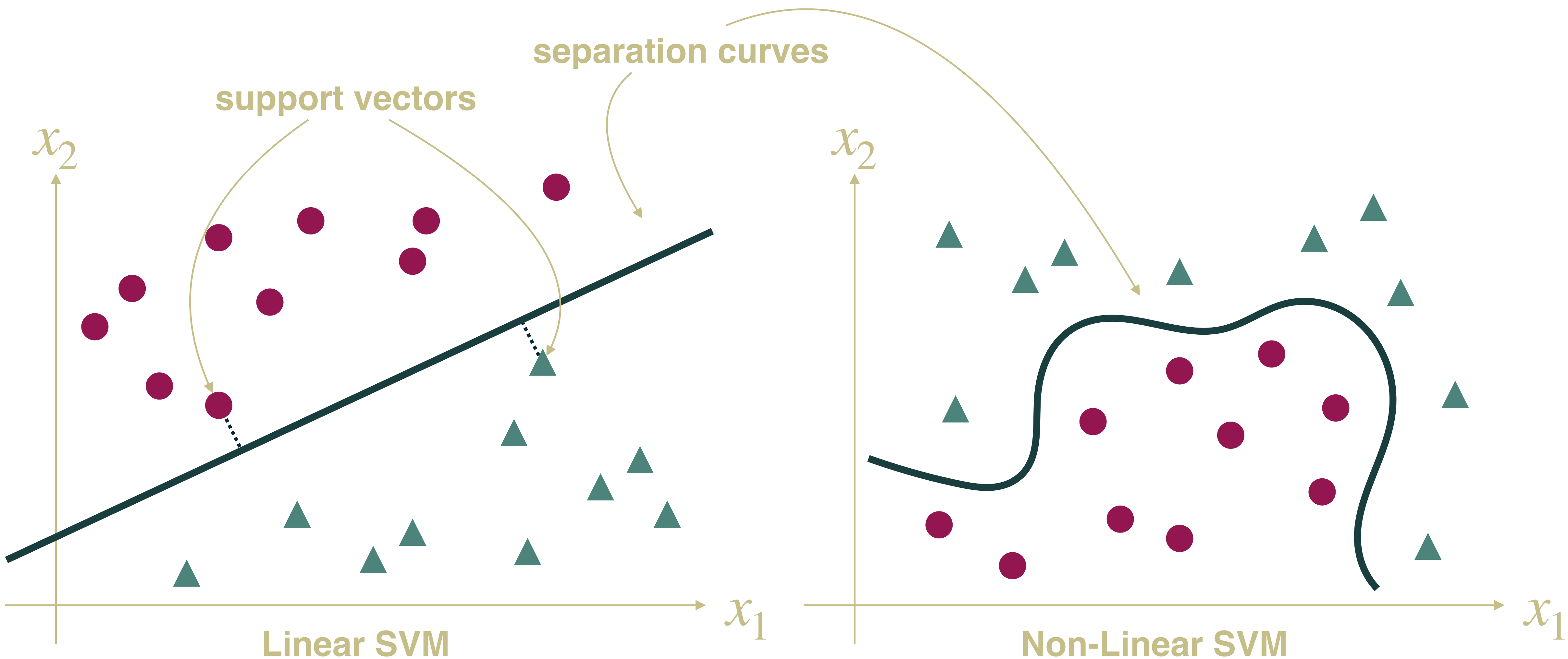
L	U	Intervals	Symbolic	DeepPoly	Neurify	**Product**
0.5	3	37,9 %	48,8 %	48,9 %	46,5 %	**59,2 %**
	5	41,0 %	56,1 %	56,3 %	53,1 %	**68,2 %**
0.25	3	70,6 %	83,6 %	81,8 %	81,4 %	**87,0 %**
	5	83,1 %	91,7 %	91,6 %	92,3 %	**95,5 %**

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 ML Models

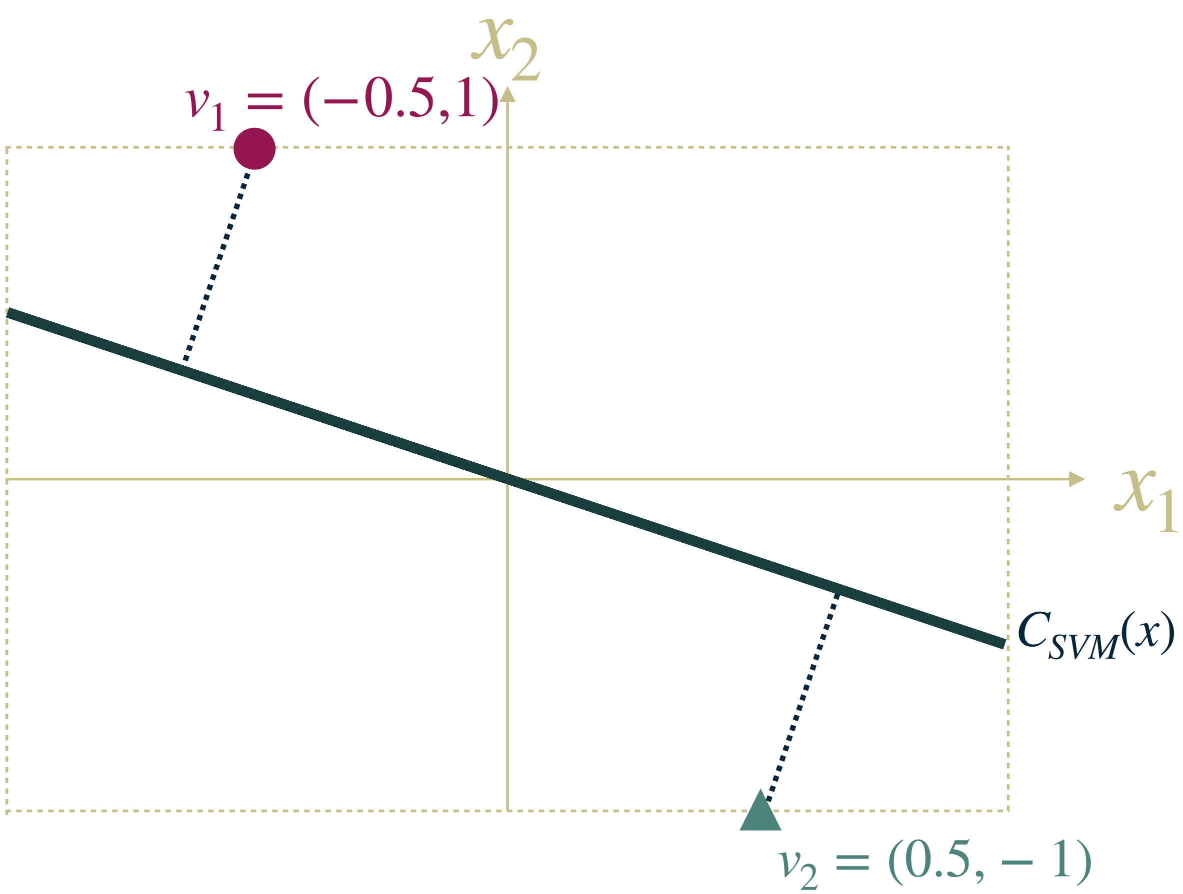
Support Vector Machines (SVMs)



Support Vector Machines (SVMs)

Example

● $\mapsto -1$
▲ $\mapsto 1$



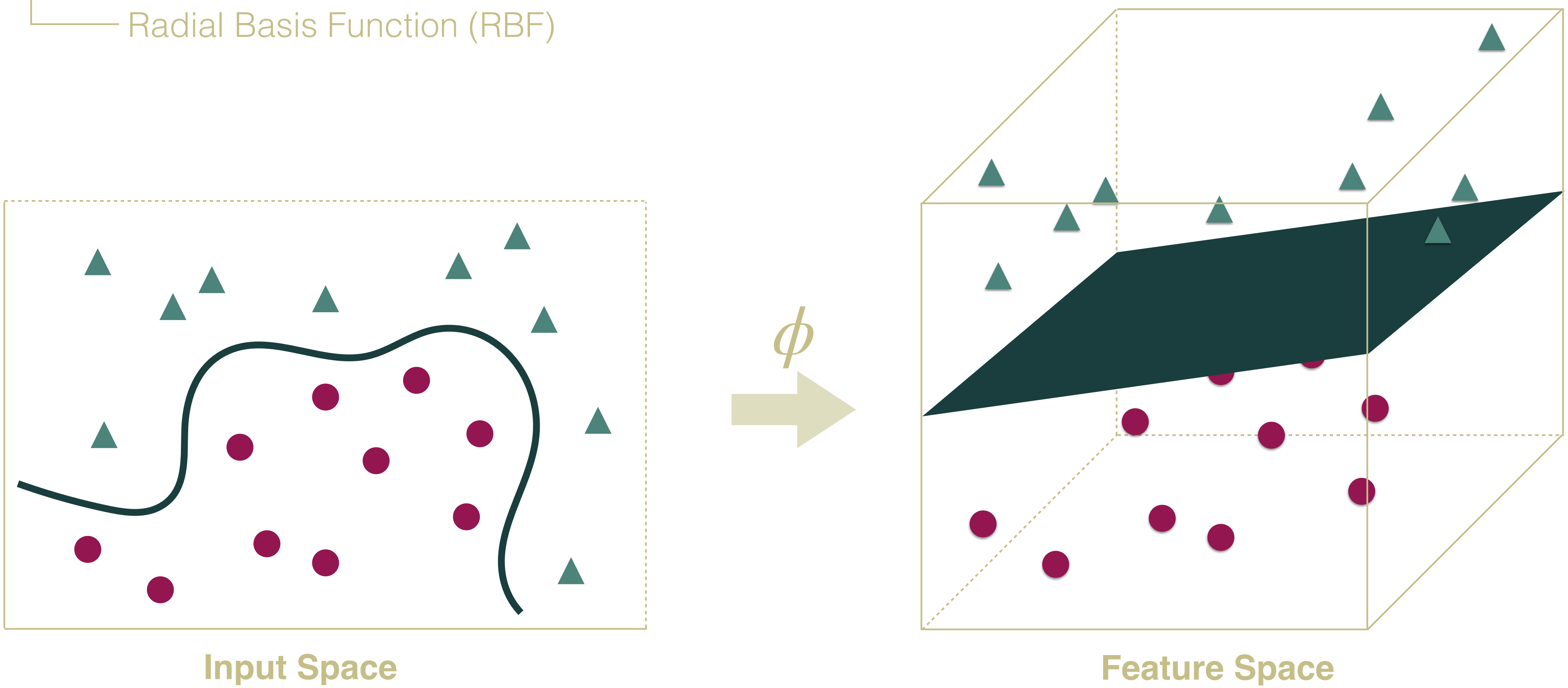
weights

$$C_{SVM}(x) = sgn\left(SVM(x)\right) = sgn\left(-1 * 0.5(v_1 \cdot x) + 1 * 0.5(v_2 \cdot x)\right)$$
$$= sgn\left(0.5x_1 - x_2\right)$$

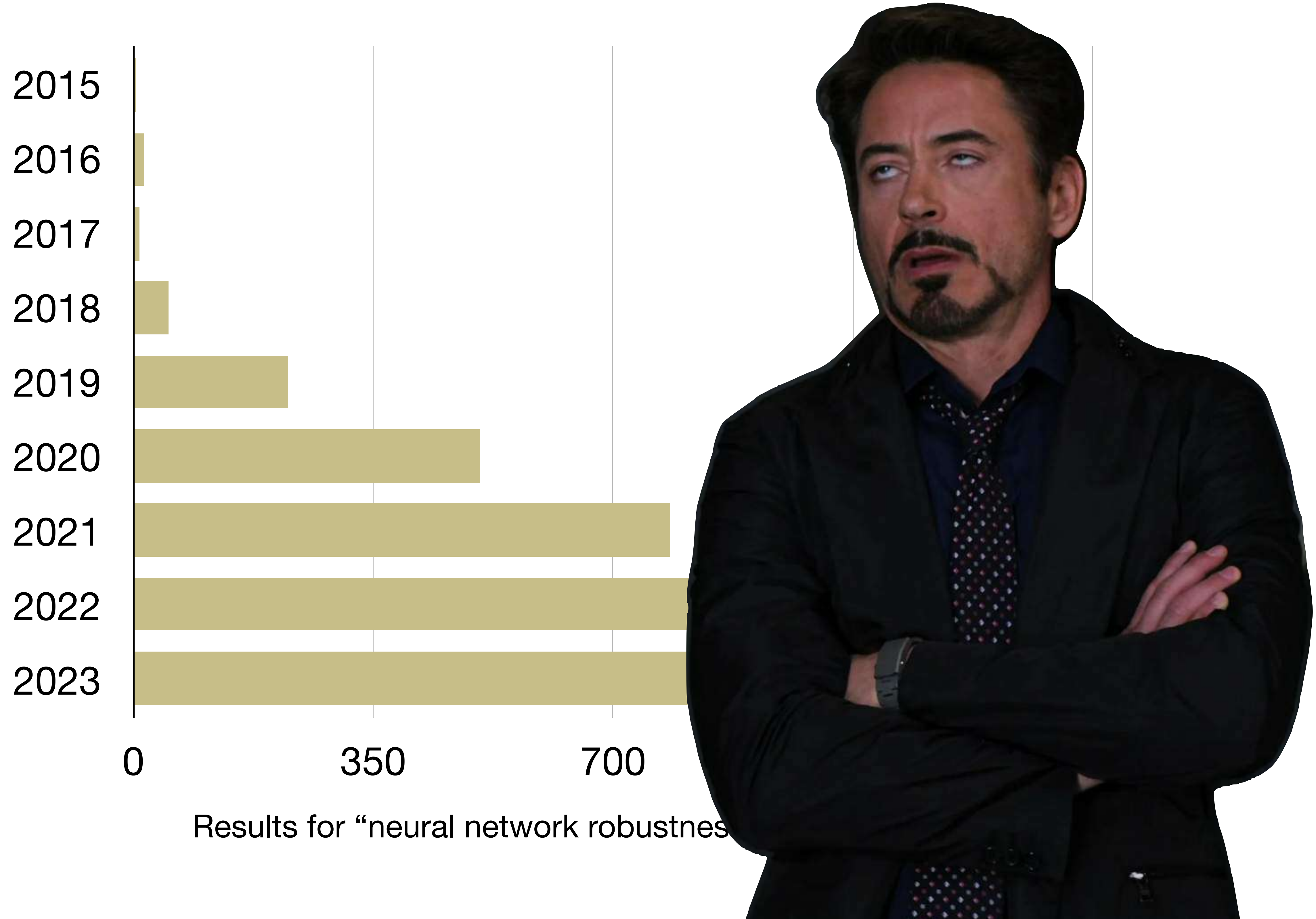
Non-Linear SVMs

Kernel Functions

- Polynomial
- Radial Basis Function (RBF)

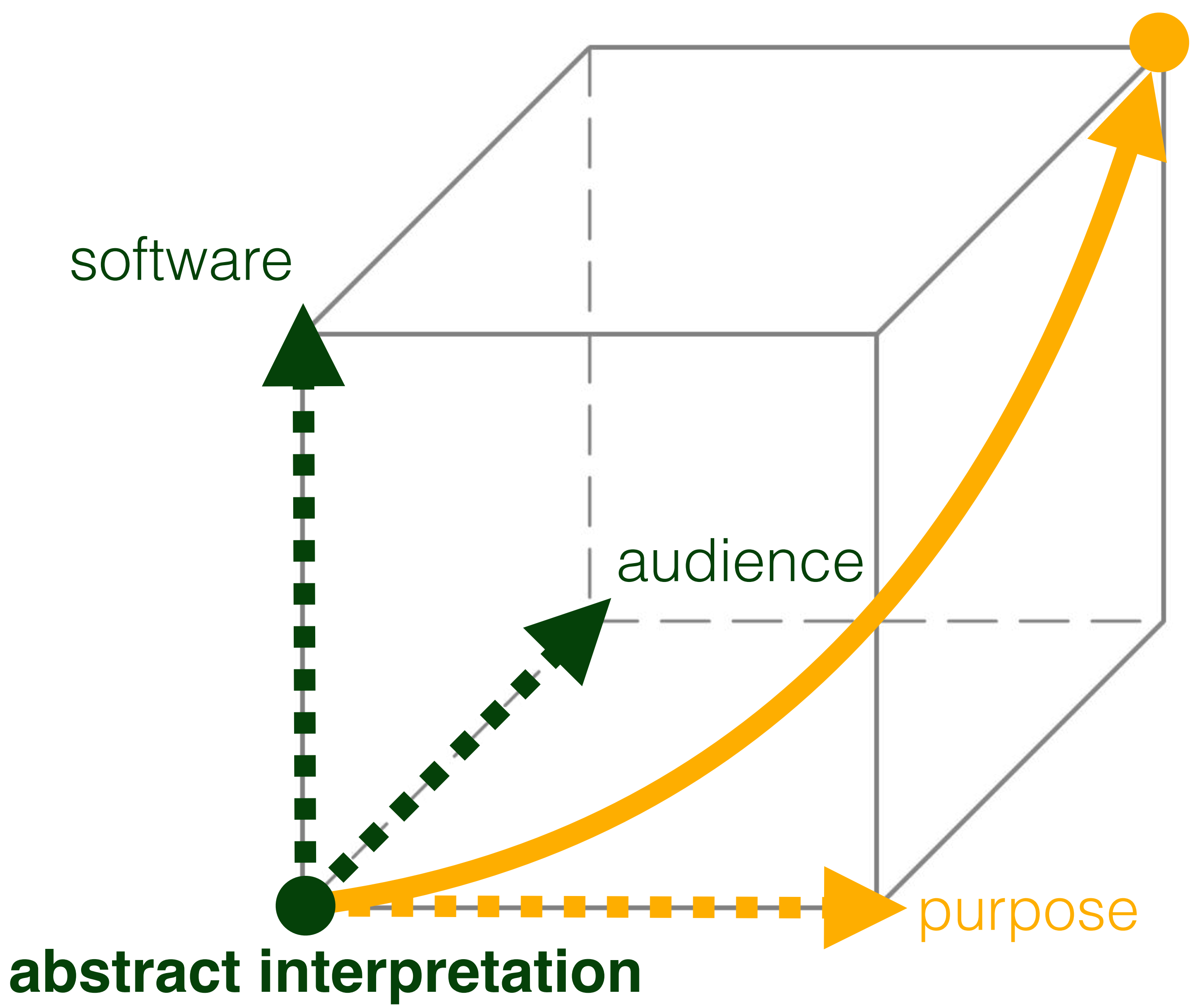


Formal Methods for ML



SVM Explainability

Explainability



Static Analysis Methods

Feature Importance Measures

Contribution of Input Features to Prediction

	Local	Global	Model- Specific	Model- Agnostic	Performance -Based	Effect -Based
Permutation Feature Importance (PFI)		X		X	X	
Partial Dependence (PD) Plots		X		X		X
Individual Conditional Expectation (ICE)		X		X		X
Accumulated Local Effects (ALE) Plots		X		X		X
Local Interpretable Model-Agnostic	X			X		X
SHapley Additive exPlanations (SHAP)	X			X		X
Individual Conditional Importance (ICI)	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 [Pal2024]

Why Another Feature Importance Measure?

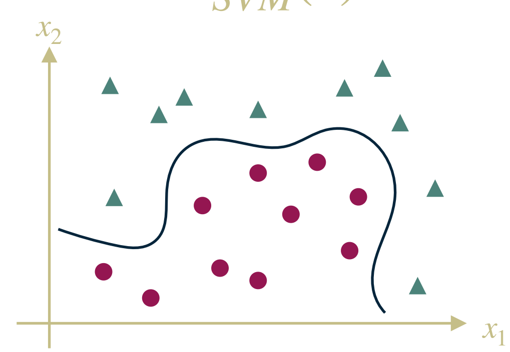
Permutation Feature Importance (PFI)	• **result may greatly vary depending on the dataset** • **resource intensive** when the number of featu[illegible] • **misleading result when features a**[illegible] • **quality of the result heavi**[illegible]**el accuracy**
Local Interpretable Model-Agnostic Explanations (LIME)	• **req**[illegible]**ngful optimal neighborhood**: [illegible]le and easily manipulable explanations • [illegible]**s that the decision boundary is linear at the local level**, but there is no theoretically guarantee that this is the case
Sh[illegible]xPlanations (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 [R19]

Reduced Affine Form (RAF) Abstraction

$$\begin{tikzcd} \mathbb{R}^n \arrow[rr, "SVM(x)"] \arrow[rrrr, dotted, bend right, "C_{SVM}(x)"'] & & \mathbb{R} \arrow[rr, dashed, "sgn(SVM(x))"] & & \{ \bullet, \blacktriangle \} \end{tikzcd}$$



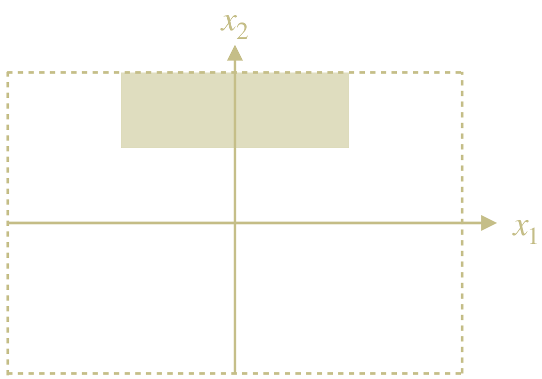
$$\begin{tikzcd} (RAF_n)^n \arrow[rr, "SVM^\sharp(x^\sharp)"] \arrow[rrrr, dotted, bend right] & & RAF_n \arrow[rr, dashed] & & \{ \bullet, \blacktriangle \} \end{tikzcd}$$

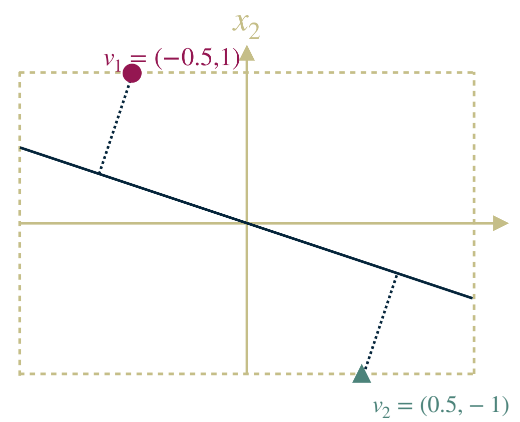
$$RAF_n \stackrel{\text{def}}{=} \left\{ a_0 + \sum_{i=1}^{n} a_i\epsilon_i + a_r\epsilon_r \mid a_0, a_1, \ldots a_n \in \mathbb{R}, a_r \in \mathbb{R}_{\geq 0} \right\} \cup \{ \top_{RAF} \}$$

Abstract Interpretation of SVMs

Example

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





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

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

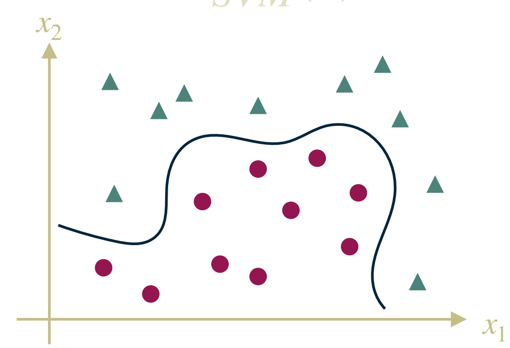
$-0.75 - 0.25\epsilon_1 + 0.25\epsilon_2$

$$SVM^\sharp((-0.5\epsilon_1,\ 0.75 - 0.25\epsilon_2))$$
$$= -0.5(-0.5(-0.5\epsilon_1)+1(0.75 - 0.25\epsilon_2))+0.5(0.5(-0.5\epsilon_1)-1(0.75 - 0.25\epsilon_2))$$
$$= -0.5(0.75 + 0.25\epsilon_1 - 0.25\epsilon_2))+0.5(-0.75 - 0.25\epsilon_1 + 0.25\epsilon_2))$$
$$= -0.75 - 0.25\epsilon_1 + 0.25\epsilon_2$$

Abstract Feature Importance [Pal2024]

$$\mathbb{R}^n \xrightarrow{SVM(x)} \mathbb{R} \xrightarrow{sgn(SVM(x))} \{●, ▲\}$$

$$C_{SVM}(x)$$



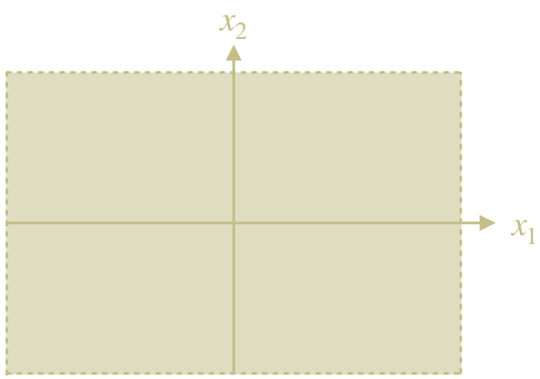
$$(RAF_n)^n \xrightarrow{SVM^\sharp(x^\sharp)} RAF_n \dashrightarrow \{●, ▲\}$$

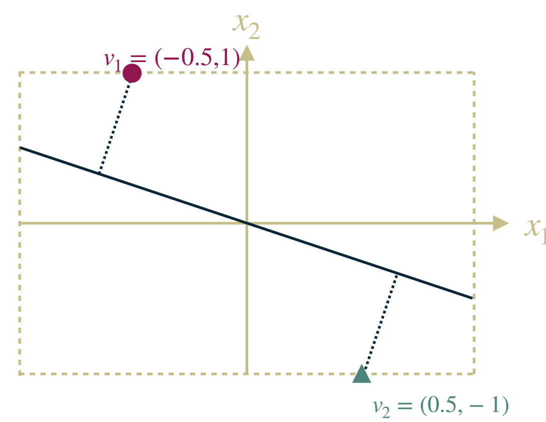
$$RAF_n \stackrel{\text{def}}{=} \left\{ a_0 + \sum_{i=1}^{n} a_i \epsilon_i + a_r \epsilon_r \mid a_0, a_1, \ldots a_n \in \mathbb{R}, a_r \in \mathbb{R}_{\geq 0} \right\} \cup \{ \top_{RAF} \}$$

Abstract Feature Importance [Pal2024]

Example

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





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

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

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

AFI vs PFI

German Dataset

		Grade for each feature										
Linear	Baseline (13.55s)	**5**	**5**	**5**	**6**	**6**	**7**	**7**	**7**	**7**	**8**	Distance
	AFI (0.01s)	**5**	**5**	**5**	**6**	**6**	**7**	8	**7**	**7**	**8**	**1.0**
	PFI (4.07s)	**5**	**5**	6	7	7	9	6	6	**7**	7	3.16
RBF	Baseline (17.98s)	**5**	**5**	**5**	**6**	**6**	**7**	**7**	**7**	**8**	**8**	Distance
	AFI (0.02s)	**5**	6	**5**	**6**	**6**	8	**7**	**7**	**8**	7	**1.73**
	PFI (6.23s)	6	7	**5**	**6**	7	8	**7**	6	7	5	4.24
Polynomial	Baseline (15.83s)	**5**	**5**	**5**	**6**	**7**	**7**	**7**	**7**	**7**	**8**	Distance
	AFI (0.01s)	7	6	7	7	5	**7**	6	6	5	**8**	**4.47**
	PFI (4.15s)	6	7	9	7	6	**7**	5	6	6	6	5.74

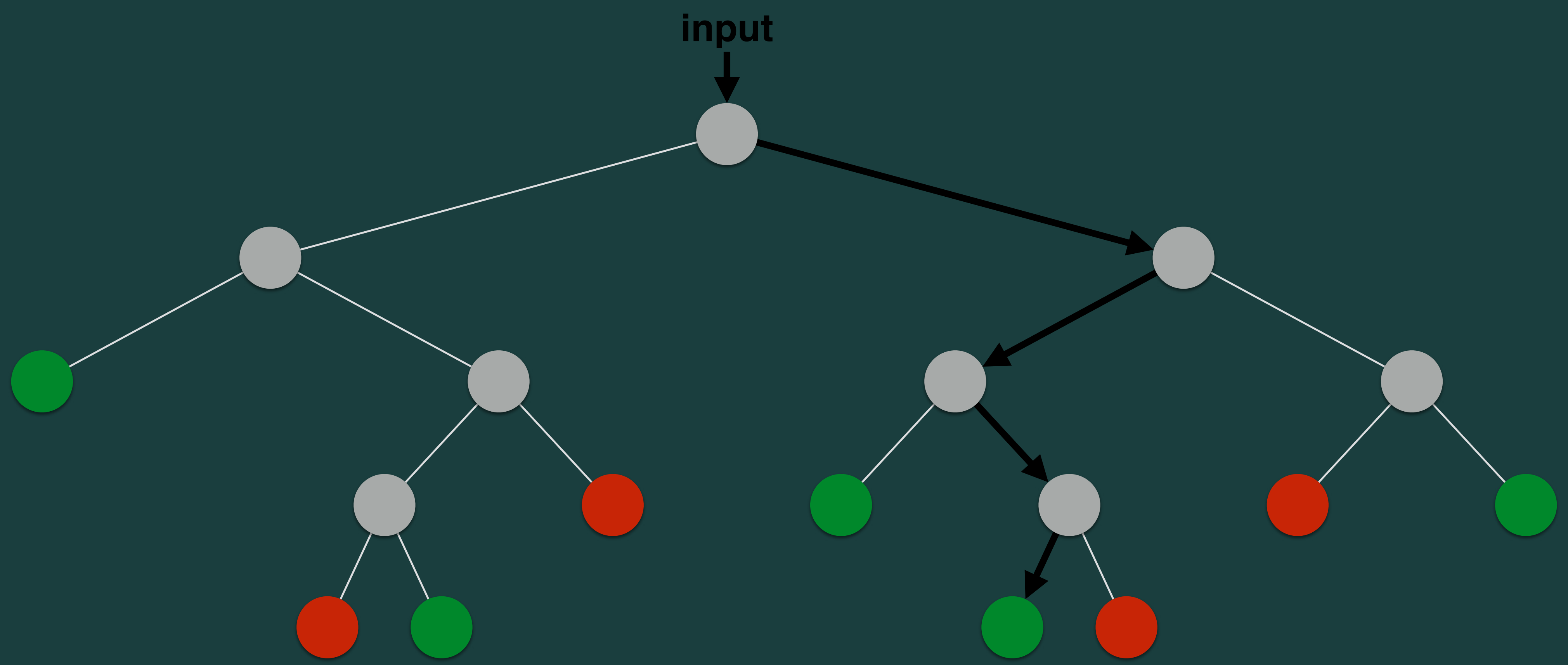
AFI vs PFI

	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

Decision Tree Ensembles



- **A. Kantchelian, J. D. Tygar, and A. Joseph**. *Evasion and Hardening of Tree Ensemble Classifiers*. In ICML 2016.
  H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh. *Robustness Verification of Tree-based Models*. In NeurIPS 2019.
  approaches for finding the nearest adversarial example

Decision Tree Ensembles

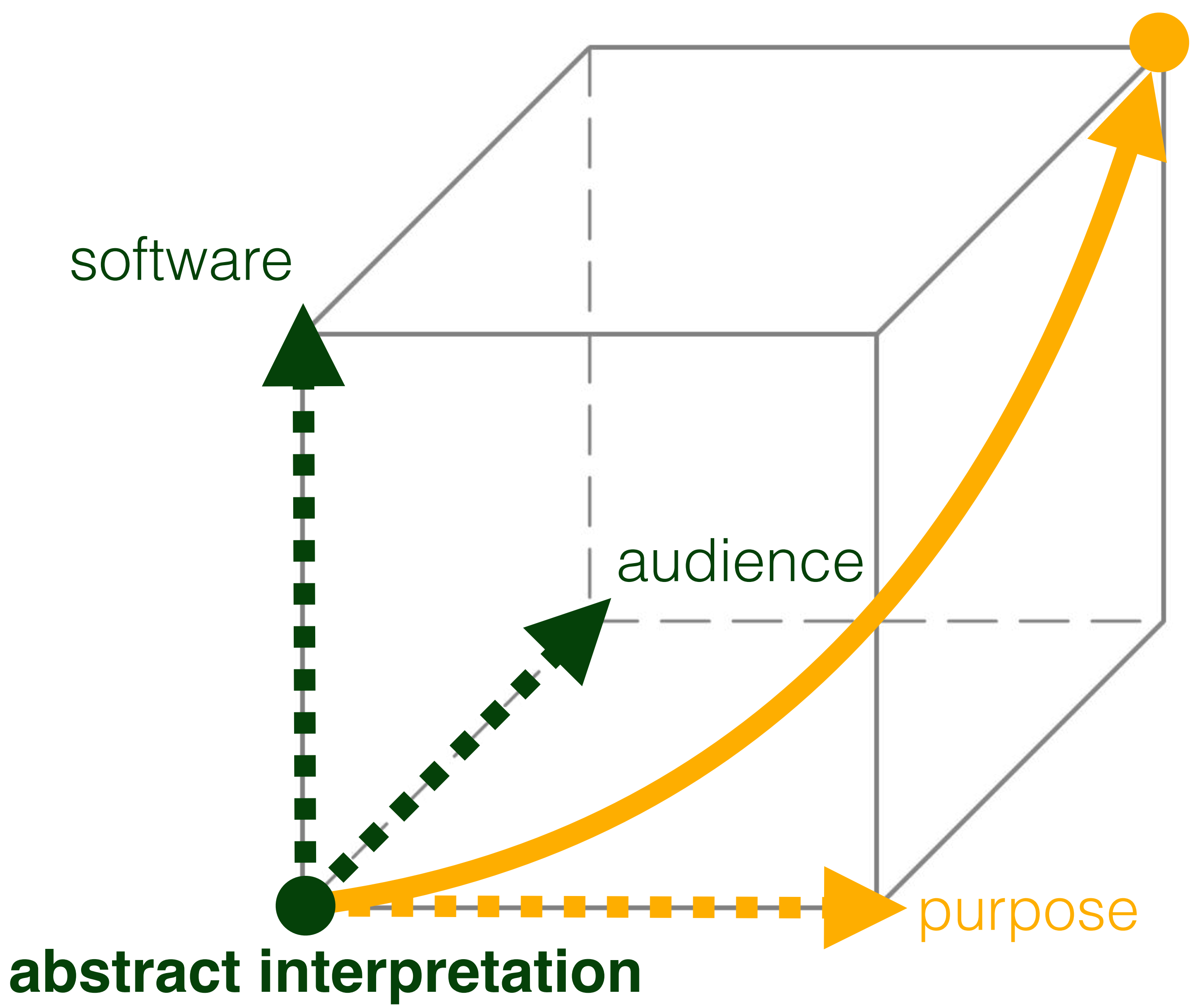
- **N. Sato, H. Kuruma, Y. Nakagawa, and H. Ogawa**. *Formal Verification of Decision-Tree Ensemble Model and Detection of its Violating-Input-Value Ranges*. 2020.
  approach for safety verification

- **G. Einziger, M. Goldstein, Y. Sa'ar, and I. Segall.** *Verifying Robustness of Gradient Boosted Models*. In AAAI 2019.
  SMT-based approach for local robustness

- **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 robustness

Formal Methods for Model Training

Formal Methods for Training



Robust Training

Minimizing the Worst-Case Loss for Each Input

loss function (e.g, cross-entropy)

expected value

$$\min_{\boldsymbol{\theta}} \ \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \left[\max_{\mathbf{x}'\in\mathcal{C}(\mathbf{x})} \mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}'), \mathbf{y}) \right]$$

model parameters

data label

model

data point

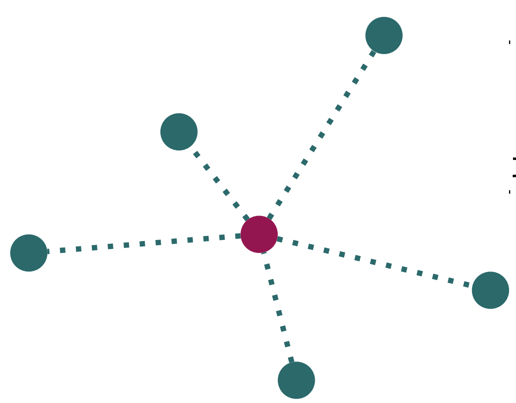
perturbation domain

Robust Training

Minimizing the Worst-Case Loss for Each Input

Adversarial Training

Minimizing a **Lower Bound on the Worst-Case Loss** for Each Input



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

$$\geq$$

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

generate adversarial inputs and use them as training data

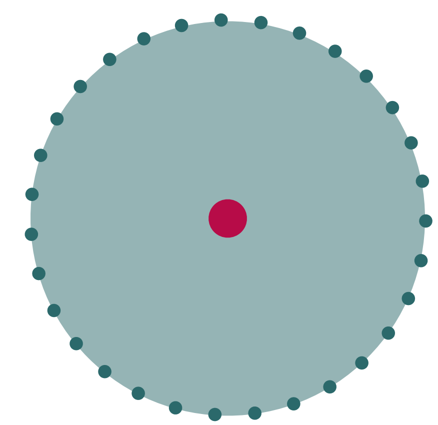
Certified Training

Minimizing an **Upper Bound on the Worst-Case Loss** for Each Input

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

$$\leq$$

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



use upper bound as regularizer to encourage robustness

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

Certified Training

- **M. Mirman, T. Gehr, and M. Vechev**. *Differentiable Abstract Interpretation for Provably Robust Neural Networks* In ICML 2018.
  abstract interpretation-based approach targeting neural networks

- **F. Ranzato and M. Zanella**. *Genetic Adversarial Training of Decision Trees*. In GECCO 2021.
  abstract interpretation-based approach targeting decision trees

Certified 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

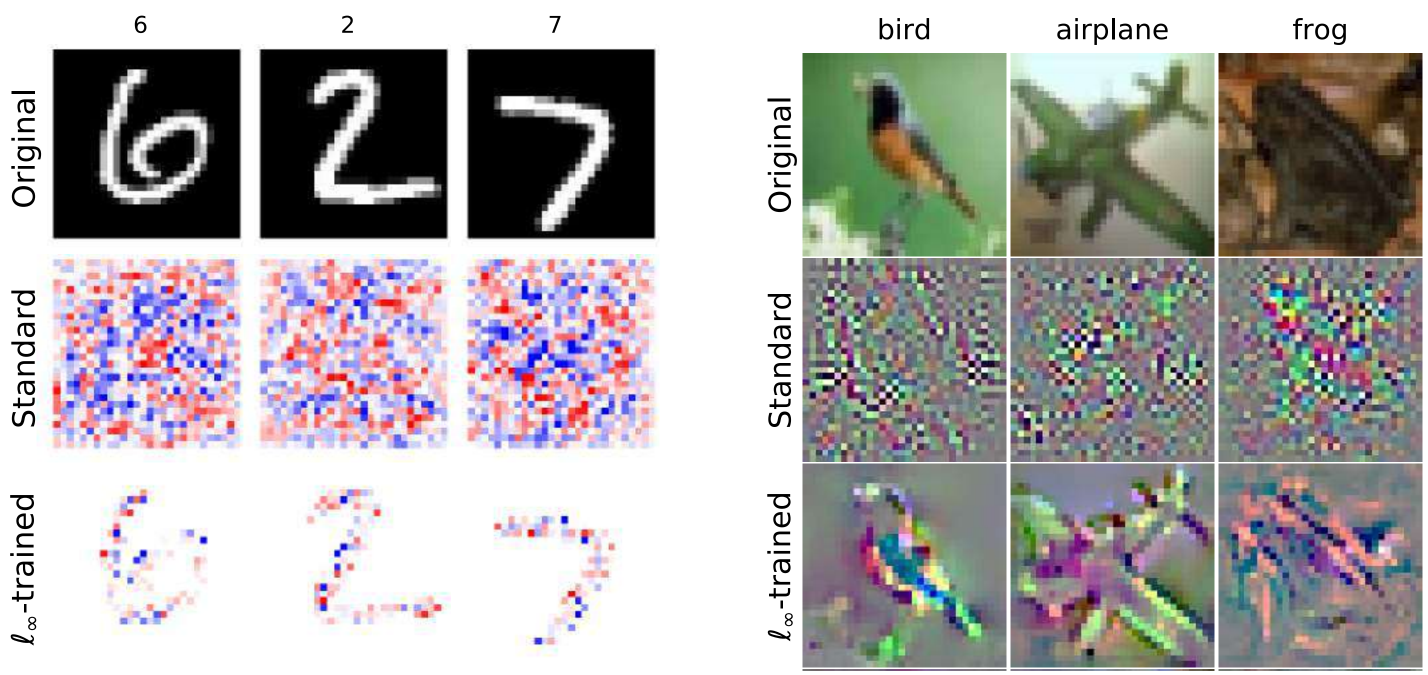
Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, untargeted attack

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
1	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

Robust Training

Perceptually Aligned Gradients



Adversarial Training

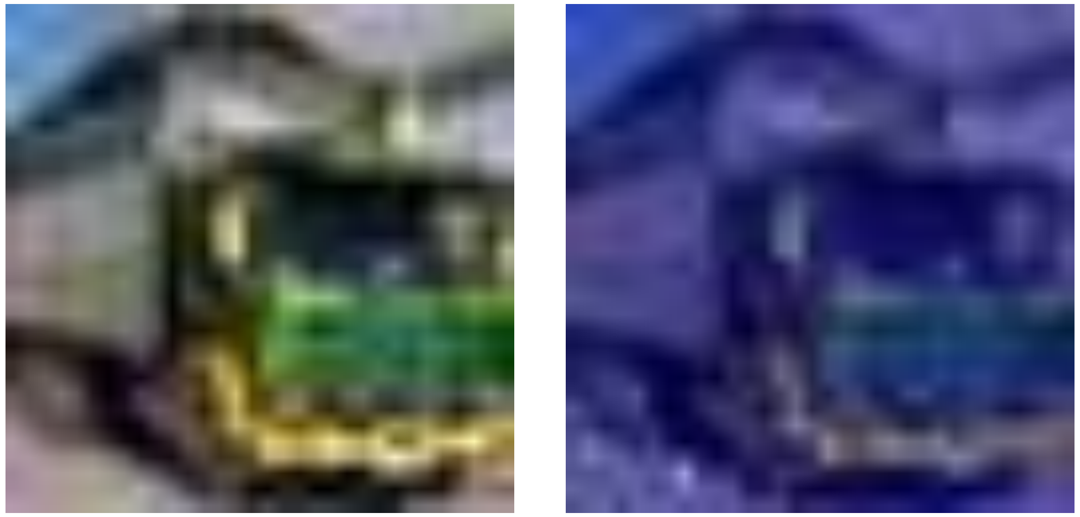


Fig. 6. Input Image   Fig. 7. Integrated Gradients

Certified Training

Robust Training

Minimizing the Worst-Case Loss for Each Input

Adversarial Training

Minimizing a Lower Bound on the Worst-Case Loss for Each Input

Certified Training

Minimizing an Upper Bound on the Worst-Case Loss for Each Input

Hybrid Training

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

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**

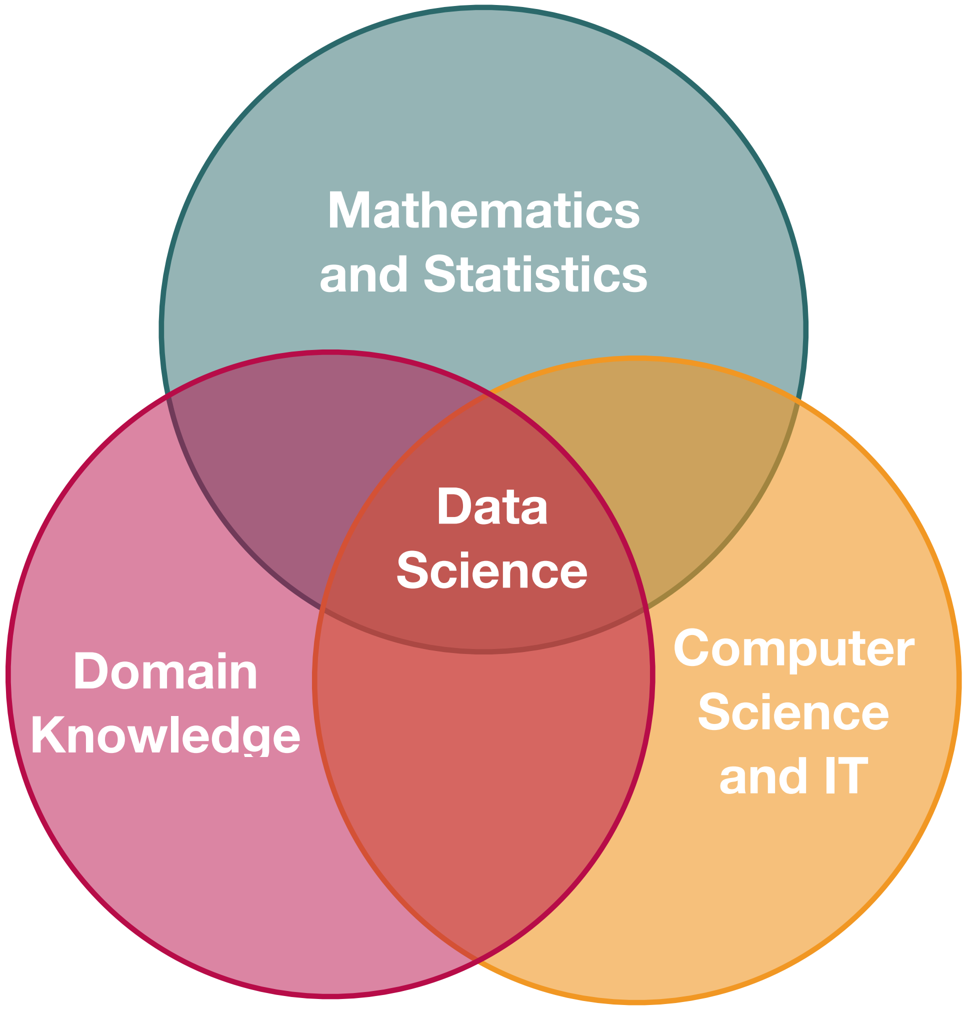
Hybrid Training

- **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

Formal Methods for Data Preparation

Data Scientists



https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

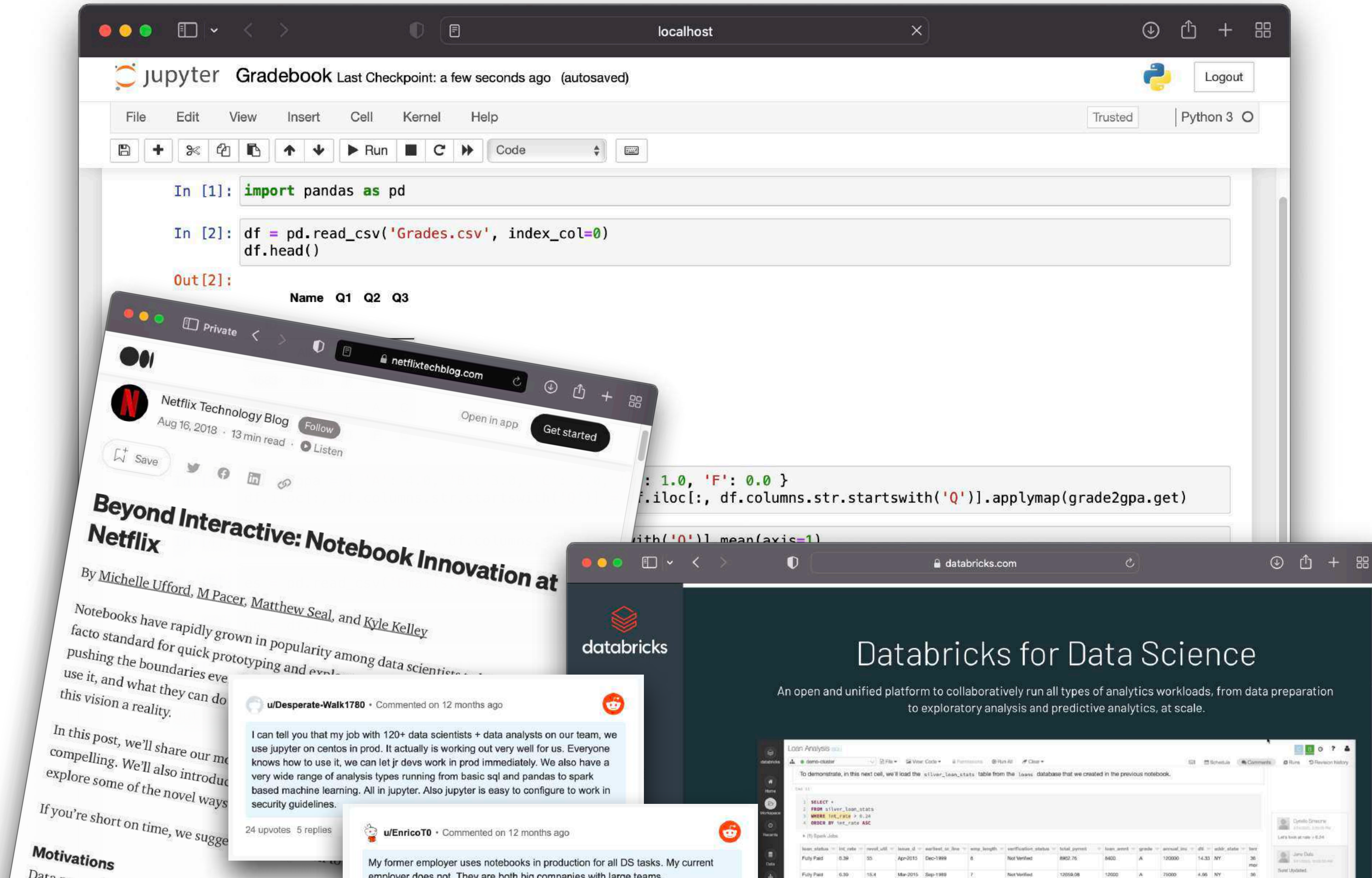
Data Scientist: The Sexiest Job of the 21st Century

Andrew McAfee and Erik Brynjolfsson

Andrew J Buboltz, silk screen on a page from a high school yearbook, 8.5" x 12", 2011 Tamar Cohen

When Jonathan Goldman arrived for work in June 2006 at [LinkedIn, the business networking site], the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know

Jupyter Notebooks



Jupyter Notebooks

P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Jupyter Notebooks

P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Jupyter Notebooks

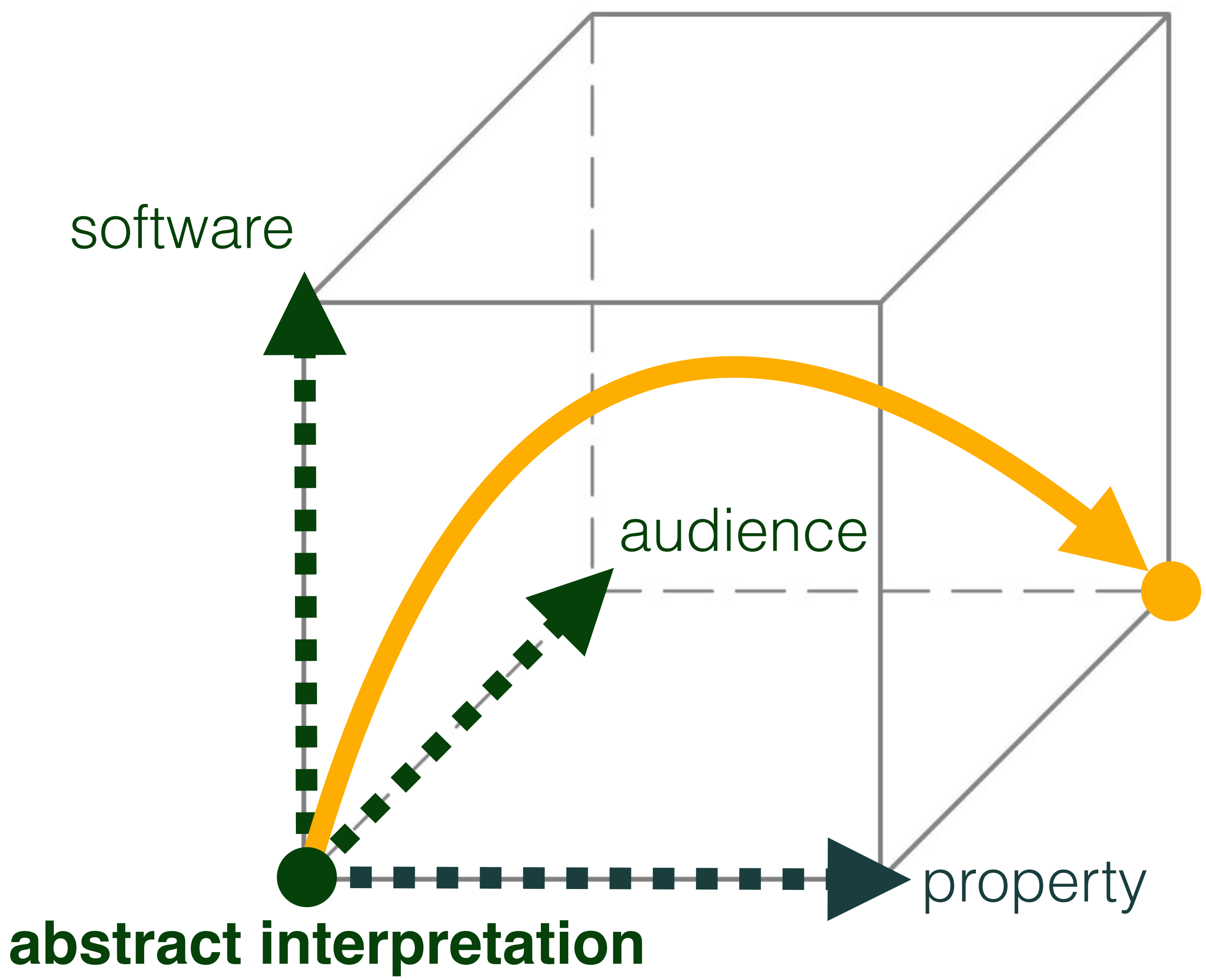
```
[1] d = genfromtxt('data.csv')

[2] selector = SelectKBest(k=25)
    x = selector.fit_transform(d)

[3] x = genfromtxt('data2.csv')

[4] x_train, x_test, y_train, y_test =
      train_test_split(x, ...)

[5] lr = LogisticRegression()
    lr.fit(x_train, y_train)
    y_pred = lr.predict(x_test)
```

STALE DATA

P. Subotić et al. - A Static Analysis Framework for Data Science Notebooks (ICSE 2022)

Anomalously Unused Data

(Un)used Data Analysis



The Reinhart-Rogoff Paper

FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy | April 18, 2013

The Excel Depression

By PAUL KRUGMAN
Published: April 18, 2013 | 470 Comments

FACEBOOK
TWITTER
GOOGLE+
SAVE
EMAIL
SHARE
PRINT
REPRINTS

In this age of information, math errors can lead to disaster. NASA's Mars Orbiter crashed because engineers forgot to convert to metric measurements; JPMorgan Chase's "London Whale" venture went bad in part because modelers divided by a sum instead of an average. So, did an Excel coding error destroy the economies of the Western world?

Enlarge This Image

The story so far: At the beginning of 2010, two Harvard economists, Carmen Reinhart and Kenneth Rogoff, circulated a paper, "Growth in a Time of Debt," that purported to identify a critical "threshold," a tipping point, for government indebtedness. Once debt exceeds 90 percent of gross domestic product, they claimed, economic growth drops off sharply.

Ms. Reinhart and Mr. Rogoff had credibility thanks to a

England Covid-19 Cases Error

SCIENCE | US & WORLD | TECH

Excel spreadsheet error blamed for UK's 16,000 missing coronavirus cases

The case went missing after the spreadsheet hit its filesize limit

By James Vincent | Oct 5, 2020, 9:41am EDT

NEWS

Covid-19: Only half of 16 000 patients missed from England's official figures have been contacted

Check for updates

The BMJ

Cite this as: *BMJ* 2020;371:m3891

http://dx.doi.org/10.1136/bmj.m3891

Published: 06 October 2020

Elisabeth Mahase

Details of nearly 16 000 cases of covid-19 were not transferred to England's NHS Test and Trace service and were missed from official figures because of an error in the process for updating the data.

England's health and social care secretary, Matt Hancock, told the House of Commons on Monday 5 October that after the error was discovered on Friday 2 October "6500 hours of extra contact tracing" had been carried out over the weekend. But as at Monday morning only half (51%) of the people had been reached by contact tracers.

In response, Labour's shadow health secretary, ... said, "Thousands of people are ... exposed to covid, ...

data and furthermore have issued guidance on validation and risk management for these products if they are to be used in such a safety critical manner."

The error came as the Labour Party's leader, Keir Starmer, said that the prime minister had "lost control" of covid-19, with no clear strategy for beating it. Speaking to the *Observer*, Starmer set out his five point plan for covid-19, which starts with publishing the criteria for local restrictions, as the German government did. Secondly, he said public health messaging should be improved by adding a feature to the NHS covid-19 app so people can search their postcode and find out their local restrictions.

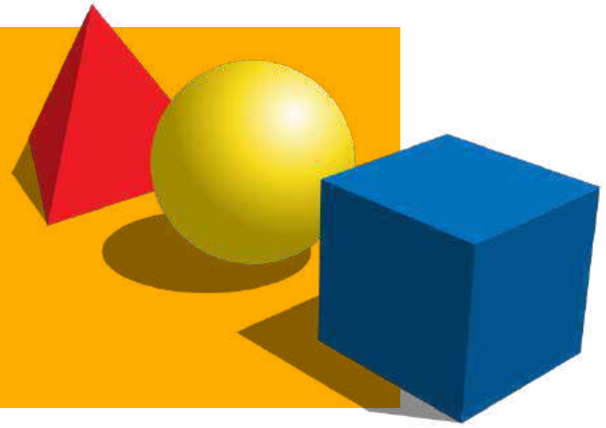
Starmer has also said he would fix the contact tracing system by investing in NHS and university ... to expand testing and at the same time ... teams in charge of contact ... in high

BMJ: first published as 10.1136/bmj.m3891 on 6 October 2020. Do

Data Usage Static Analysis [CU18]

practical tools
targeting specific programs

algorithmic approaches
to decide program properties



mathematical models
of the program behavior

Data (Non-)Usage

$$\mathcal{N}_J \stackrel{\text{def}}{=} \{[\![P]\!] \mid \text{UNUSED}_J([\![P]\!])\}$$

$\mathcal{N}_J$ is the set of all programs P (or, rather, their semantics $[\![P]\!]$) that **do not use** the value of the input variables in J

$$\begin{aligned}\text{UNUSED}_J([\![P]\!]) \stackrel{\text{def}}{=} \; & \forall t \in [\![P]\!], V \in \mathcal{R}^{|J|} \colon t_0(J) \neq V \Rightarrow \exists t' \in [\![P]\!] \colon \\ & (\forall i \colon i \notin J \Rightarrow t_0(i) = t'_0(i)) \\ & \wedge t'_0(J) = V \\ & \wedge t_\omega = t'_\omega\end{aligned}$$

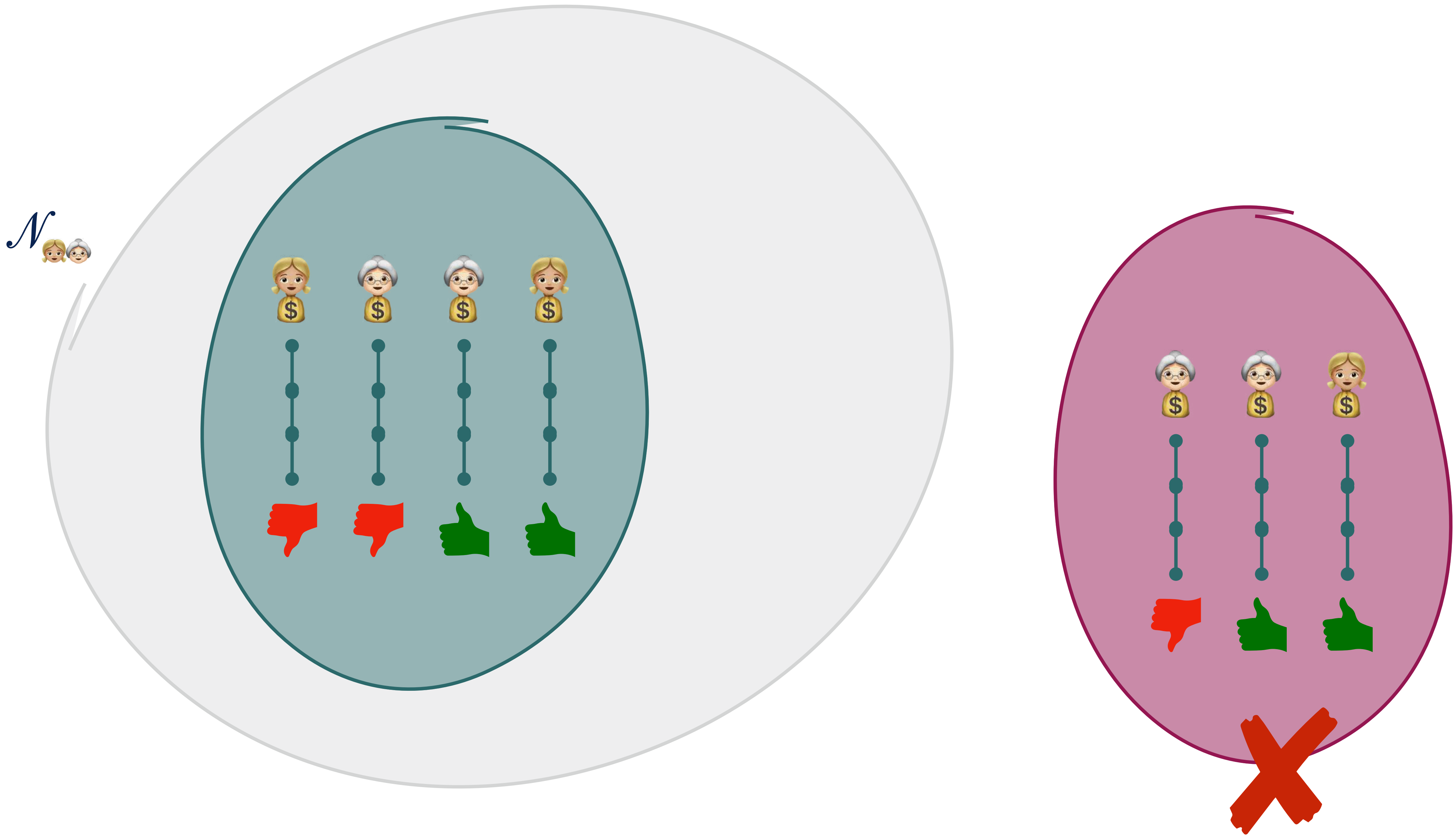
Intuitively: **any possible program outcome** is possible **from any value of the input variable** i

Theorem
$P \models \mathcal{N}_J \Leftrightarrow \{[\![P]\!]\} \subseteq \mathcal{N}_J$

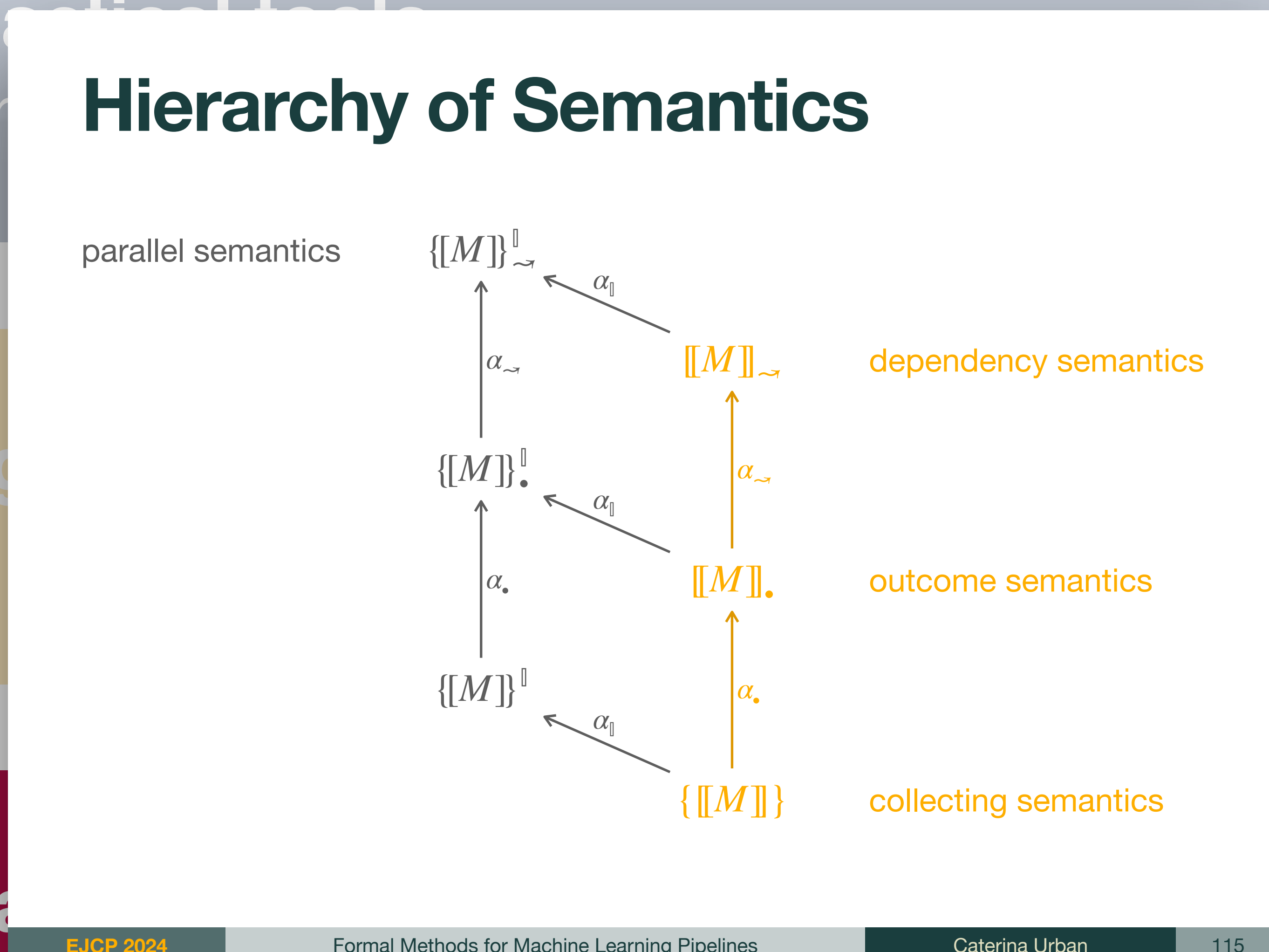
Corollary
$P \models \mathcal{N}_J$ … $\subseteq [\![P]\!]^\natural \in \mathcal{N}_J$

Data (Non-) Usage

Not a Subset-Closed Property



Data Usage Static Analysis [CU18]

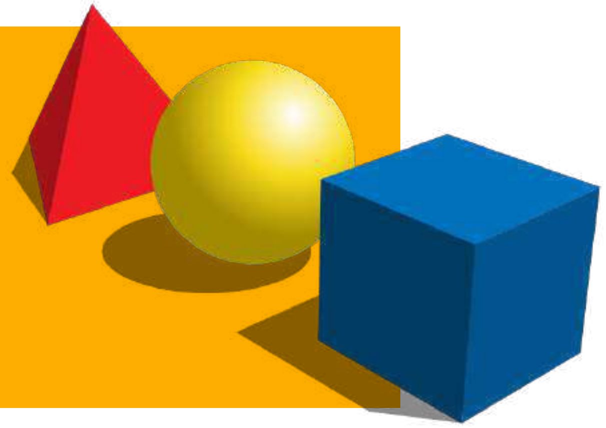


of the program behavior

Data Usage Static Analysis [CU18]

practical tools
targeting specific programs

algorithmic approaches
to decide program properties



mathematical models
of the program behavior

Data (Non-)Usage Abstractions

Over-Approximation of the Used Input Data

⇒ Under-Approximation of the Unused Input Data

$$P \models \mathcal{N}_{J^\natural \subseteq J} \Leftarrow [\![P]\!] \subseteq [\![P]\!]^\natural_A \subseteq \mathcal{N}_{J^\natural \subseteq J}$$

Example

```
english = bool(input())
math = bool(input())
science = bool(input())
bonus = bool(input())

passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus

print(passing)
```

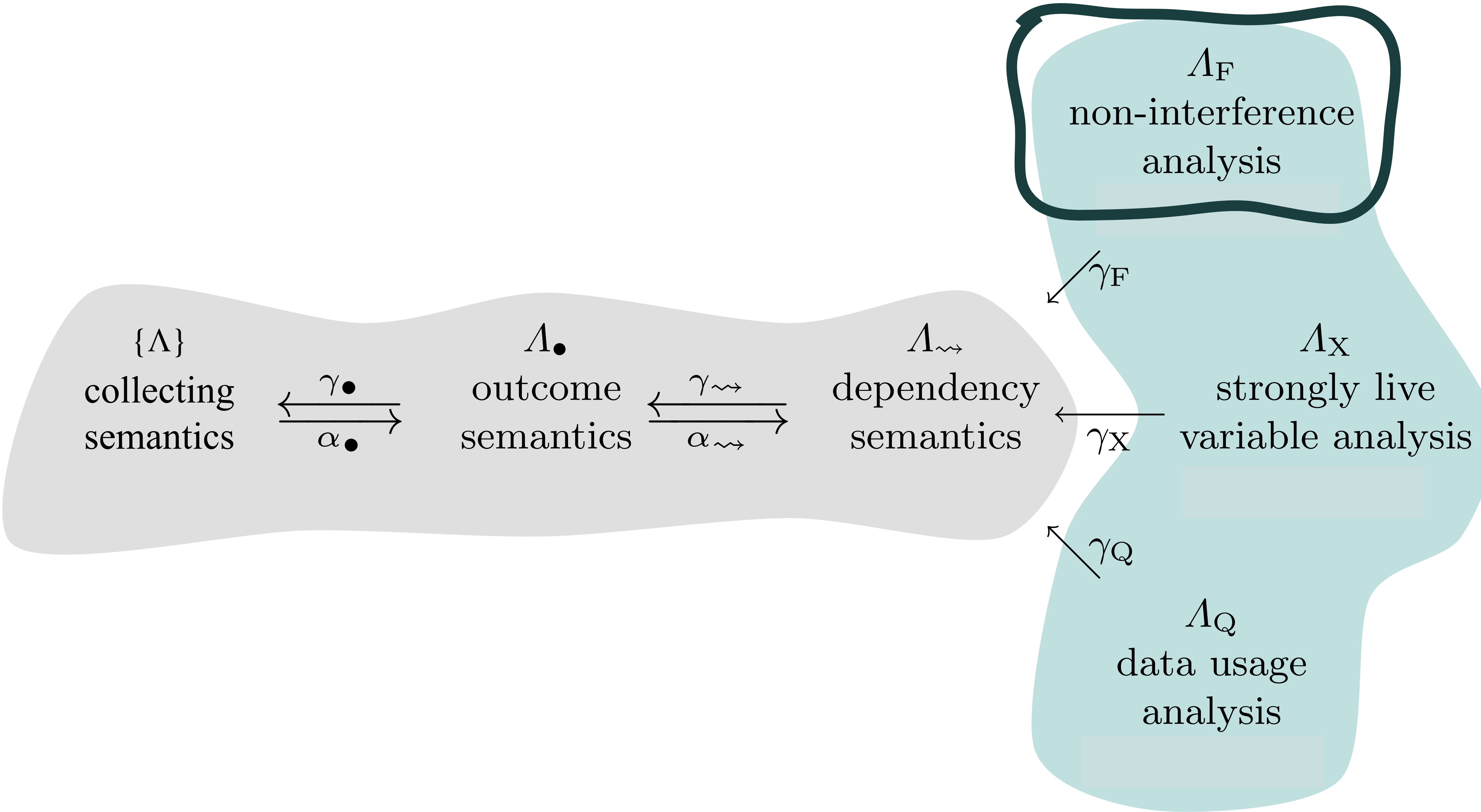
INPUT VARIABLES

ERROR: english SHOULD BE passing

ERROR: math SHOULD BE science

OUTPUT VARIABLES

the input variables english and science are unused



Λ_F
non-interference analysis
γ_F
$\{\Lambda\}$
collecting semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome semantics
$\gamma_\rightsquigarrow$
$\alpha_\rightsquigarrow$
$\Lambda_\rightsquigarrow$
dependency semantics
γ_X
Λ_X
strongly live variable analysis
γ_Q
Λ_Q
data usage analysis

Secure Information Flow

L ⇢ x
L ⇢ y
H ⇢ t
L ⇢ z
H ⇢ w

possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

$[\![P]\!]_{\mathrm{F}}$

$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e$ (expressions)

$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s$ (statements)

explicit usage flows

implicit usage flows

$$\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$

$$\Theta_{\mathrm{F}}[\![x = e]\!](S) \stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}$$

$$\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S \\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{w}(s_1) \cup \mathrm{w}(s_2)\} & \text{otherwise} \end{cases}$$

$$\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \mathrm{lfp}_S^{\sqsubseteq_{\mathrm{F}}}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$

$$\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)$$

S guarantees a unique value fore independently of values of input variables

$\mathcal{V}_{\mathrm{F}}[\![x]\!]S \Leftrightarrow L \rightsquigarrow x \in S$

set of variables modified by s_i

$\mathcal{L} \stackrel{\text{def}}{=} \{L, H\}$: set of security levels

$L \rightsquigarrow x$: dependency constraint

$\mathrm{F} \stackrel{\text{def}}{=} \{L \rightsquigarrow x \mid x \in X\}$

$\langle \mathcal{P}(\mathrm{F}), \sqsubseteq_{\mathrm{F}}, \sqcup_{\mathrm{F}} \rangle$: abstract domain

$S_1 \sqsubseteq_{\mathrm{F}} S_2 \stackrel{\text{def}}{=} S_1 \supseteq S_2$

$S_1 \sqcup_{\mathrm{F}} S_2 \stackrel{\text{def}}{=} S_1 \cap S_2$

Hypercollecting Semantics and Its Application to Static Analysis of Information Flow

Mounir Assaf
Stevens Institute of Technology, Hoboken, US
first.last@stevens.edu

David A. Naumann
Stevens Institute of Technology, Hoboken, US
first.last@stevens.edu

Julien Signoles
Software Reliability and Security Lab, CEA LIST, Saclay, FR
first.last@cea.fr

Éric Totel
CIDRE, CentraleSupélec, Rennes, FR
first.last@centralesupelec.fr

Frédéric Tronel
CIDRE, CentraleSupélec, Rennes, FR
first.last@centralesupelec.fr

program is correct if all its traces satisfy the predicate. By c… with such *trace properties*, extensional definitions of depe… involve more than one trace. To express that the final va… depend only on the initial value of a variab… *interference* in the security…

Secure Information Flow

L ⇢ x
L ⇢ y
H ⇢ t
L ⇢ z
H ⇢ w

possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

$[\![P]\!]_{\mathrm{F}}$

$$
\begin{aligned}
e &::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e && \text{(expressions)}\\
s &::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s && \text{(statements)}
\end{aligned}
$$

$$
\begin{aligned}
\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) &\stackrel{\text{def}}{=} S\\
\Theta_{\mathrm{F}}[\![x = e]\!](S) &\stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}\\
\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) &\stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S\\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{W}(s_1) \cup \mathrm{W}(s_2)\} & \text{otherwise} \end{cases}\\
\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) &\stackrel{\text{def}}{=} \mathrm{lfp}_S^{\sqsubseteq_{\mathrm{F}}}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]\\
\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) &\stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)
\end{aligned}
$$

```
passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus
H ⇢ english, math, science, bonus, passing
H ⇢ english, math, science, bonus, passing

Secure Information Flow

L ⇢ x
L ⇢ y
H ⇢ t
L ⇢ z
H ⇢ w

possibilistic non-interference coincides with input data (non-)usage when the set J of unused input variables contains *all* input variables:

- **input variables are high-security variables**
- **output variables are low-security variables**

and the program is terminating

$[\![P]\!]_{\mathrm{F}}$

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

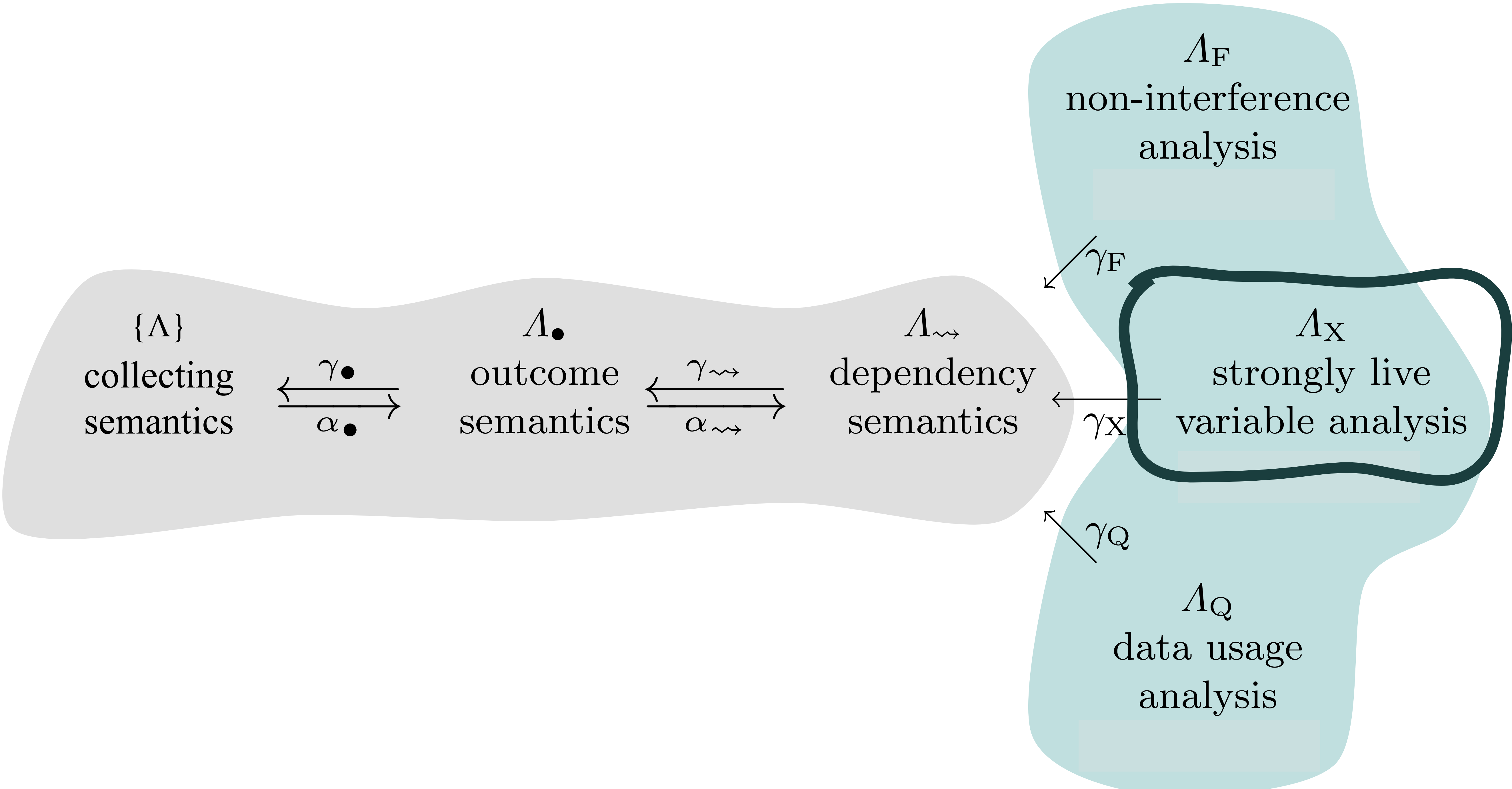
$$\Theta_{\mathrm{F}}[\![\texttt{skip}]\!](S) \stackrel{\text{def}}{=} S$$
$$\Theta_{\mathrm{F}}[\![x = e]\!](S) \stackrel{\text{def}}{=} \{L \rightsquigarrow y \in S \mid y \neq x\} \cup \{L \rightsquigarrow x \mid \mathcal{V}_{\mathrm{F}}[\![e]\!]S\}$$
$$\Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](S) \stackrel{\text{def}}{=} \begin{cases} \Theta_{\mathrm{F}}[\![s_1]\!](S) \sqcup_{\mathrm{F}} \Theta_{\mathrm{F}}[\![s_2]\!](S) & \text{if } \mathcal{V}_{\mathrm{F}}[\![e]\!]S \\ \{L \rightsquigarrow x \in S \mid x \notin \mathrm{W}(s_1) \cup \mathrm{W}(s_2)\} & \text{otherwise} \end{cases}$$
$$\Theta_{\mathrm{F}}[\![\texttt{while}\ e\texttt{:}\ s]\!](S) \stackrel{\text{def}}{=} \mathrm{lfp}_S^{\sqsubseteq_{\mathrm{F}}}\ \Theta_{\mathrm{F}}[\![\texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!]$$
$$\Theta_{\mathrm{F}}[\![s_1\ s_2]\!](S) \stackrel{\text{def}}{=} \Theta_{\mathrm{F}}[\![s_2]\!] \circ \Theta_{\mathrm{F}}[\![s_1]\!](S)$$

```
passing = True
while not english:
    english = False
```

L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus
L ⇢ passing, H ⇢ english, math, science, bonus

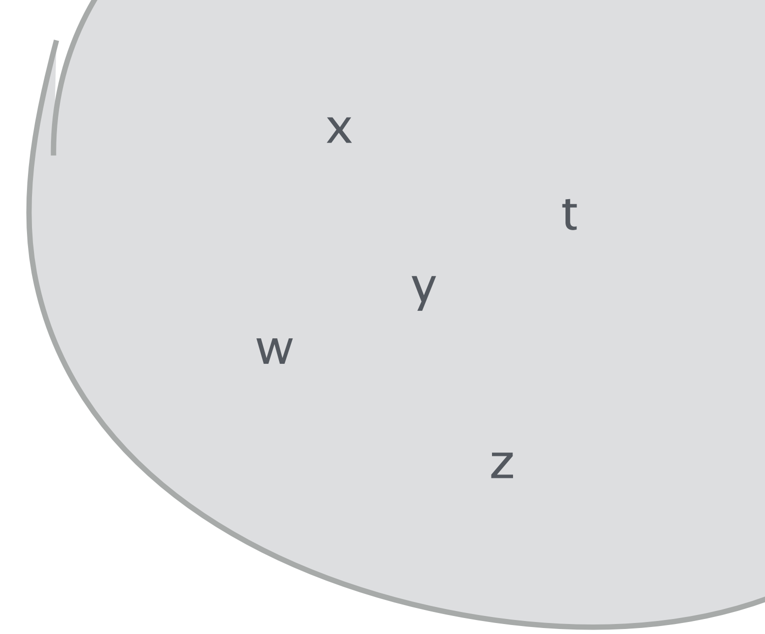
Theorem

$$P \models \mathcal{N}_J^{+} \Leftarrow [\![P]\!] \subseteq [\![P]\!]_F^{\natural} \subseteq \mathcal{N}_J^{+}$$



$\{\Lambda\}$
collecting semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome semantics
$\gamma_\rightsquigarrow$
$\alpha_\rightsquigarrow$
$\Lambda_\rightsquigarrow$
dependency semantics
Λ_F
non-interference analysis
γ_F
γ_X
Λ_X
strongly live variable analysis
γ_Q
Λ_Q
data usage analysis

Strong-Liveness



a variable is strongly live if

- **it is used in an assignment to another strongly live variable**
- **it is used in a statement other than an assignment**

$\llbracket P \rrbracket_{\mathrm{X}}$

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$

$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\Theta_{\mathrm{X}}\llbracket \texttt{skip} \rrbracket(S) \stackrel{\text{def}}{=} S$$

$$\Theta_{\mathrm{X}}\llbracket x = e \rrbracket(S) \stackrel{\text{def}}{=} \begin{cases} (S \setminus \{x\}) \cup \textsc{vars}(e) & x \in S \\ S & \text{otherwise} \end{cases}$$

$$\Theta_{\mathrm{X}}\llbracket \texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2 \rrbracket(S) \stackrel{\text{def}}{=} \textsc{vars}(b) \cup \Theta_{\mathrm{X}}\llbracket s_1 \rrbracket(S) \cup \Theta_{\mathrm{X}}\llbracket s_2 \rrbracket(S)$$

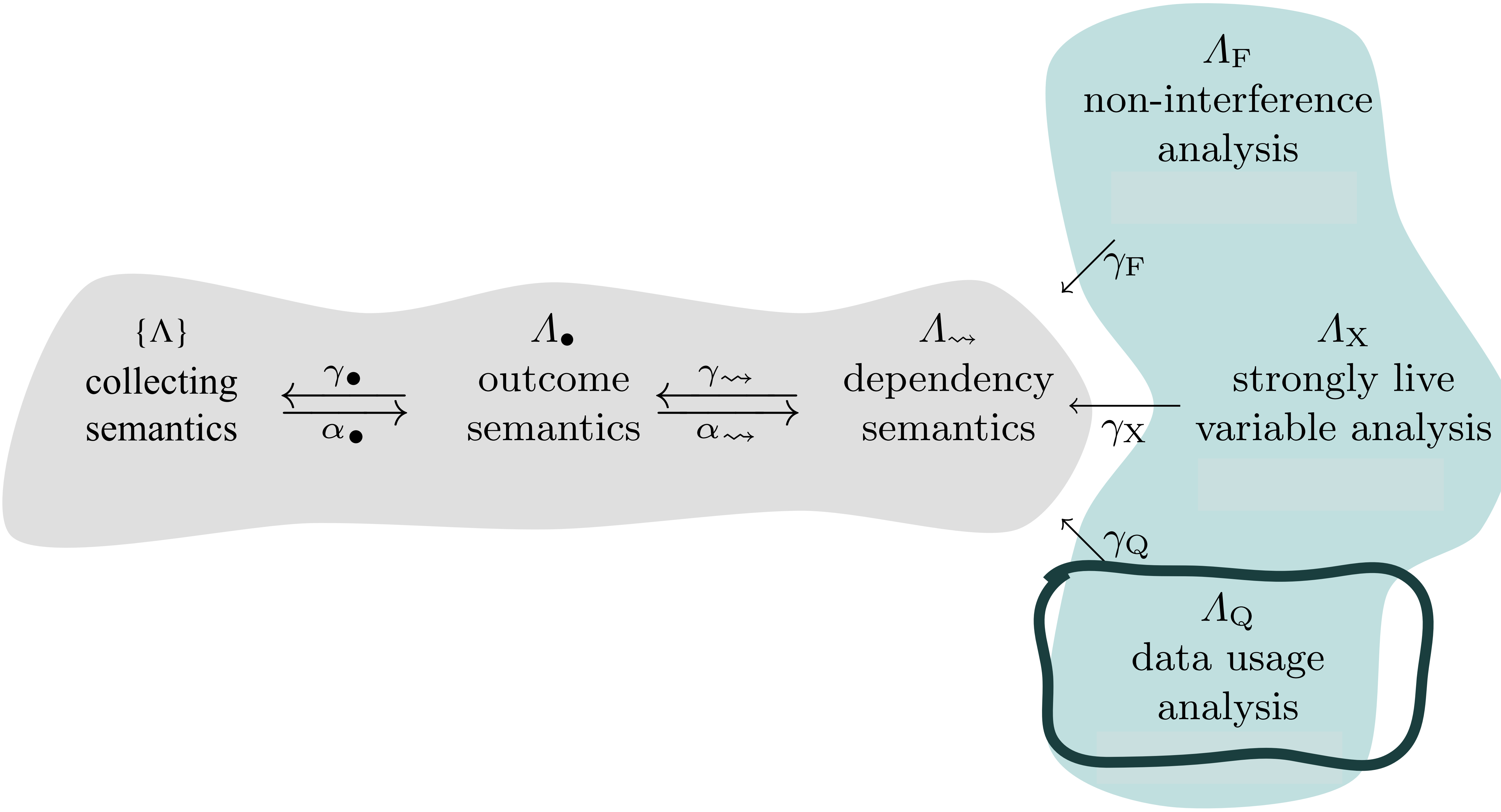
$$\Theta_{\mathrm{X}}\llbracket \texttt{while}\ b\texttt{:}\ s \rrbracket(S) \stackrel{\text{def}}{=} \textsc{vars}(b) \cup \Theta_{\mathrm{X}}\llbracket s \rrbracket(S)$$

$$\Theta_{\mathrm{X}}\llbracket s_1\ s_2 \rrbracket(S) \stackrel{\text{def}}{=} \Theta_{\mathrm{X}}\llbracket s_1 \rrbracket \circ \Theta_{\mathrm{X}}\llbracket s_2 \rrbracket(S)$$

Theorem

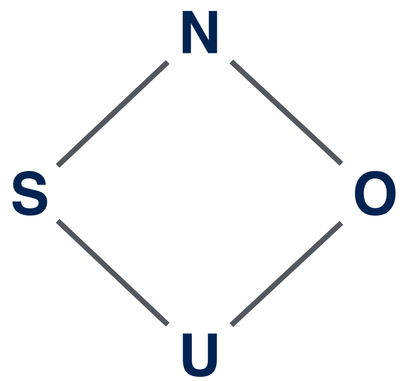
$$P \models \mathcal{N}_J \Leftarrow \llbracket P \rrbracket \subseteq \llbracket P \rrbracket^{\natural}_{X} \subseteq \mathcal{N}_J$$

Program	Strongly live variables
	{ bonus, math, english }
passing = True	{ bonus, math, english }
if not english:	{ bonus, math }
english = False	{ bonus, math }
if not math:	{ bonus, math }
passing = False or bonus	{ bonus, math }
if not math:	{ bonus }
passing = False or bonus	{ passing }


Λ_F
non-interference analysis
γ_F
$\{\Lambda\}$
collecting semantics
$\gamma_\bullet$
$\alpha_\bullet$
$\Lambda_\bullet$
outcome semantics
$\gamma_\rightsquigarrow$
$\alpha_\rightsquigarrow$
$\Lambda_\rightsquigarrow$
dependency semantics
γ_X
Λ_X
strongly live variable analysis
γ_Q
Λ_Q
data usage analysis

Syntactic (Non-)Usage

x ⇢ U
y ⇢ S | y ⇢ U
t ⇢ N
z ⇢ N
w ⇢ O | w ⇢ U



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

$[\![P]\!]_Q$

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q) \sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

$$\Theta_Q[\![\texttt{while}\ b\texttt{:}\ s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_t^{\sqsubseteq_Q}\ \Theta_Q[\![\texttt{if}\ b\texttt{:}\ s\ \texttt{else: skip}]\!]$$

$$\Theta_Q[\![s_1\ s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

```
passing = True

if not english:

    english = False
    ●  math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
●  math, bonus, passing ⇢ U
if not math:
    ●  math ⇢ S, bonus ⇢ U, passing ⇢ O | …
    passing = False or bonus
    ●  math, bonus, passing ⇢ S | math, bonus, passing ⇢ U
●  math, bonus, passing ⇢ U
if not math:
    ●  bonus ⇢ U, passing ⇢ O | passing ⇢ U
    passing = False or bonus
    ●  passing ⇢ S | passing ⇢ U
●  passing ⇢ U
```

Syntactic (Non-)Usage

x ⇢ U

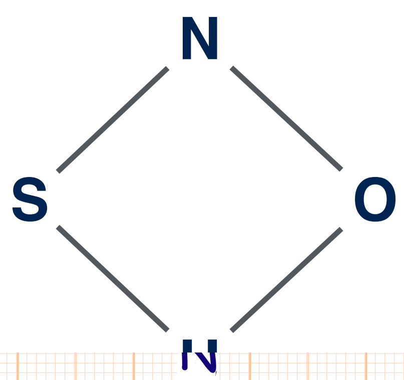
y ⇢ S | y ⇢ U

t ⇢ N

z ⇢ N

w ⇢ O | w ⇢ U

$[\![P]\!]_Q$



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

the input variables english and science are definetly not used by the program

```
passing = True
if not english:
    english = False
if not math:
    passing = False or bonus
if not math:
    passing = False or bonus
```

math, bonus ⇢ U, passing ⇢ O

math, bonus, passing ⇢ U

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

math, bonus, passing ⇢ S | math, bonus, passing ⇢ U

math, bonus, passing ⇢ U

passing ⇢ U

$$\Theta_Q[\![\texttt{skip}]\!](q) \stackrel{\text{def}}{=} q$$

$$\Theta_Q[\![x = e]\!](q) \stackrel{\text{def}}{=} \text{ASSIGN}[\![x = e]\!](q)$$

$$\Theta_Q[\![\texttt{if } b\texttt{: } s_1 \texttt{ else: } s_2]\!](q) \stackrel{\text{def}}{=} \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_1]\!] \circ \text{PUSH}(q)$$
$$\sqcup_Q \text{POP} \circ \text{FILTER}[\![b]\!] \circ \Theta_Q[\![s_2]\!] \circ \text{PUSH}(q)$$

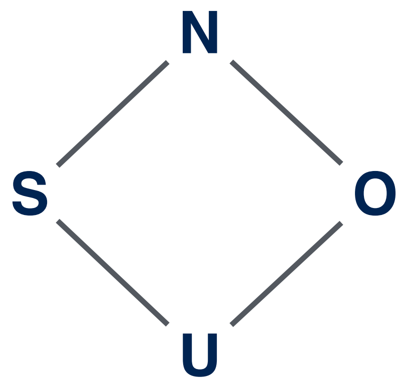
$$\Theta_Q[\![\texttt{while } b\texttt{: } s]\!](q) \stackrel{\text{def}}{=} \text{lfp}_t^{\sqsubseteq_Q} \Theta_Q[\![\texttt{if } b\texttt{: } s \texttt{ else: skip}]\!]$$

$$\Theta_Q[\![s_1 \; s_2]\!](q) \stackrel{\text{def}}{=} \Theta_Q[\![s_1]\!] \circ \Theta_Q[\![s_2]\!](q)$$

Syntactic (Non-)Usage

x ⇢ U
y ⇢ S | y ⇢ U
t ⇢ N
z ⇢ N
w ⇢ O | w ⇢ U

$[\![P]\!]_Q$



- **U**: used in the current scope (or an inner scope)
- **S**: used in an outer scope
- **O**: used in an outer scope and overridden in the current scope
- **N**: not used

$$e ::= v \mid x \mid \texttt{not}\ e \mid e\ \texttt{and}\ e \mid e\ \texttt{or}\ e \qquad \text{(expressions)}$$
$$s ::= \texttt{skip} \mid x = e \mid \texttt{if}\ e\texttt{:}\ s\ \texttt{else:}\ s \mid \texttt{while}\ e\texttt{:}\ s \mid s\ s \qquad \text{(statements)}$$

$$\begin{aligned}
\Theta_{\mathrm{Q}}[\![\texttt{skip}]\!](q) &\stackrel{\text{def}}{=} q \\
\Theta_{\mathrm{Q}}[\![x = e]\!](q) &\stackrel{\text{def}}{=} \mathrm{ASSIGN}[\![x = e]\!](q) \\
\Theta_{\mathrm{Q}}[\![\texttt{if}\ b\texttt{:}\ s_1\ \texttt{else:}\ s_2]\!](q) &\stackrel{\text{def}}{=} \mathrm{POP} \circ \mathrm{FILTER}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \mathrm{PUSH}(q) \\
&\qquad \sqcup_{\mathrm{Q}} \mathrm{POP} \circ \mathrm{FILTER}[\![b]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!] \circ \mathrm{PUSH}(q) \\
\Theta_{\mathrm{Q}}[\![\texttt{while}\ b\texttt{:}\ s]\!](q) &\stackrel{\text{def}}{=} \mathrm{lfp}_t^{\sqsubseteq_{\mathrm{Q}}}\ \Theta_{\mathrm{Q}}[\![\texttt{if}\ b\texttt{:}\ s\ \texttt{else:}\ \texttt{skip}]\!] \\
\Theta_{\mathrm{Q}}[\![s_1\ s_2]\!](q) &\stackrel{\text{def}}{=} \Theta_{\mathrm{Q}}[\![s_1]\!] \circ \Theta_{\mathrm{Q}}[\![s_2]\!](q)
\end{aligned}$$

```
passing = True
while not english:
    english = False
```

passing ⇢ O
passing ⇢ U

passing ⇢ U

Theorem

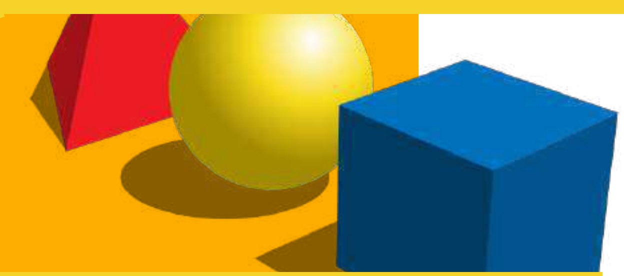
$$P \models \mathcal{N}_J^+ \Leftarrow [\![P]\!] \subseteq [\![P]\!]_Q^\natural \subseteq \mathcal{N}_J^+$$

Data Usage Static Analysis [CU18]

practical tools
targeting specific programs

secure information flow

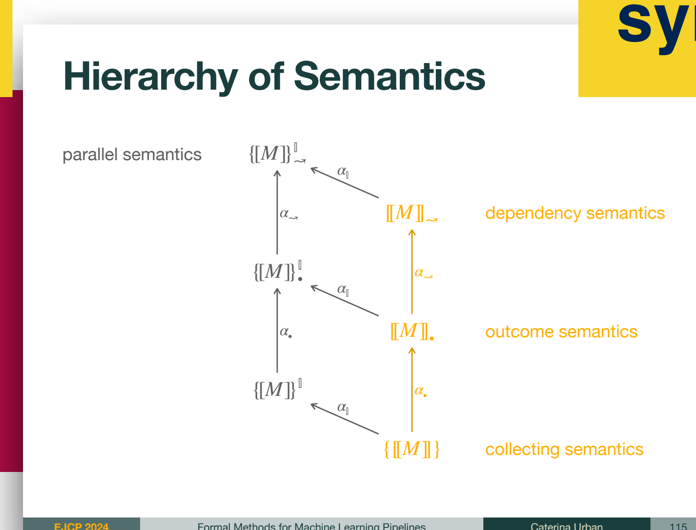
algorithmic approaches
to decide program properties

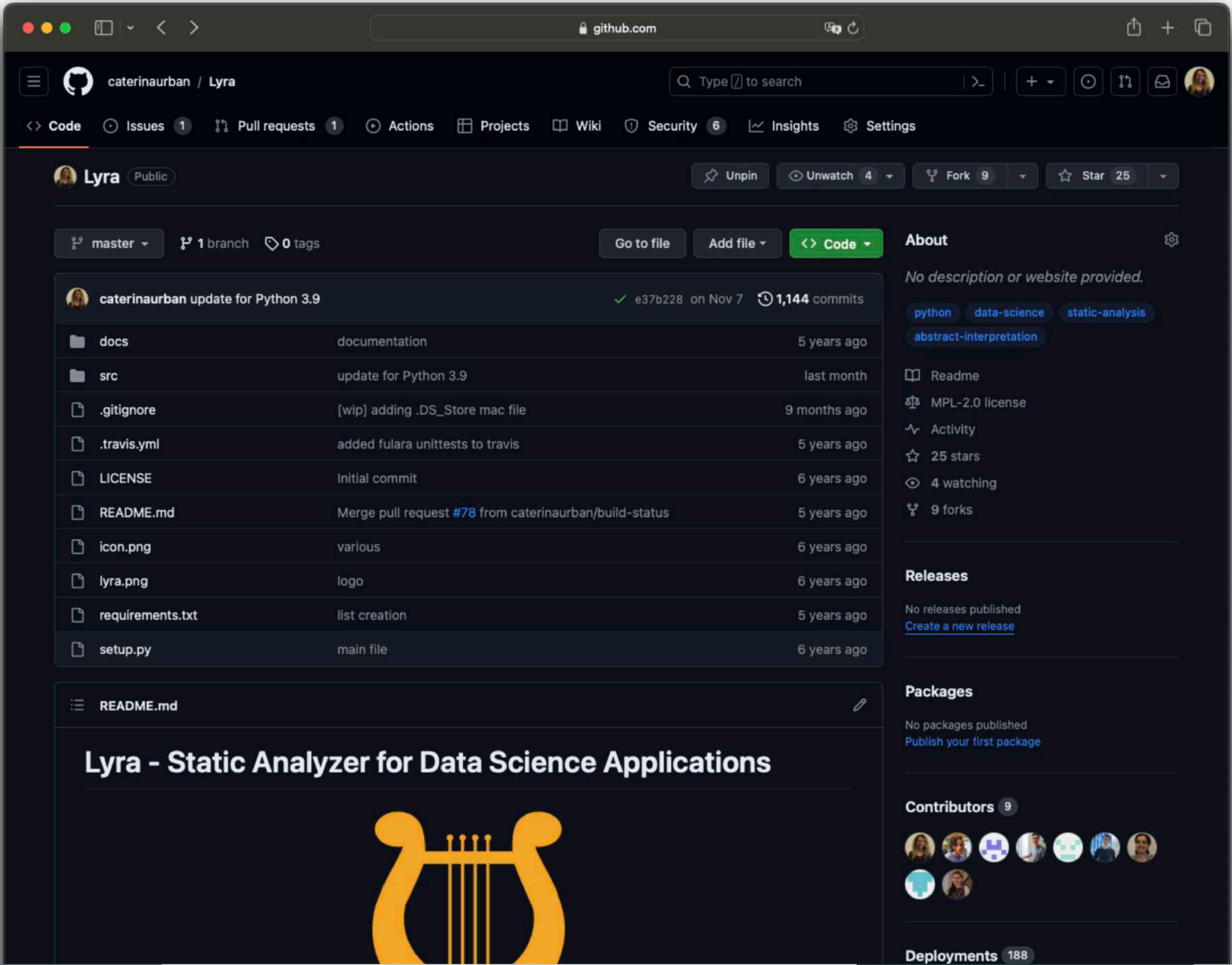


strongly-live variable analysis

syntactic non-usage

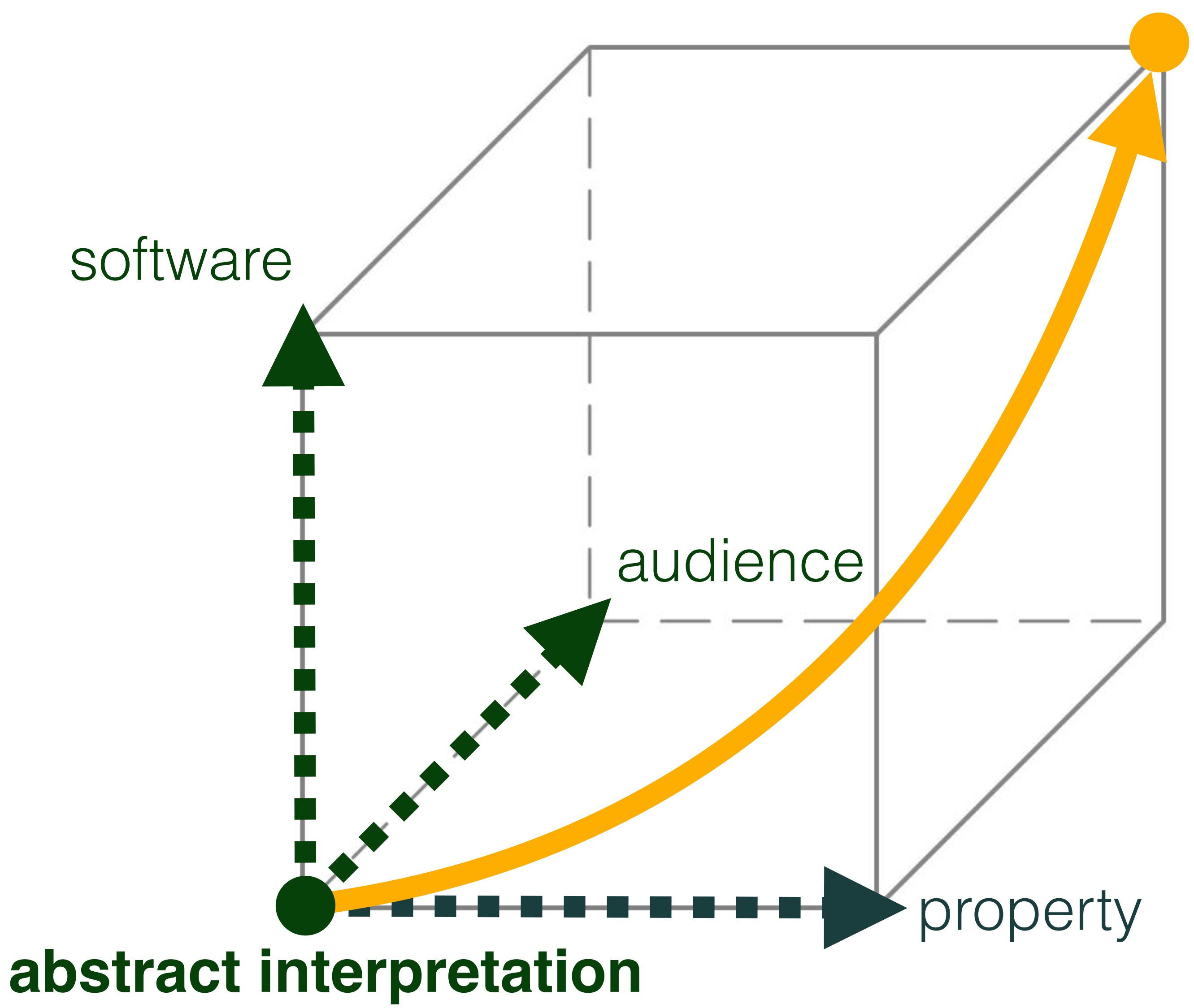
mathematical models
of the program behavior

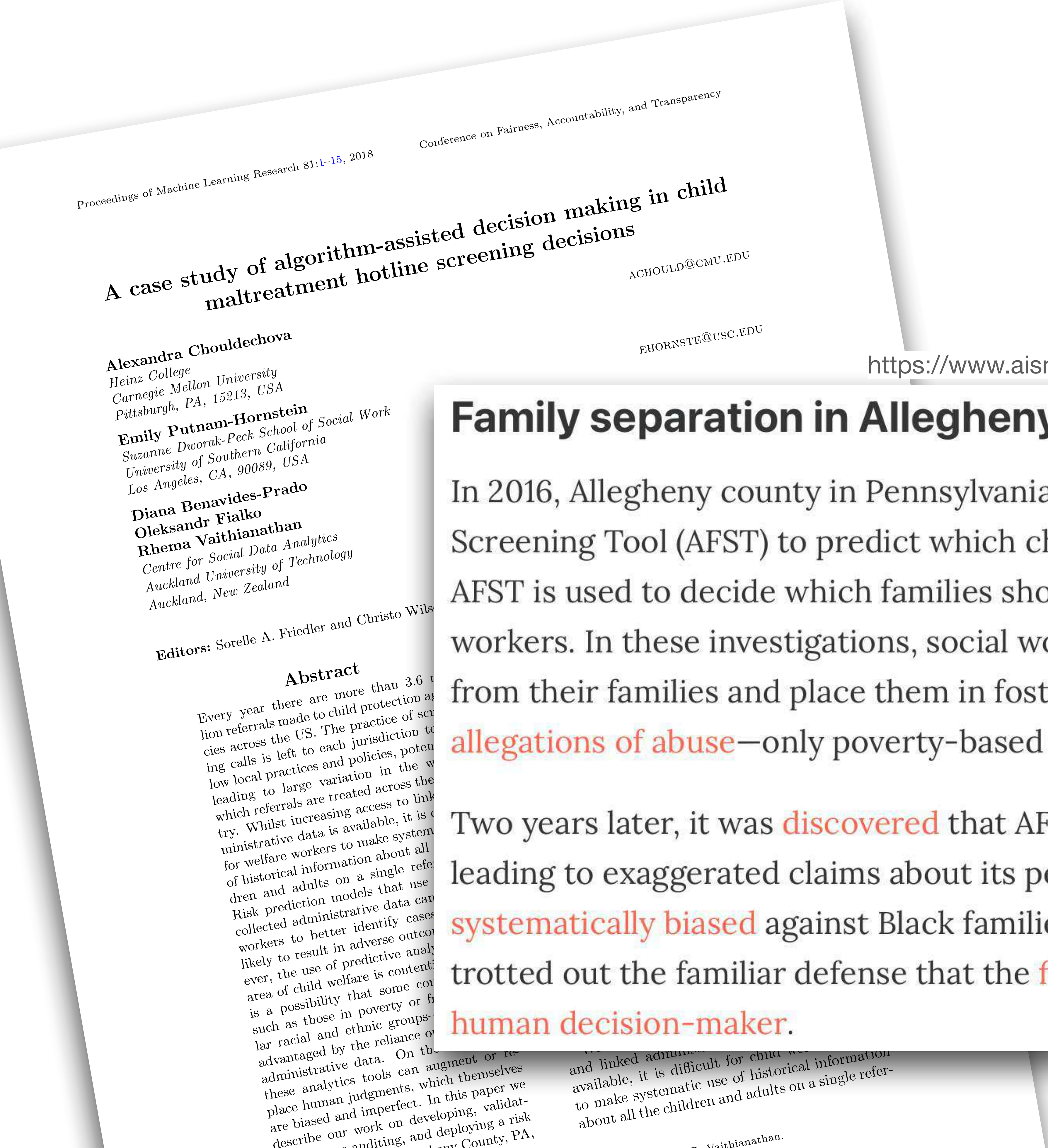



github.com
caterinaurban / Lyra
Code
Issues 1
Pull requests 1
Actions
Projects
Wiki
Security 6
Insights
Settings
Lyra Public
Unpin
Unwatch 4
Fork 9
Star 25
master
1 branch
0 tags
Go to file
Add file
Code
caterinaurban update for Python 3.9
e37b228 on Nov 7
1,144 commits
docs documentation 5 years ago
src update for Python 3.9 last month
.gitignore [wip] adding .DS_Store mac file 9 months ago
.travis.yml added fulara unittests to travis 5 years ago
LICENSE Initial commit 6 years ago
README.md Merge pull request #78 from caterinaurban/build-status 5 years ago
icon.png various 6 years ago
lyra.png logo 6 years ago
requirements.txt list creation 5 years ago
setup.py main file 6 years ago
README.md
Lyra - Static Analyzer for Data Science Applications
About
No description or website provided.
python
data-science
static-analysis
abstract-interpretation
Readme
MPL-2.0 license
Activity
25 stars
4 watching
9 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Contributors 9
Deployments 188

Data Leakage

Data Leakage Analysis




Proceedings of Machine Learning Research 81:1–15, 2018 Conference on Fairness, Accountability, and Transparency

A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions

Alexandra Chouldechova ACHOULD@CMU.EDU
Heinz College
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Emily Putnam-Hornstein EHORNSTE@USC.EDU
Suzanne Dworak-Peck School of Social Work
University of Southern California
Los Angeles, CA, 90089, USA

Diana Benavides-Prado
Oleksandr Fialko
Rhema Vaithianathan
Centre for Social Data Analytics
Auckland University of Technology
Auckland, New Zealand

Editors: Sorelle A. Friedler and Christo Wils

Abstract

https://www.aisnakeoil.com/p/the-bait-and-switch-behind-ai-risk

Family separation in Allegheny county

In 2016, Allegheny county in Pennsylvania adopted the Allegheny Family Screening Tool (AFST) to predict which children are at risk of maltreatment. AFST is used to decide which families should be investigated by social workers. In these investigations, social workers can forcibly remove children from their families and place them in foster care, even if there are no allegations of abuse—only poverty-based neglect.

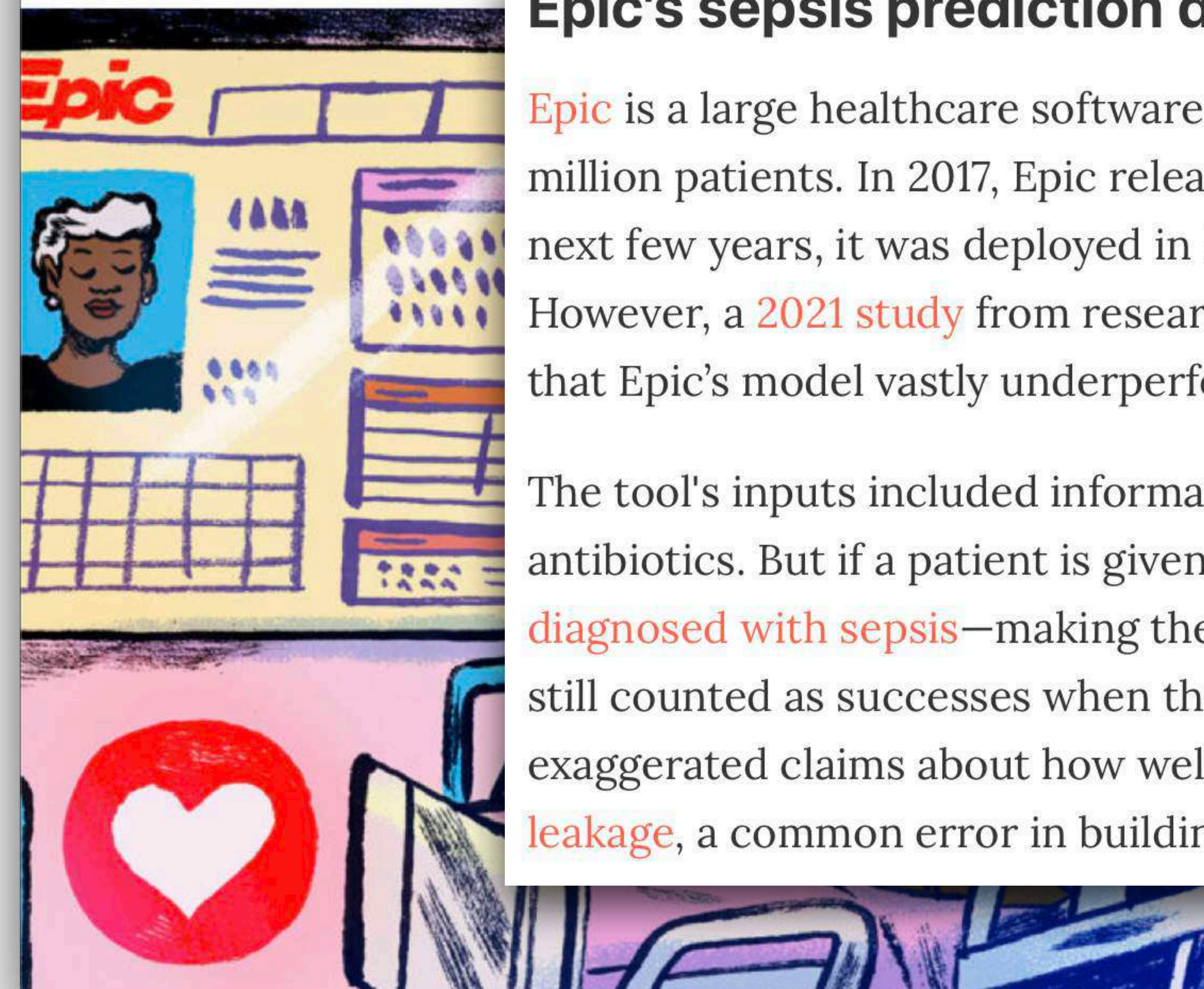
Two years later, it was discovered that AFST suffered from data leakage, leading to exaggerated claims about its performance. In addition, the tool was systematically biased against Black families. When questioned, the creators trotted out the familiar defense that the final decision is always made by a human decision-maker.

statnews.com

A STAT INVESTIGATION

Epic's sepsis algorithm is going off the rails in the real world. The use of these variables may explain why

By Casey Ross Sept. 27, 2021



https://www.aisnakeoil.com/p/the-bait-and-switch-behind-ai-risk

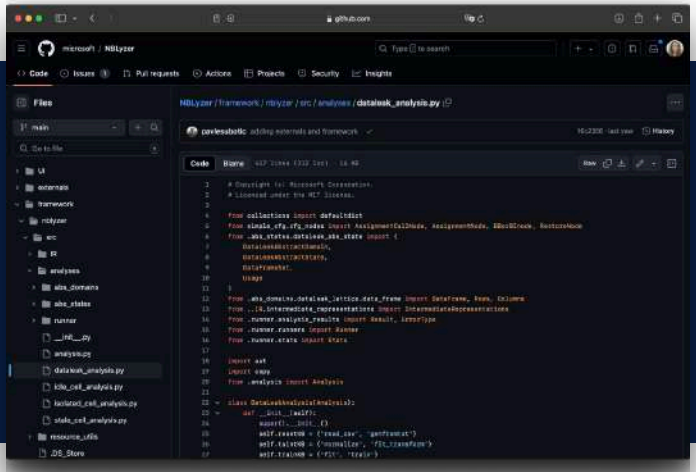
Epic's sepsis prediction debacle

Epic is a large healthcare software company. It stores health data for over 300 million patients. In 2017, Epic released a sepsis prediction model. Over the next few years, it was deployed in hundreds of hospitals across the U.S. However, a 2021 study from researchers at the University of Michigan found that Epic's model vastly underperformed compared to the developer's claims.

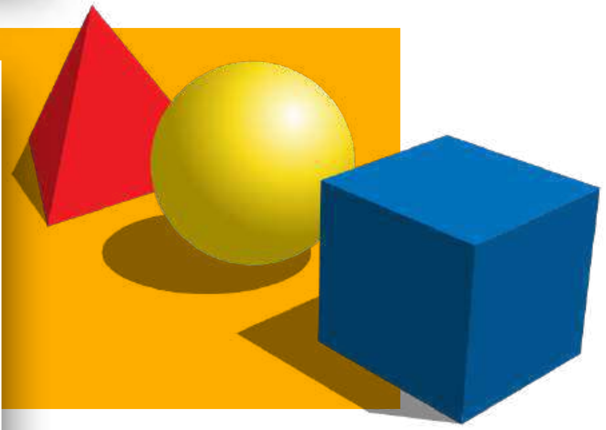
The tool's inputs included information about whether a patient was given antibiotics. But if a patient is given antibiotics, they have already been diagnosed with sepsis—making the tool's prediction useless. These cases were still counted as successes when the developer evaluated the tool, leading to exaggerated claims about how well it performed. This is an example of data leakage, a common error in building AI tools.

Data Leakage Analysis [Subotic24]

practical tools
targeting specific programs



algorithmic approaches
to decide program properties



An Abstract Interpretation-Based
Data Leakage Static Analysis

Filip Drobnjaković[1], Pavle Subotić[1], and Caterina Urban[2]

[1] Microsoft, Serbia
[2] Inria & ENS | PSL, France

Abstract. Data leakage is a well-known problem in machine learning which occurs when the training and testing datasets are not independent. This phenomenon leads to overly optimistic accuracy estimates at training time, followed by a significant drop in performance when models are deployed in the real world. This can be dangerous, notably when models are used for risk prediction in high-stakes applications.
In this paper, we propose an abstract interpretation-based static analysis to prove the absence of data leakage. We implemented it in the NBLyzer framework and we demonstrate its performance and precision on 2111 Jupyter notebooks from the Kaggle competition platform.

1 Introduction

As artificial intelligence (AI) continues its unprecedented impact on society, ensuring machine learning (ML) models are accurate is crucial. To this end, ML models need to be correctly trained and tested. This iterative task is typically performed within data science notebook environments [19,9]. A notable bug that can be introduced during this process is known as a *data leakage* [18]. Data leakages have been identified as a pervasive problem by the data science community [10,11,17]. In a number of recent cases data leakages crippled the performance of real-world risk prediction systems with dangerous consequences in high-stakes applications such as child welfare [1] and healthcare [24].

Data leakages arise when dependent data is used to train and test a model. This can come in the form of overlapping data sets or, more insidiously, by library transformations that create indirect data dependencies.

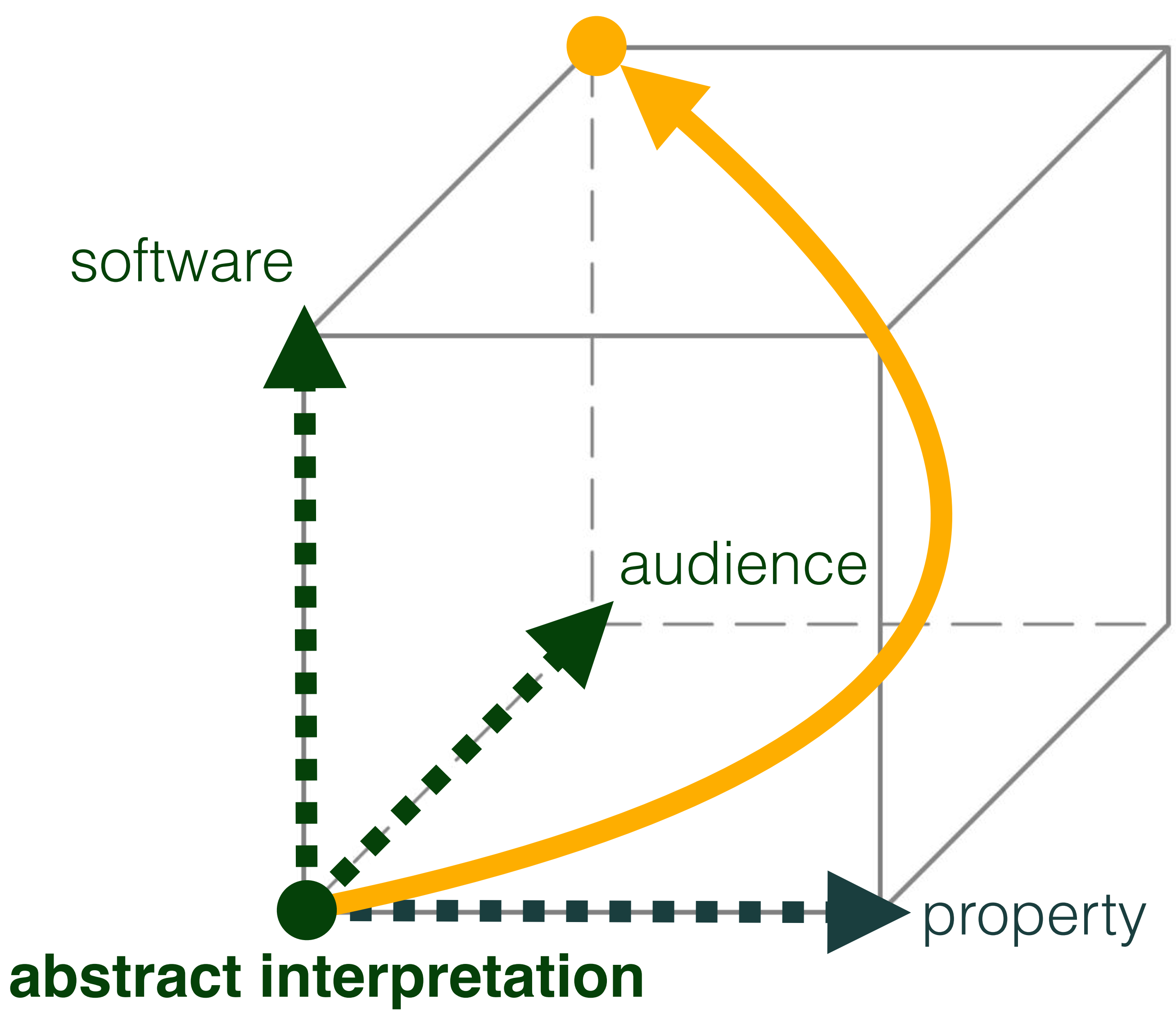
Example 1 (Motivating Example). Consider the following excerpt of a data science notebook (based on *569.ipynb* from our benchmarks, and written in the small language that we introduce in Section 3.3):

```
data = read("data.csv")
X_norm = normalize(X)
X_train = X_norm.select[[⌊0.025 * R_X_norm⌋ + 1, ..., R_X_norm]][]
X_test = X_norm.select[[0, ..., ⌊0.025 * R_X_norm⌋]][]
train(X_train)
test(X_test)
```

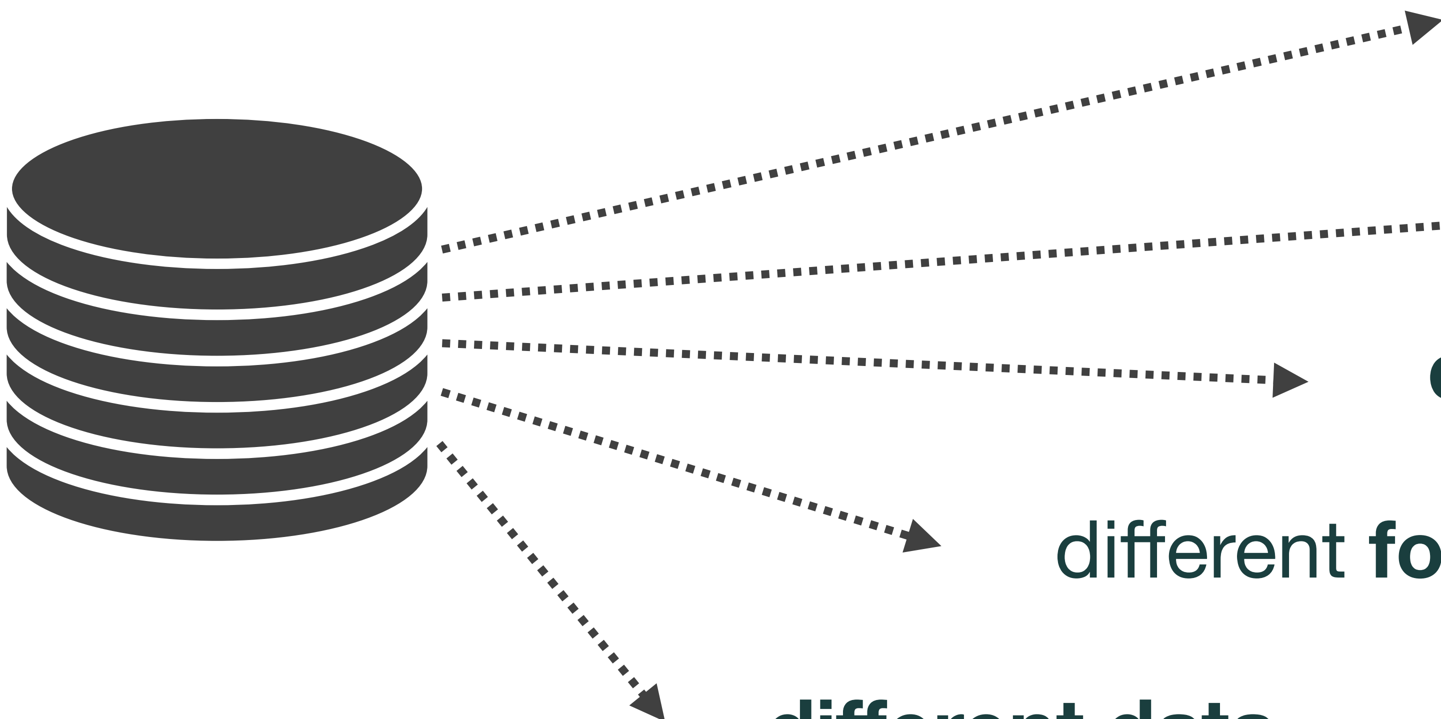
mathematical models
of the program behavior

Unexpected Data

(Un)expected Data Analysis



Unexpected Data



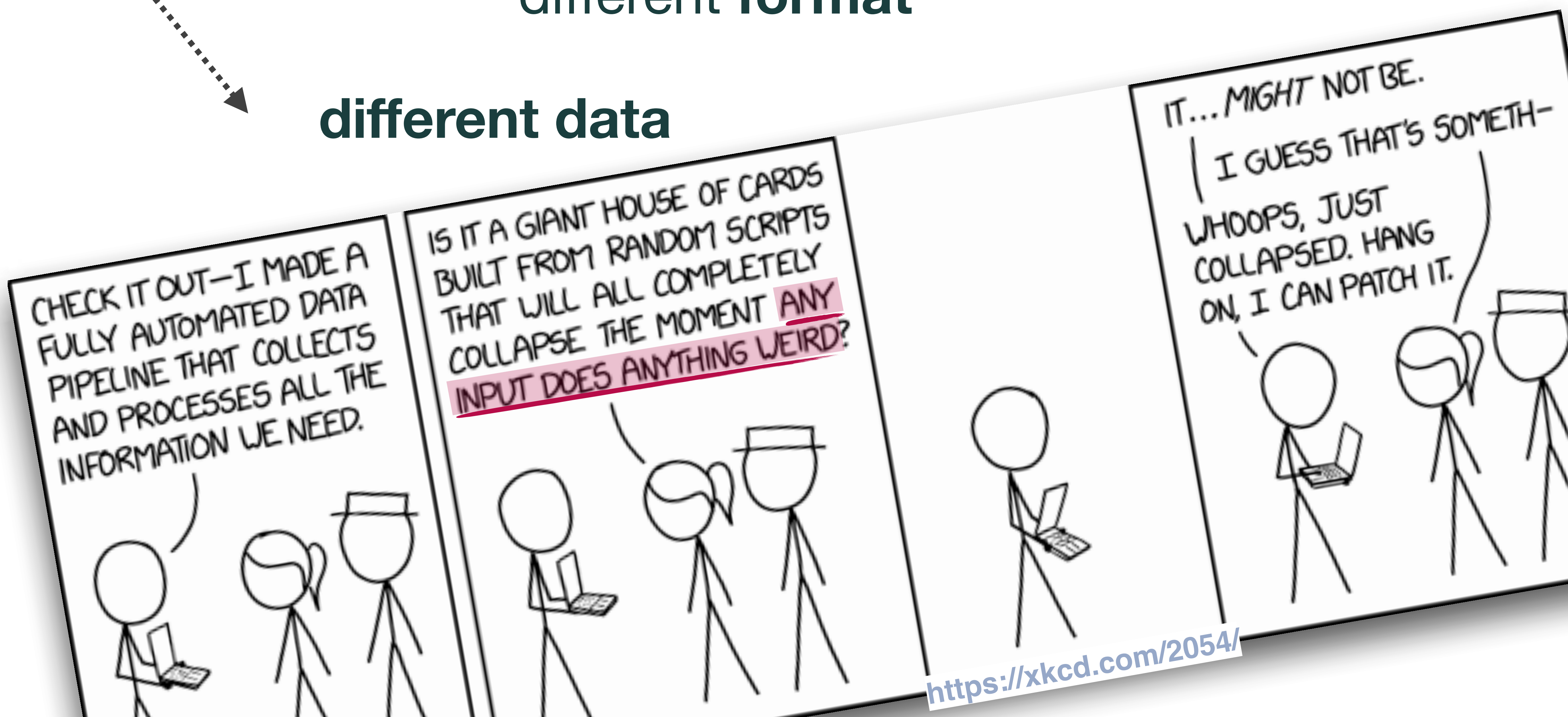
missing values

extra values

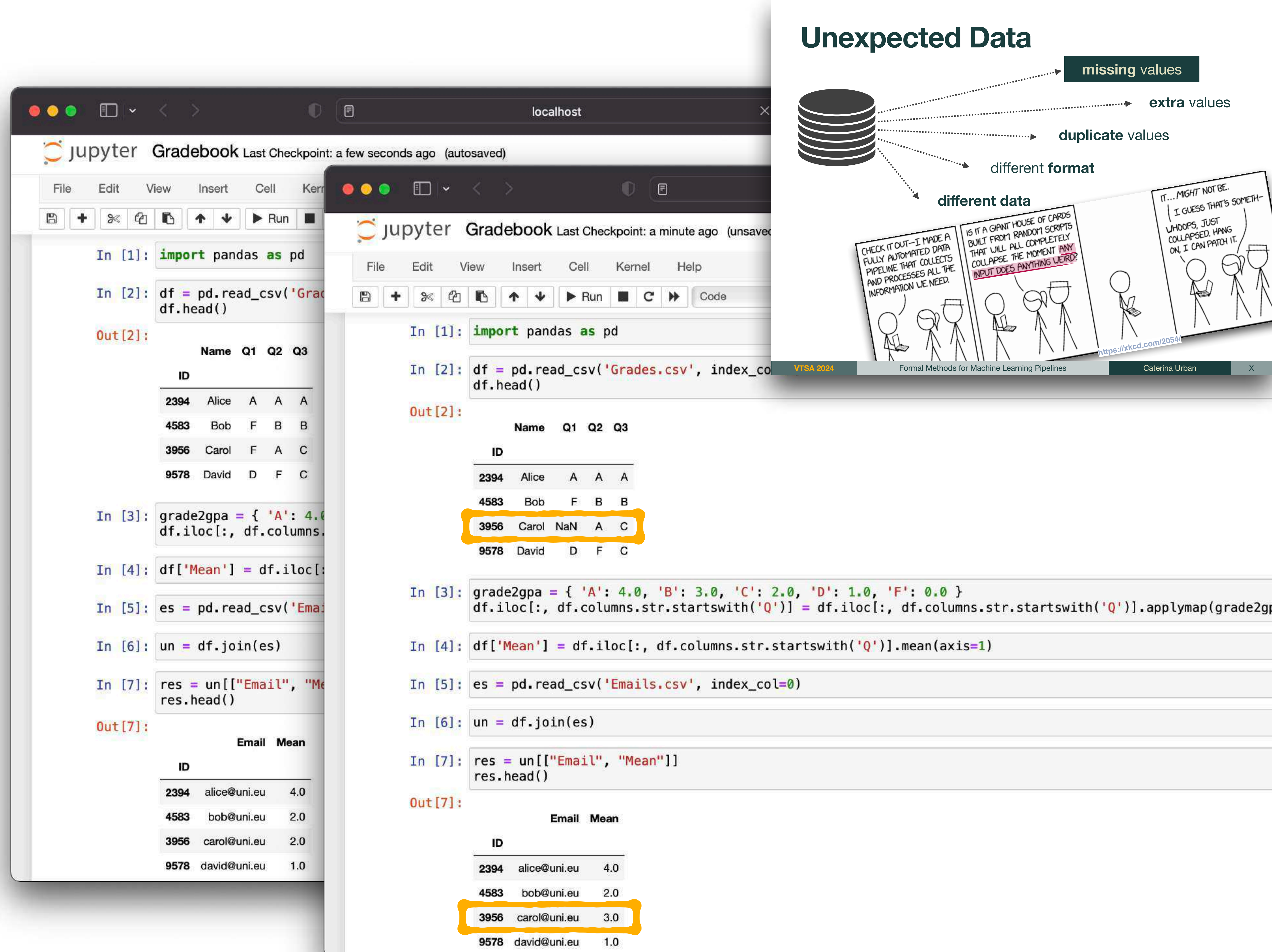
duplicate values

different **format**

different data



https://xkcd.com/2054/


Unexpected Data
missing values
extra values
duplicate values
different format
different data
CHECK IT OUT—I MADE A FULLY AUTOMATED DATA PIPELINE THAT COLLECTS AND PROCESSES ALL THE INFORMATION WE NEED.
IS IT A GIANT HOUSE OF CARDS BUILT FROM RANDOM SCRIPTS THAT WILL ALL COMPLETELY COLLAPSE THE MOMENT ANY INPUT DOES ANYTHING WEIRD?
IT... MIGHT NOT BE.
I GUESS THAT'S SOMETH—
WHOOPS, JUST COLLAPSED. HANG ON, I CAN PATCH IT.
https://xkcd.com/2054/
VTSA 2024
Formal Methods for Machine Learning Pipelines
Caterina Urban
localhost
jupyter Gradebook Last Checkpoint: a few seconds ago (autosaved)
jupyter Gradebook Last Checkpoint: a minute ago (unsaved
File Edit View Insert Cell Kernel Help
Run
Code
In [1]: import pandas as pd
In [2]: df = pd.read_csv('Grades.csv', index_co
df.head()
Out[2]:
Name Q1 Q2 Q3
ID
2394 Alice A A A
4583 Bob F B B
3956 Carol NaN A C
9578 David D F C
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gp
In [4]: df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)
In [5]: es = pd.read_csv('Emails.csv', index_col=0)
In [6]: un = df.join(es)
In [7]: res = un[["Email", "Mean"]]
res.head()
Out[7]:
Email Mean
ID
2394 alice@uni.eu 4.0
4583 bob@uni.eu 2.0
3956 carol@uni.eu 3.0
9578 david@uni.eu 1.0

localhost

jupyter Gradebook Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kern

```
In [1]: import pandas as pd
In [2]: df = pd.read_csv('Grad
        df.head()
```

Out[2]:

ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A
4583	Bob	F	B	B
3956	Carol	F	A	C
9578	David	D	F	C

```
In [3]: grade2gpa = { 'A': 4.0
        df.iloc[:, df.columns.
In [4]: df['Mean'] = df.iloc[:
In [5]: es = pd.read_csv('Emai
In [6]: un = df.join(es)
In [7]: res = un[["Email", "Me
        res.head()
```

Out[7]:

ID	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	2.0
3956	carol@uni.eu	2.0
9578	david@uni.eu	1.0

jupyter Gradebook Last Checkpoint: 2 minutes ago (unsave

File Edit View Insert Cell Kernel Help

Run Code

```
In [1]: import pandas as pd
In [2]: df = pd.read_csv('Grades.csv', index_col=0)
        df.head()
```

Out[2]:

ID	Name	Q1	Q2	Q3
2394	Alice	A	A	A	A
4583	Bob	F	B	B	NaN
3956	Carol	F	A	C	NaN
9578	David	D	F	C	NaN

```
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
        df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gp
In [4]: df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)
In [5]: es = pd.read_csv('Emails.csv', index_col=0)
In [6]: un = df.join(es)
In [7]: res = un[["Email", "Mean"]]
        res.head()
```

Out[7]:

	Email	Mean
2394	alice@uni.eu	4.0
4583	bob@uni.eu	3.0
3956	carol@uni.eu	3.0
9578	david@uni.eu	1.0

Unexpected Data

missing values

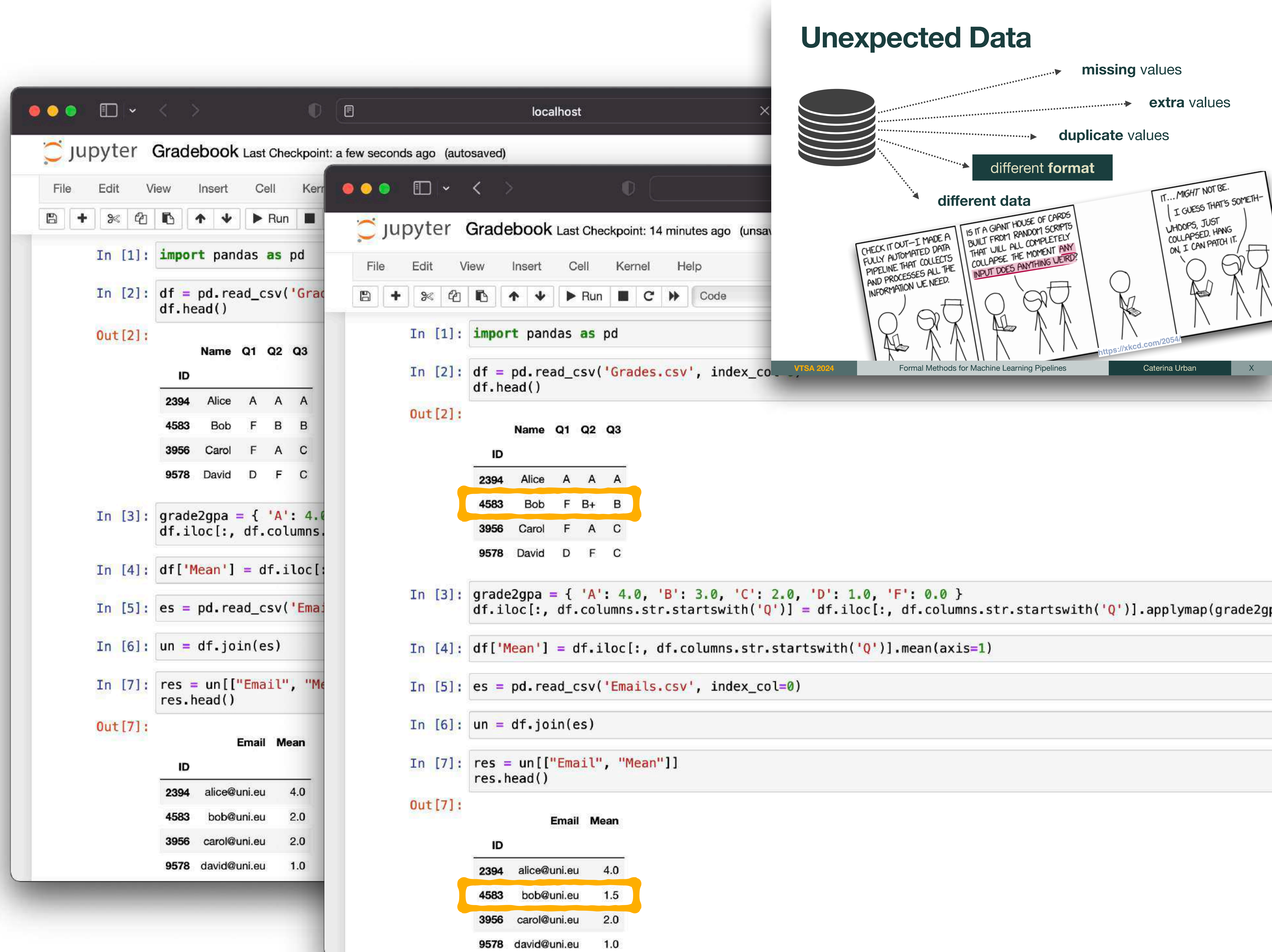
extra values

duplicate values

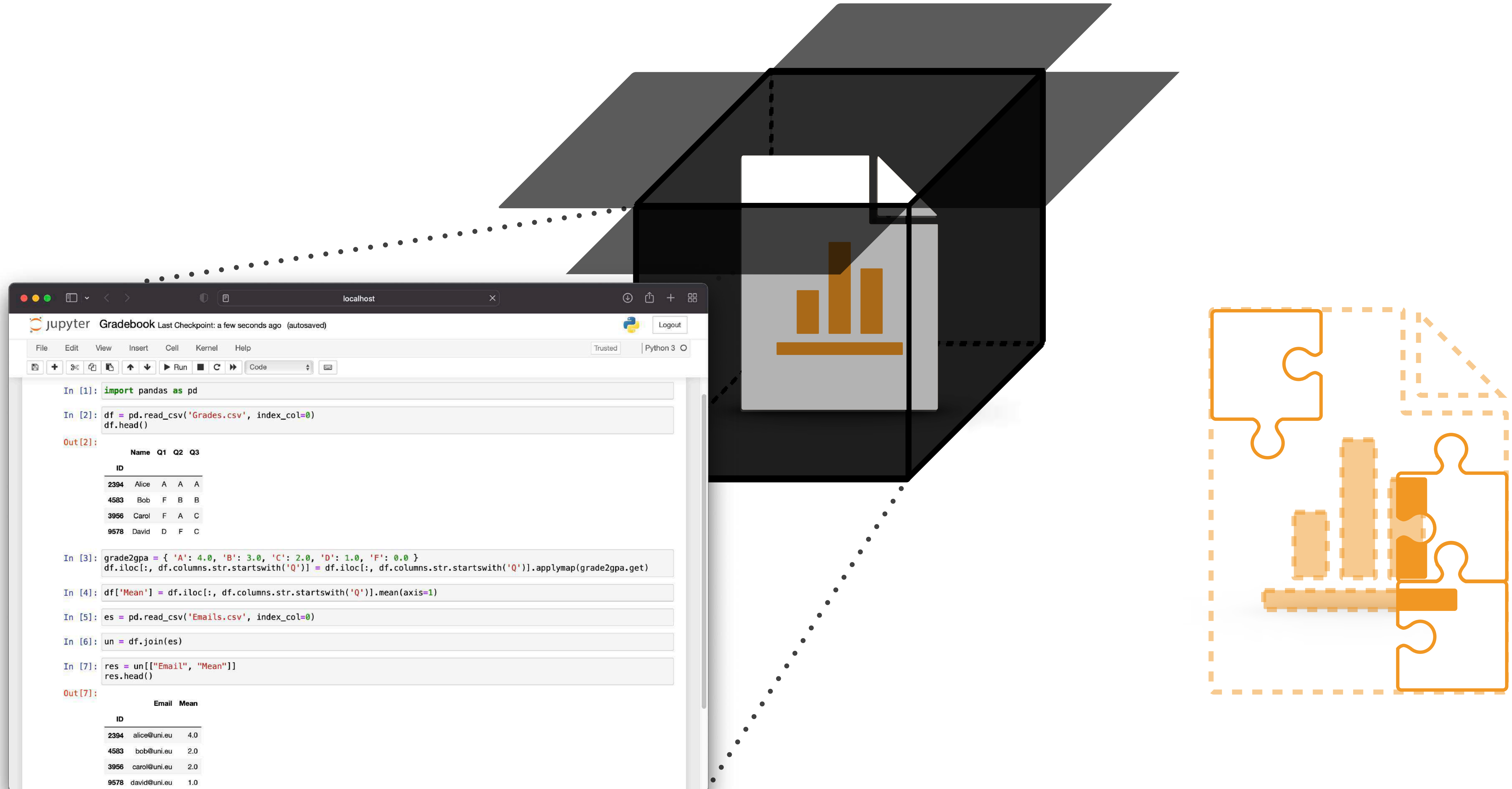
different **format**

different data

https://xkcd.com/2054/

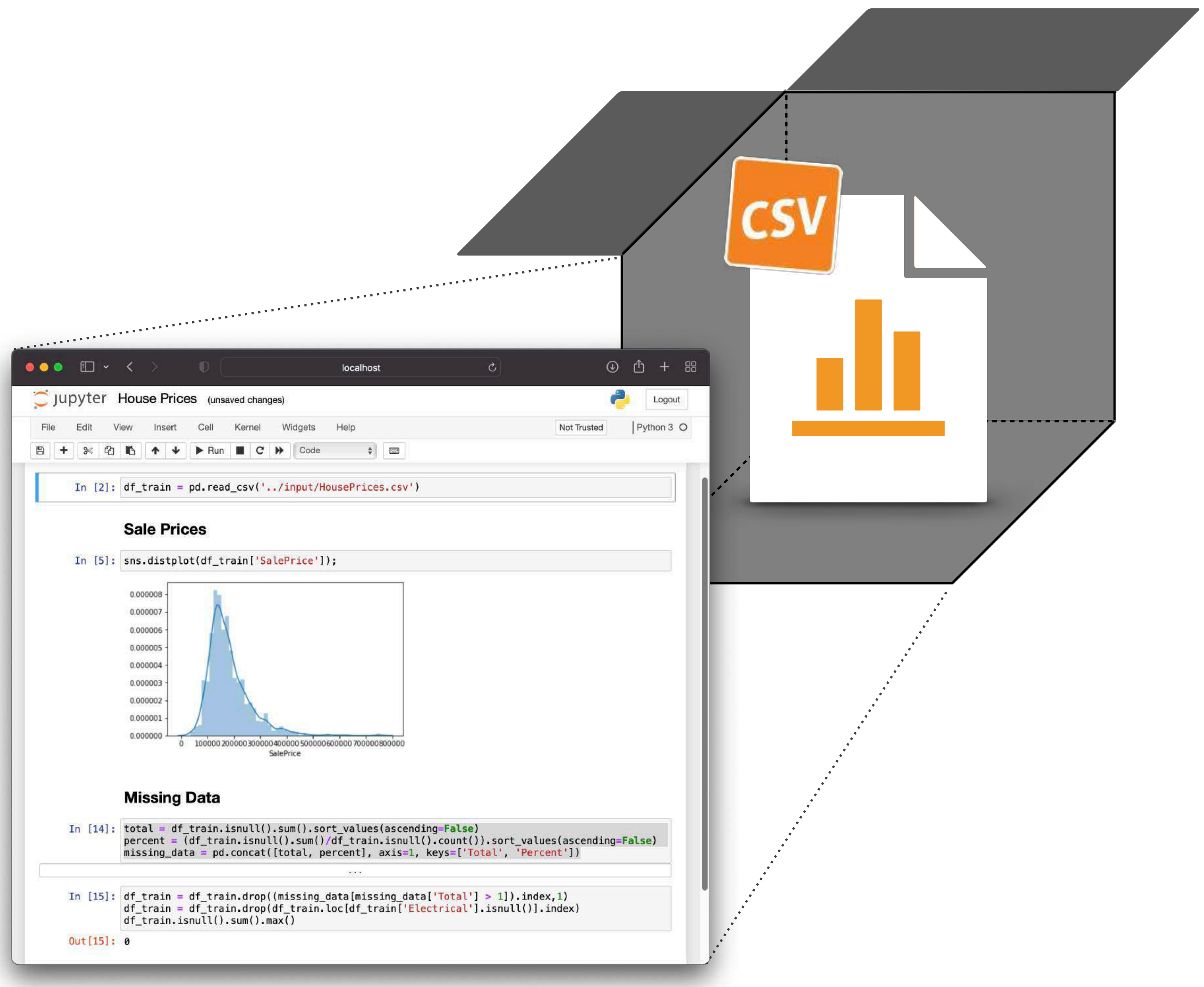

Unexpected Data
missing values
extra values
duplicate values
different format
different data
CHECK IT OUT—I MADE A FULLY AUTOMATED DATA PIPELINE THAT COLLECTS AND PROCESSES ALL THE INFORMATION WE NEED.
IS IT A GIANT HOUSE OF CARDS BUILT FROM RANDOM SCRIPTS THAT WILL ALL COMPLETELY COLLAPSE THE MOMENT ANY INPUT DOES ANYTHING WEIRD?
IT... MIGHT NOT BE.
I GUESS THAT'S SOMETH—
WHOOPS, JUST COLLAPSED. HANG ON, I CAN PATCH IT.
https://xkcd.com/2054/
VTSA 2024
Formal Methods for Machine Learning Pipelines
Caterina Urban
localhost
jupyter Gradebook Last Checkpoint: a few seconds ago (autosaved)
jupyter Gradebook Last Checkpoint: 14 minutes ago
File Edit View Insert Cell Kernel Help
Run
Code
In [1]: import pandas as pd
In [2]: df = pd.read_csv('Grades.csv', index_col=0)
df.head()
Out[2]:
Name Q1 Q2 Q3
ID
2394 Alice A A A
4583 Bob F B+ B
3956 Carol F A C
9578 David D F C
In [3]: grade2gpa = { 'A': 4.0, 'B': 3.0, 'C': 2.0, 'D': 1.0, 'F': 0.0 }
df.iloc[:, df.columns.str.startswith('Q')] = df.iloc[:, df.columns.str.startswith('Q')].applymap(grade2gp
In [4]: df['Mean'] = df.iloc[:, df.columns.str.startswith('Q')].mean(axis=1)
In [5]: es = pd.read_csv('Emails.csv', index_col=0)
In [6]: un = df.join(es)
In [7]: res = un[["Email", "Mean"]]
res.head()
Out[7]:
Email Mean
ID
2394 alice@uni.eu 4.0
4583 bob@uni.eu 1.5
3956 carol@uni.eu 2.0
9578 david@uni.eu 1.0

Data Expectations Analysis



Data

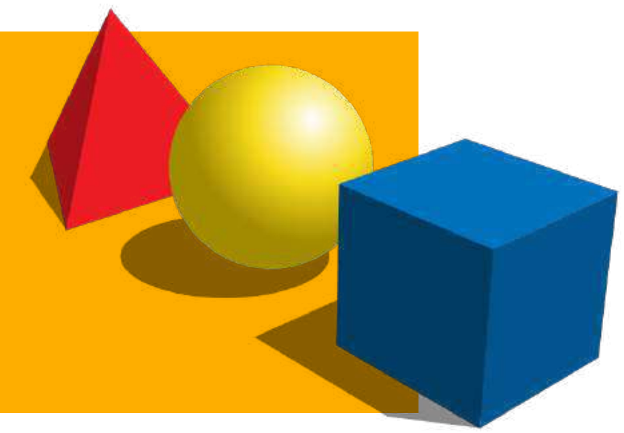
1st Challenge: Multi-Dimensional Data Structures



Data Expectations Analysis

practical tools
targeting specific programs

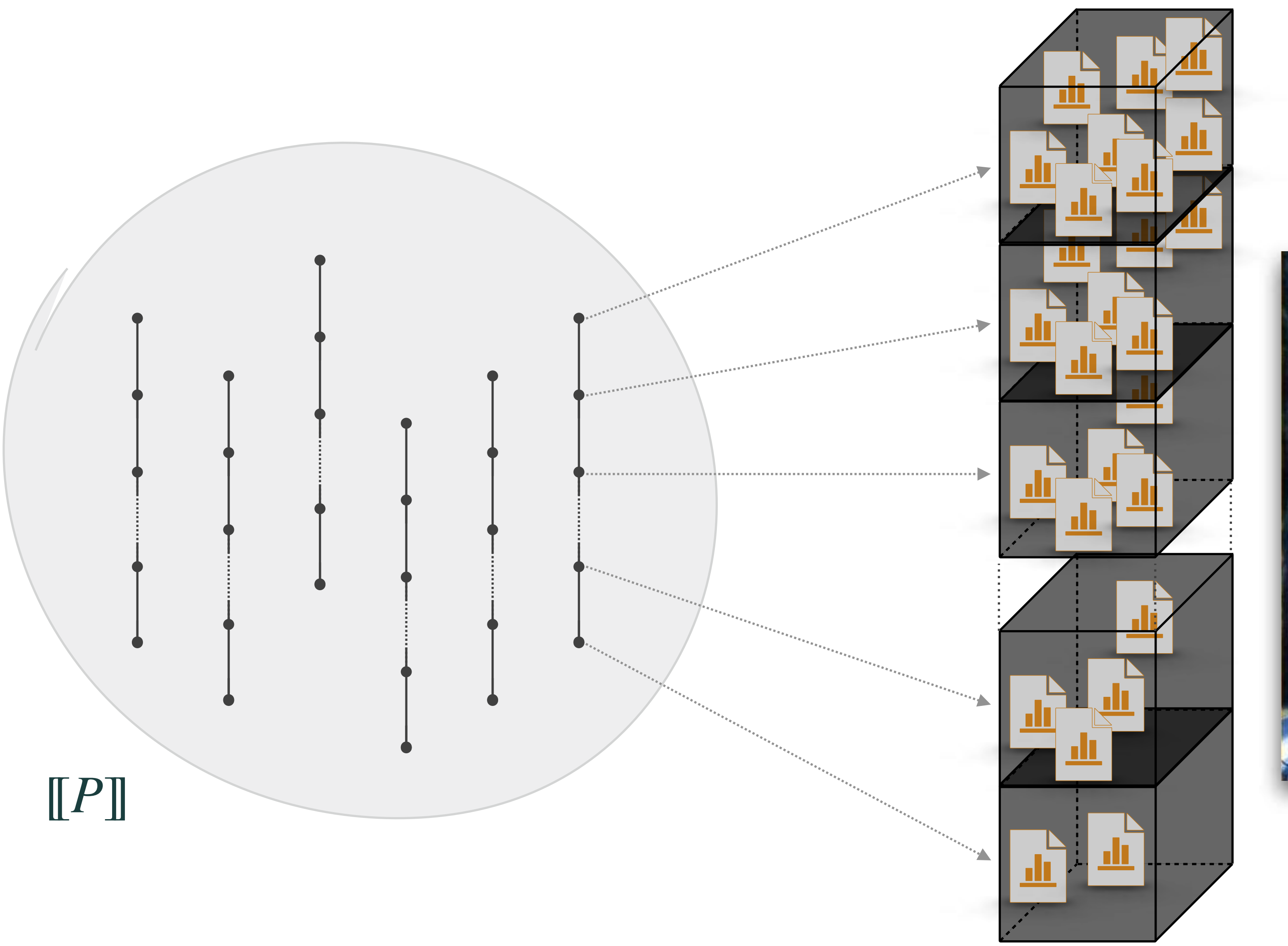
algorithmic approaches
to decide program properties

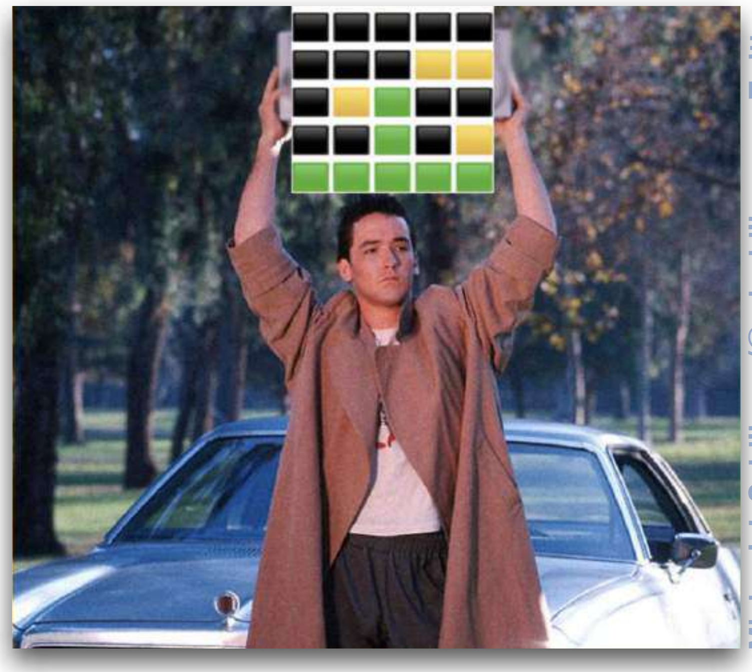


mathematical models
of the program behavior

Concrete Semantics

2nd Challenge: Indirect Reasoning





Michael J. Seidlinger/@mjseidlinger on Twitter

Abstract Semantics

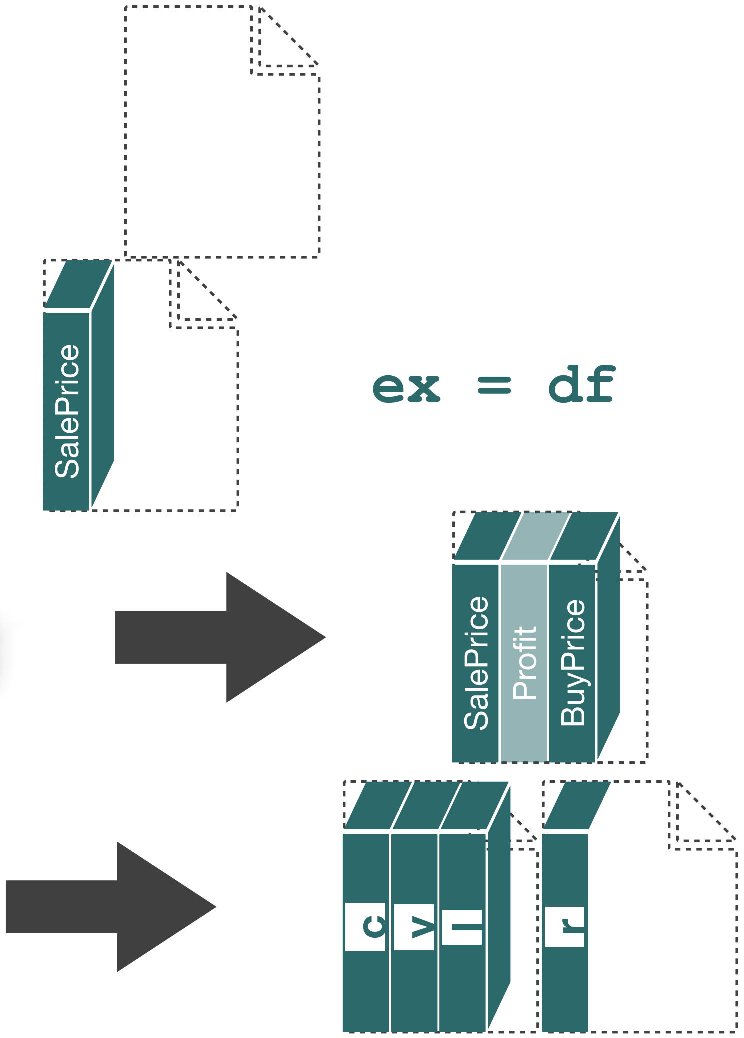
3rd Challenge: Complex Library Calls

```
import pandas as pd
df = pd.read_csv("HousePrices.csv")
```

```
ex = df[df.SalePrice >= 1000000]
```

```
ex['Profit'] = ex['SalePrice'] - ex['BuyPrice']
```

```
⋮
dL = pd.read_csv("L.csv")
dP = dL.pivot(index=c, columns=y, values=l)
dR = pd.read_csv("R.csv")
dG = dP.loc[:, 0:35].groupby(dR[r])
```



ex = df

dR[r] ∈ dG

dP ∩ dR

Pratical Static Analysis

Necessary vs Sufficient Data Expectations

Automatic Inference of Necessary Preconditions

Patrick Cousot[1], Radhia Cousot[2], Manuel Fähndrich[3], and Francesco Logozzo[3]

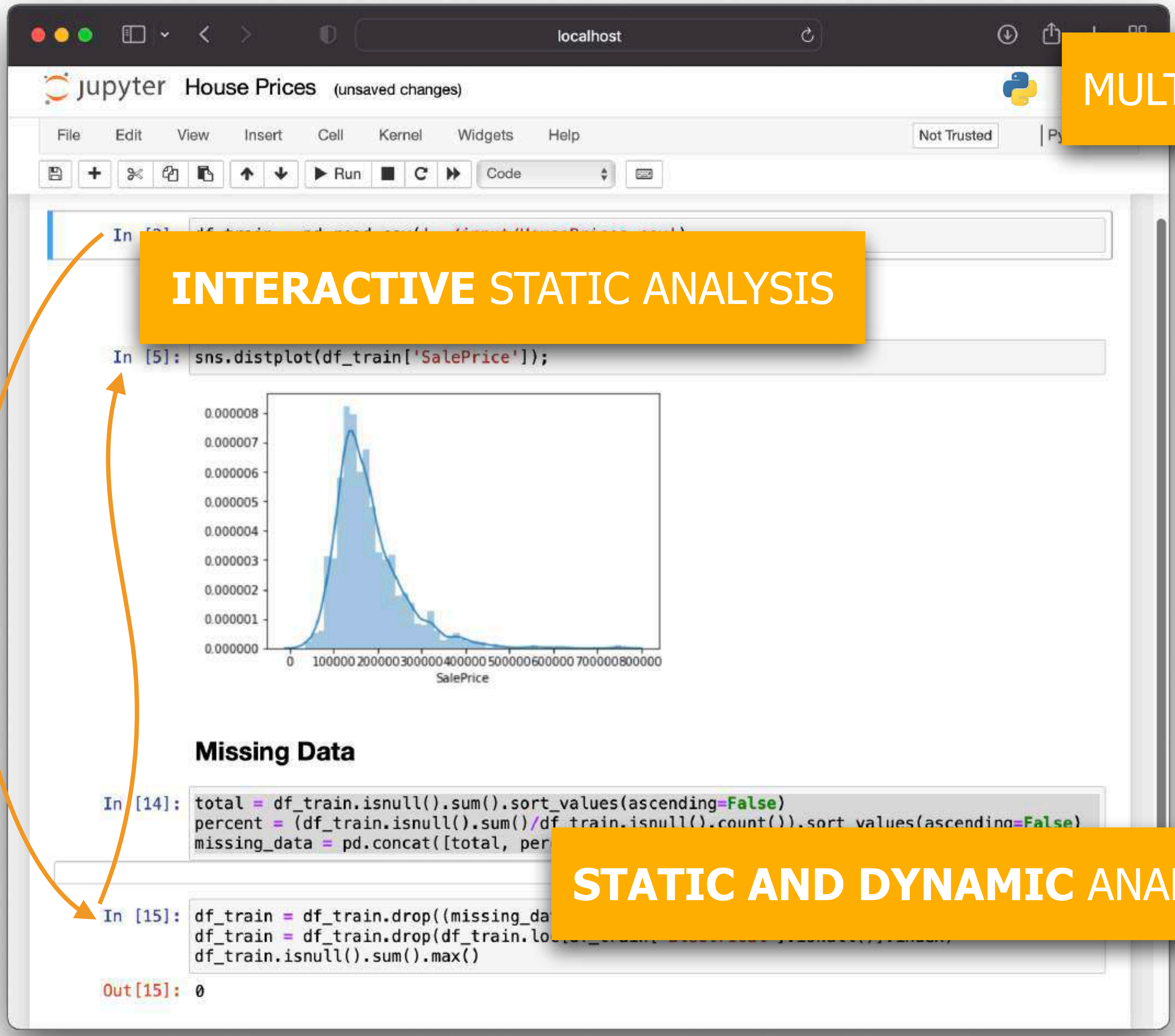
[1] NYU, ENS, CNRS, INRIA
pcousot@cims.nyu.edu
[2] CNRS, ENS, INRIA
rcousot@ens.fr
[3] Microsoft Research
{maf,logozzo}@microsoft.com

Abstract. We consider the problem of *automatic* precondition inference. We argue that the common notion of *sufficient* precondition inference (*i.e.*, under which precondition is the program correct?) imposes too large a burden on callers, and hence it is unfit for automatic program analysis. Therefore, we define the problem of *necessary* precondition inference (*i.e.*, under which precondition, if violated, will the program *always* be incorrect?). We designed and implemented several new abstract interpretation-based analyses to infer atomic, disjunctive, universally and existentially quantified necessary preconditions.

We experimentally validated the analyses on large scale industrial

Implementation

Wish List



MULTI-LANGUAGE SUPPORT

INTERACTIVE STATIC ANALYSIS

STATIC AND DYNAMIC ANALYSIS COMBINATIONS

Data Expectations Analysis

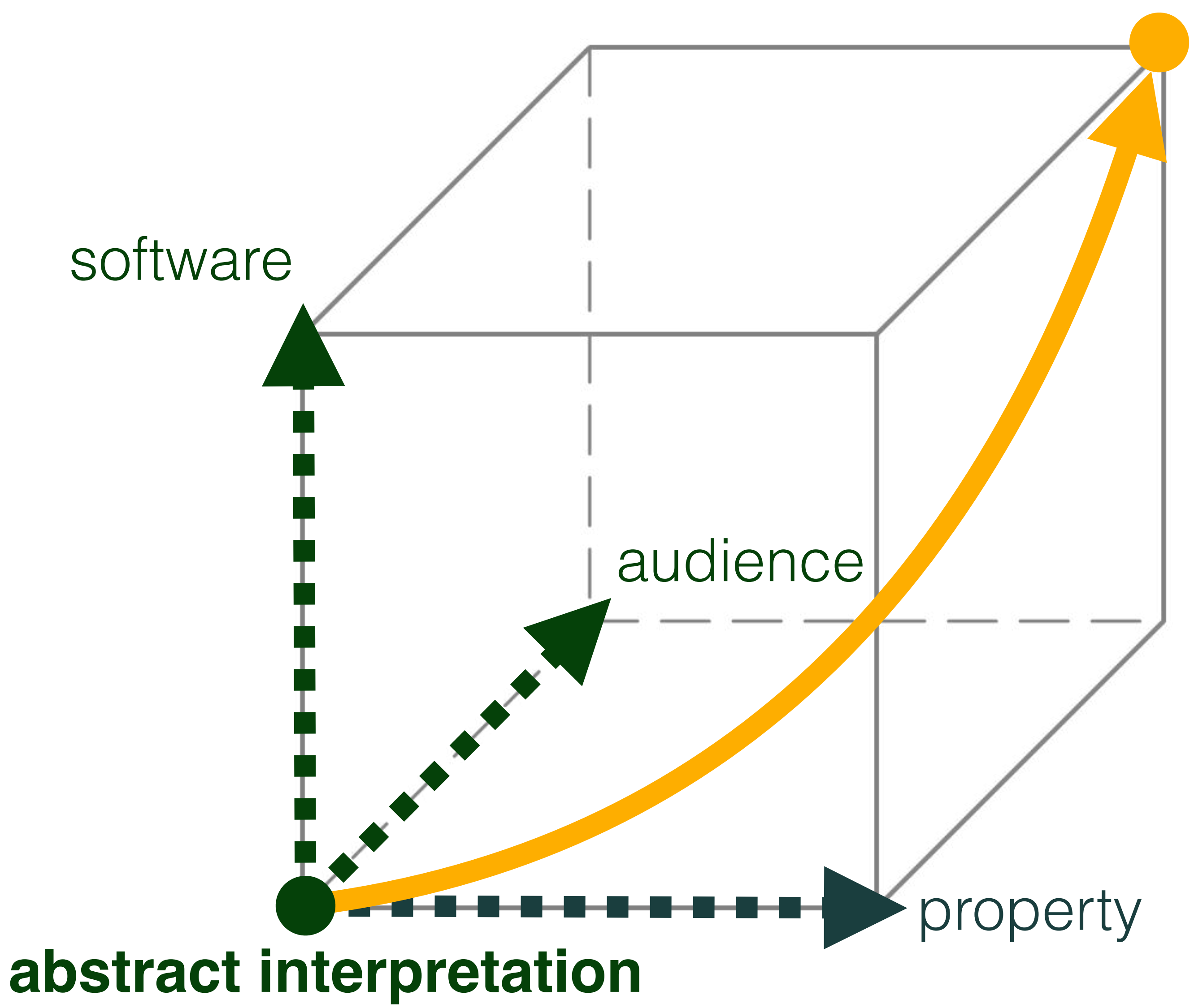
practical tools
targeting specific progr

algorithmic appro
to decide program

mathematical mode
of the program behav

(Un)expected + (Un)used Data

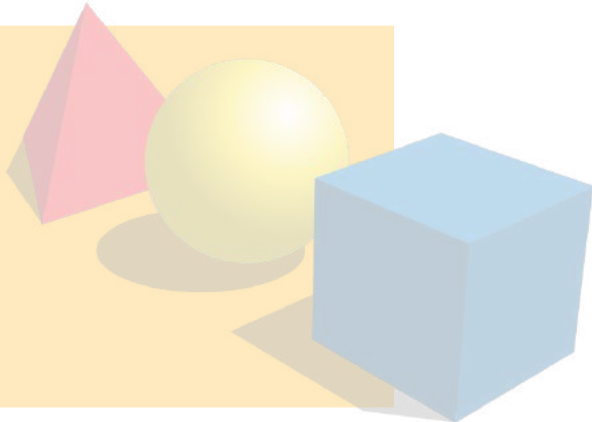
(Un)expected + (Un)used Analysis



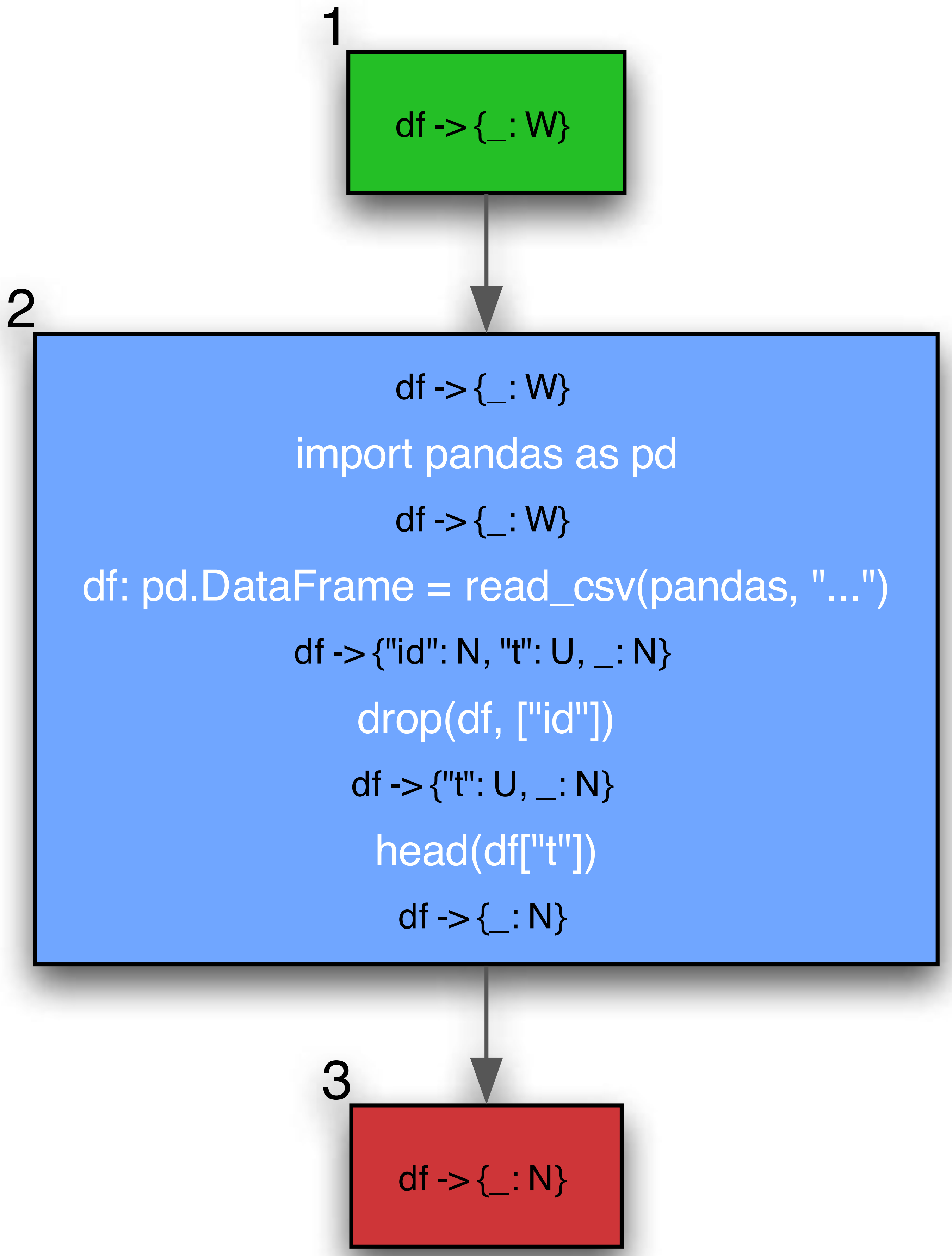
Expectations + Usage Analysis

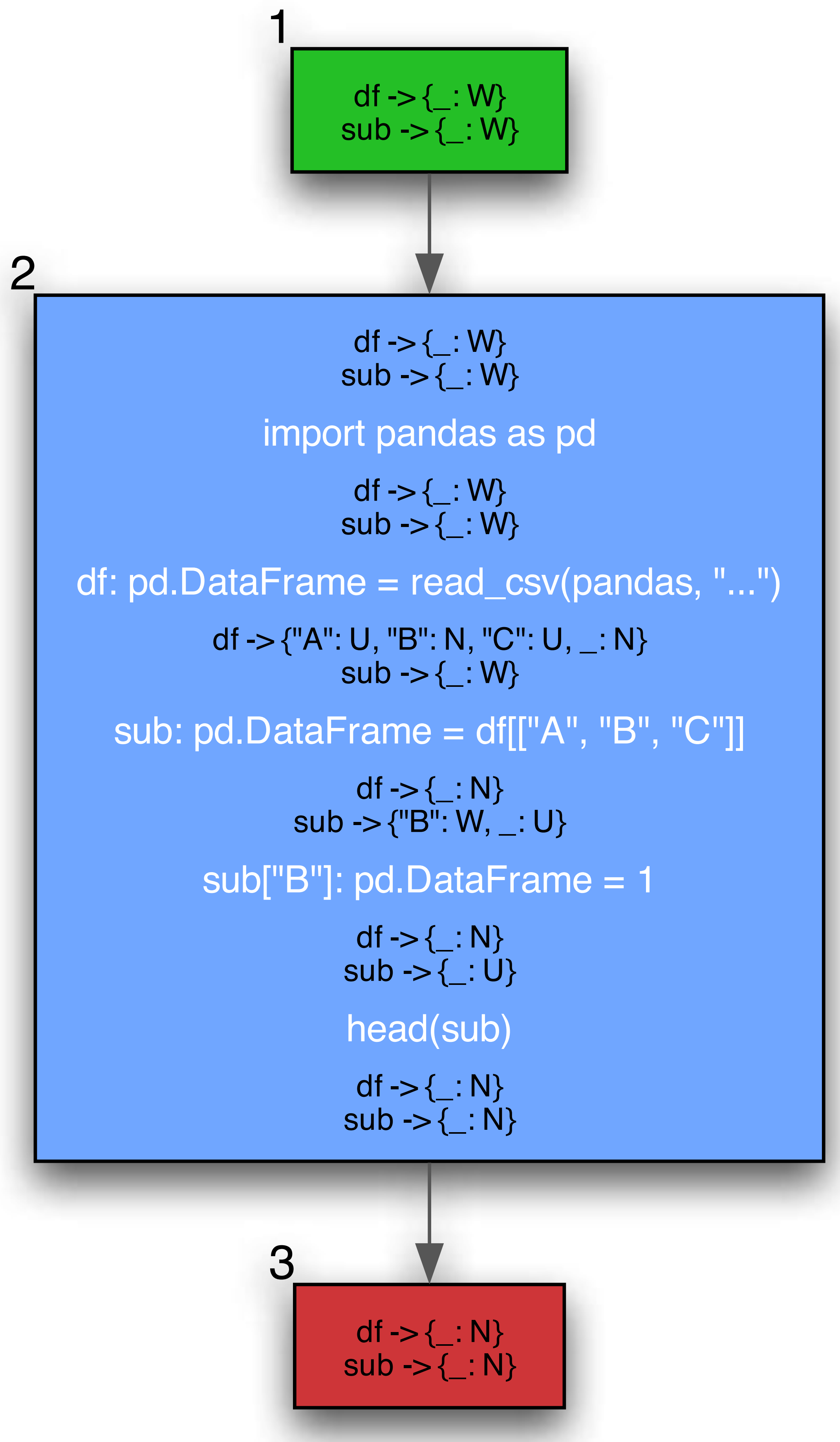
practical tools
targeting specific programs

algorithmic approaches
to decide program properties



mathematical models
of the program behavior


1
df -> {_: W}
2
df -> {_: W}
import pandas as pd
df -> {_: W}
df: pd.DataFrame = read_csv(pandas, "...")
df -> {"id": N, "t": U, _: N}
drop(df, ["id"])
df -> {"t": U, _: N}
head(df["t"])
df -> {_: N}
3
df -> {_: N}


1
df -> {_: W}
sub -> {_: W}
2
df -> {_: W}
sub -> {_: W}
import pandas as pd
df -> {_: W}
sub -> {_: W}
df: pd.DataFrame = read_csv(pandas, "...")
df -> {"A": U, "B": N, "C": U, _: N}
sub -> {_: W}
sub: pd.DataFrame = df[["A", "B", "C"]]
df -> {_: N}
sub -> {"B": W, _: U}
sub["B"]: pd.DataFrame = 1
df -> {_: N}
sub -> {_: U}
head(sub)
df -> {_: N}
sub -> {_: N}
3
df -> {_: N}
sub -> {_: N}

Bibliography

[Kurd03] **Zeshan Kurd, Tim Kelly**. Establishing Safety Criteria for Artificial Neural Networks. In KES, 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, 2019.

[Singh19] **Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev**. An Abstract Domain for Certifying Neural Networks. In POPL, 2019.

[Munakata23] **Satoshi Munakata, CU, Haruki Yokoyama, Koji Yamamoto, Kazuki Munakata**. Verifying Attention Robustness of Deep Neural Networks against Semantic Perturbations. In NFM, 2023.

[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, 2020.

Bibliography

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

[Julian16] **Kyle D. Julian, Jessica Lopez, Jeffrey S. Brush, Michael P. Owen, Mykel J. Kochenderfer**. Policy Compression for Aircraft Collision Avoidance Systems. In DASC, 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, 2017.

[Galhotra17] **Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou**. Fairness Testing: Testing Software for Discrimination. In FSE, 2017.

[Urban20] **CU, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang**. Perfectly Parallel Fairness Certification of Neural Networks. In OOPSLA, 2020.

Bibliography

[Urban21] **CU and Antoine Miné**. A Review of Formal Methods applied to Machine Learning. https://arxiv.org/abs/2104.02466, 2021.

[Pal24] **Abhinandan Pal, Francesco Ranzato, CU, Marco Zanella.** Abstract Interpretation-Based Feature Importance for Support Vector Machines. In VMCAI, 2024.

[R19] **Francesco Ranzato and Marco Zanella.** Robustness Verification of Support Vector Machines. In SAS, 2019.

[Ranzato21] **Francesco Ranzato, CU, and Marco Zanella**. *Fairness-Aware Training of Decision Trees by Abstract Interpretation*. In CIKM 2021.

[CU18] **CU and Peter Müller**. An Abstract Interpretation Framework for Data Usage. In ESOP 2018.

[Subotic24] **Filip Drobnjaković, Pavle Subotić, CU**. An Abstract Interpretation-Based Data Leakage Static Analysis. In TASE 2024.