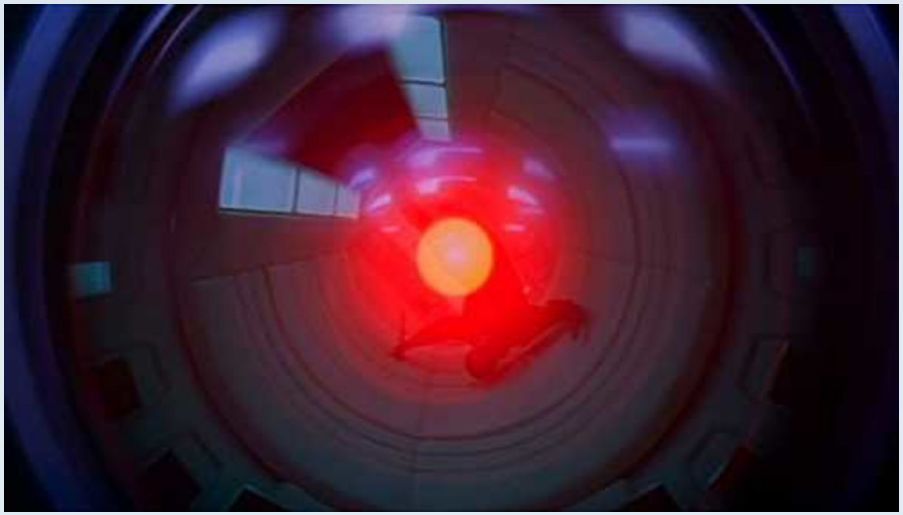
Towards Probabilistic Verification of AI Systems via Weighted Model Integration

Paolo Morettin

Advanced Topics in Machine Learning (2025)

Funded by
the European Union

Vague motivational slide



On trust

You just finished training the controller of an autonomous vehicle.

On trust

You just finished training the controller of an autonomous vehicle.

- What to do before deployment?

On trust

You just finished training the controller of an autonomous vehicle.

- What to do before deployment?
- Accuracy on the test set: **99.9%**. Do we deploy the system now?

On trust

You just finished training the controller of an autonomous vehicle.

- What to do before deployment?
- Accuracy on the test set: **99.9%**. Do we deploy the system now?
- How informative is **99.9%**?

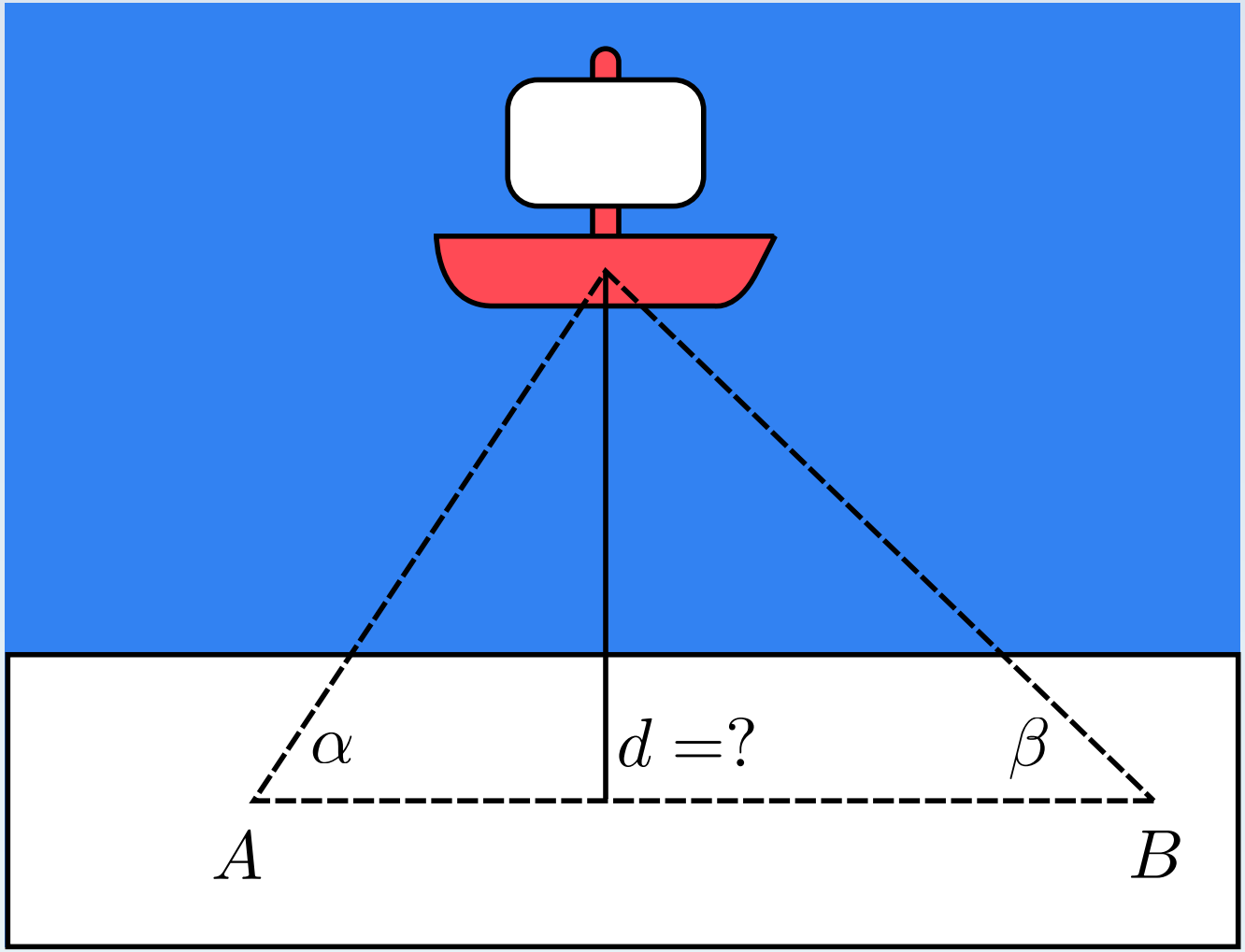
On trust

You just finished training the controller of an autonomous vehicle.

- What to do before deployment?
- Accuracy on the test set: **99.9%**. Do we deploy the system now?
- How informative is **99.9%**?

`https://en.wikipedia.org/wiki/List_of_Tesla_Autopilot_crashes`

On trust



On trust

Why are we trusting the software running on airplanes?

On trust

Why are we trusting the software running on airplanes?

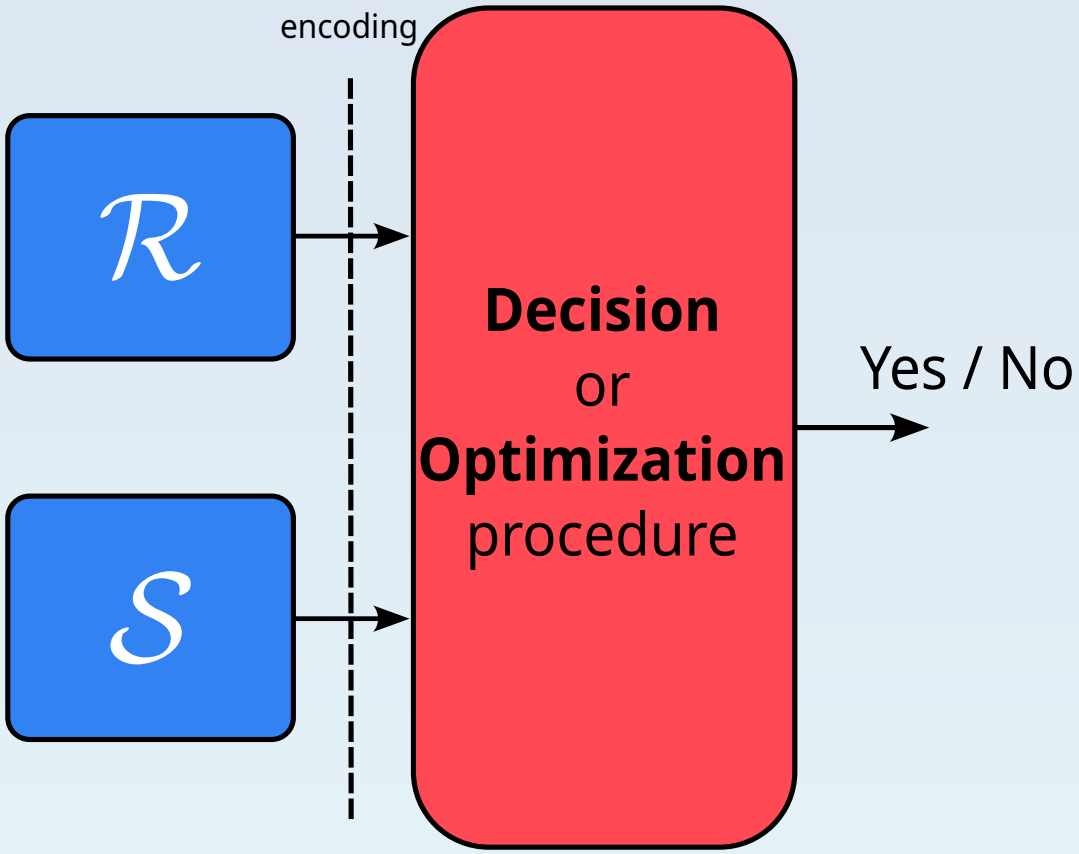
- Highly regulated domain

On trust

Why are we trusting the software running on airplanes?

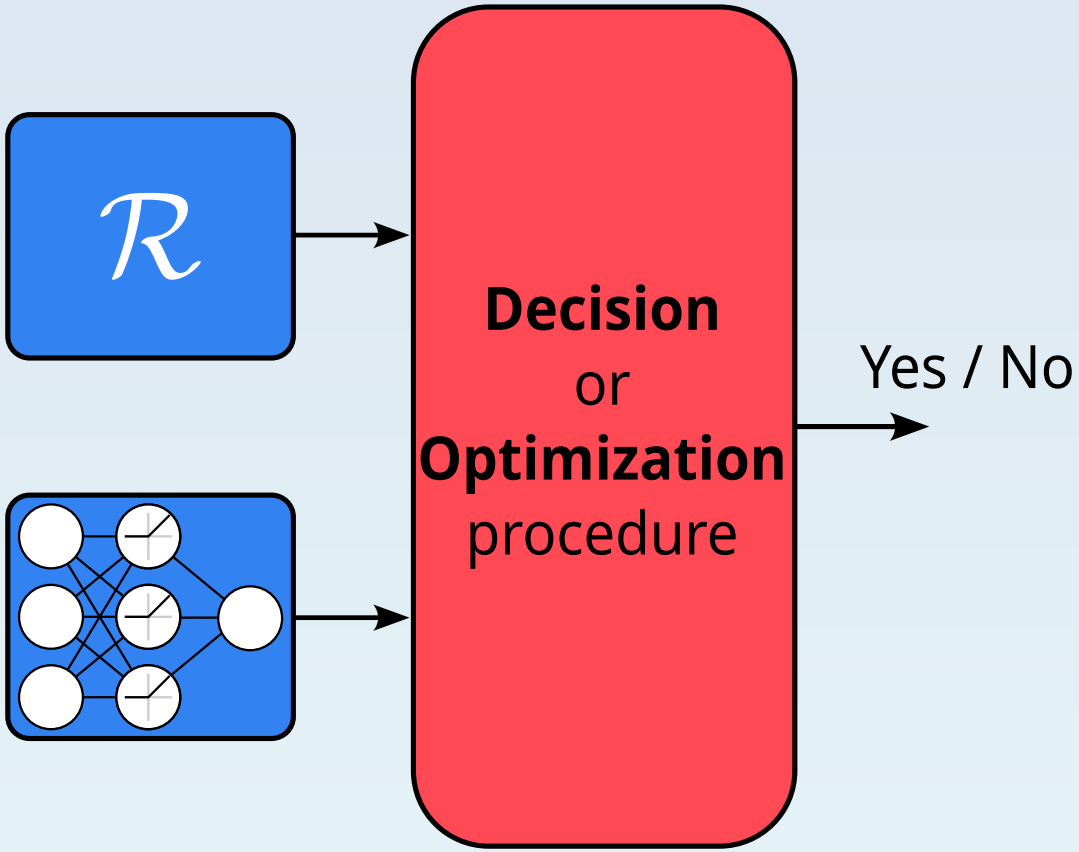
- Highly regulated domain
- We can derive **formal guarantees** on their behaviour

Formal verification

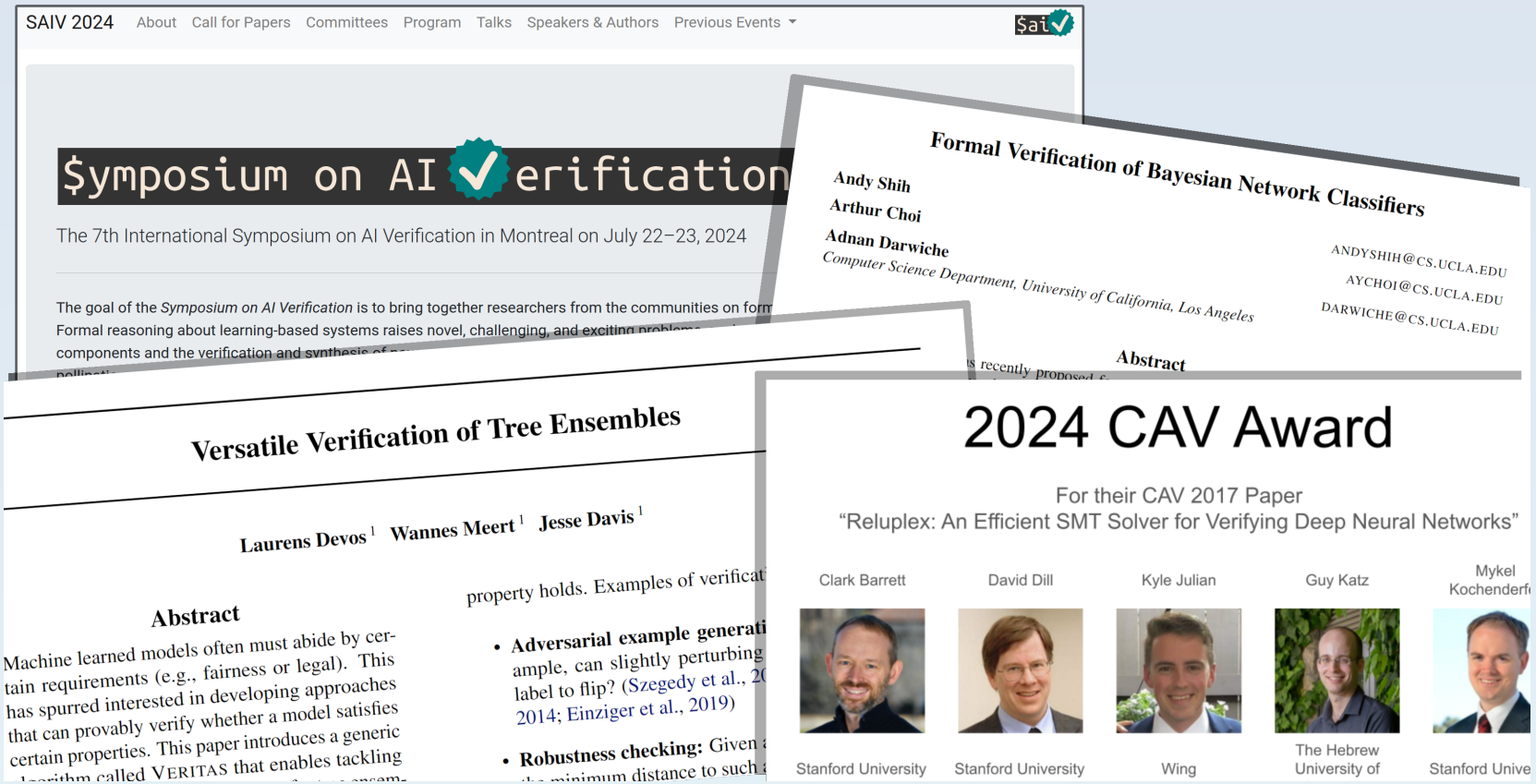


Formal verification

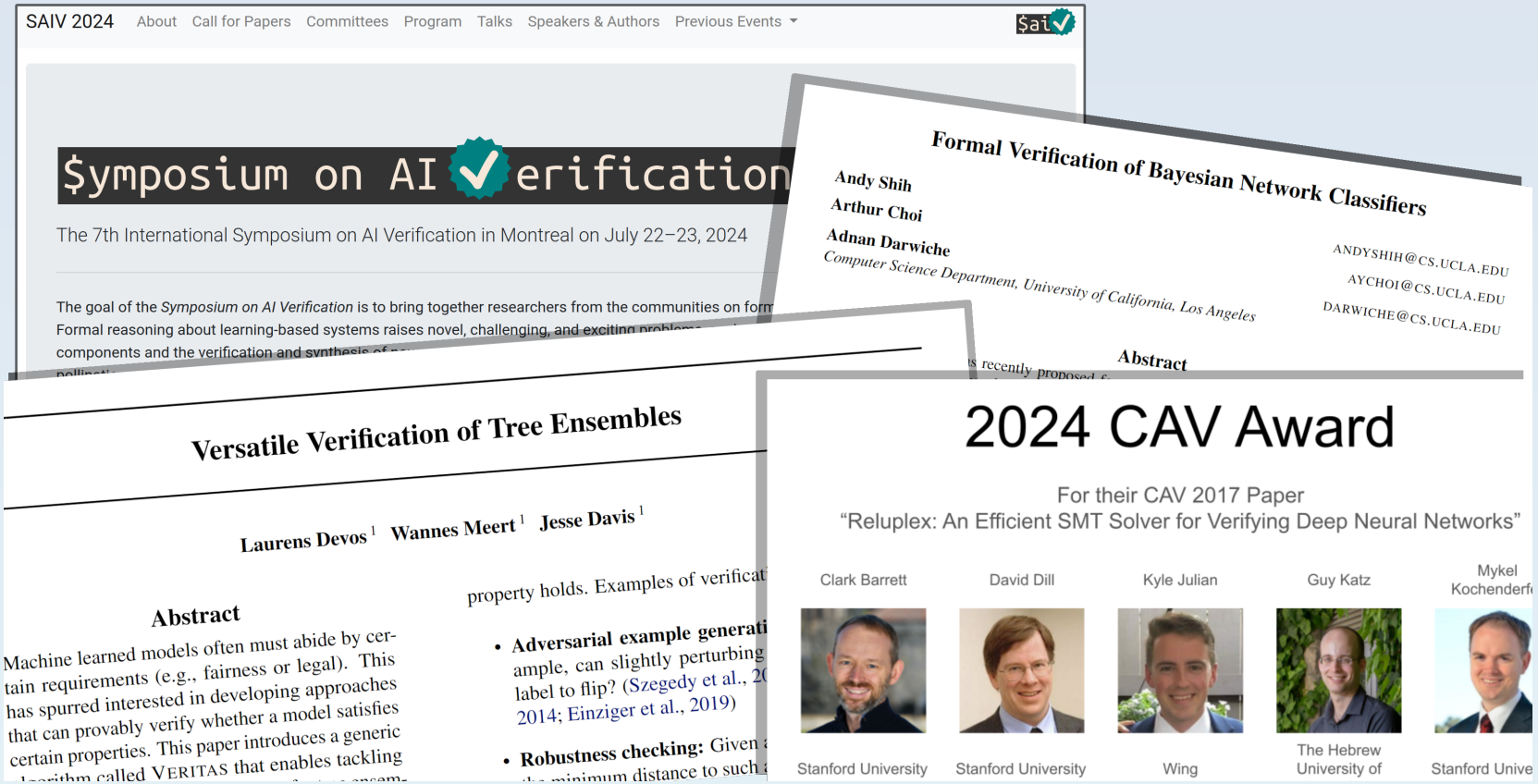
A ML model is software too..



Formal verification

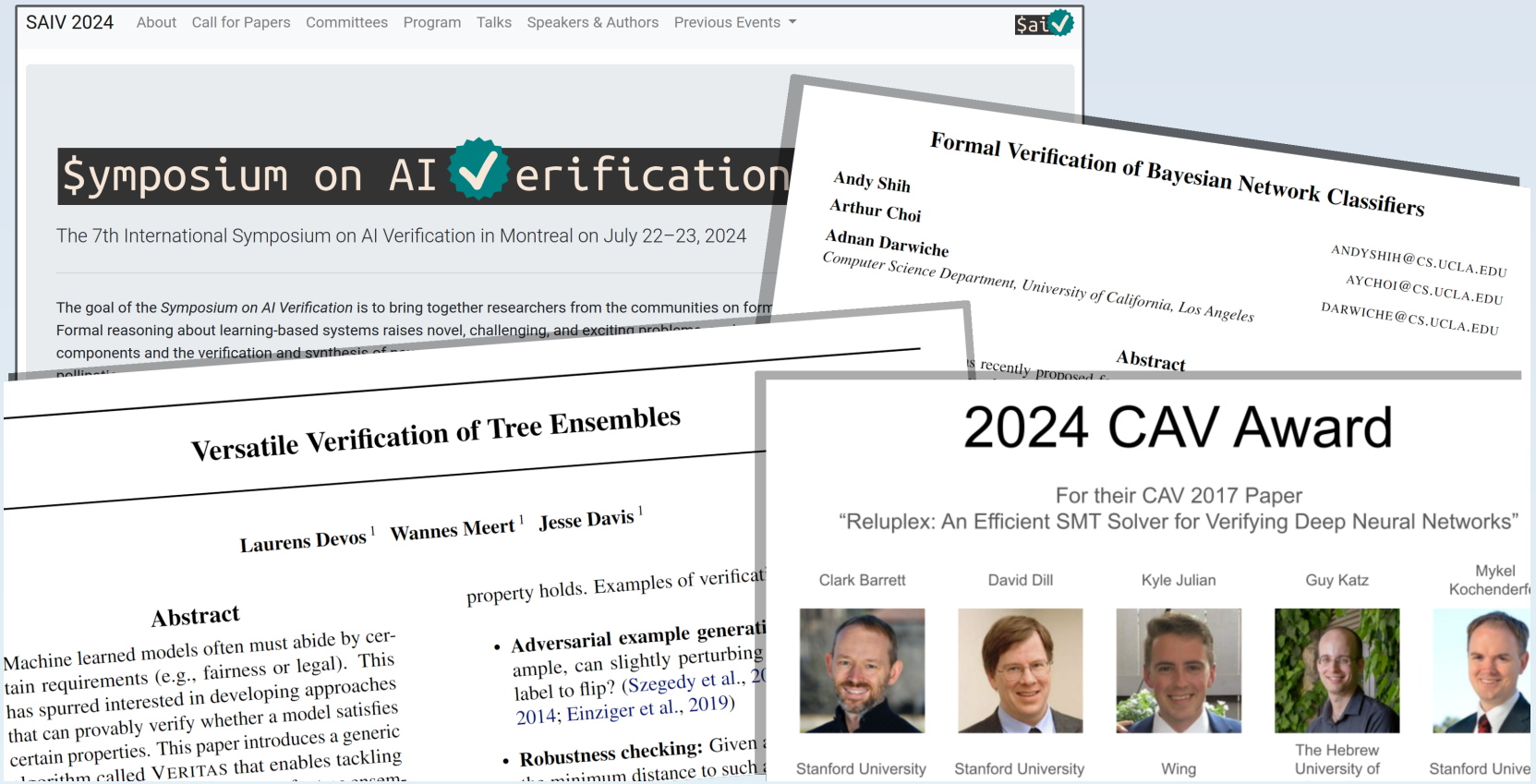


Formal verification



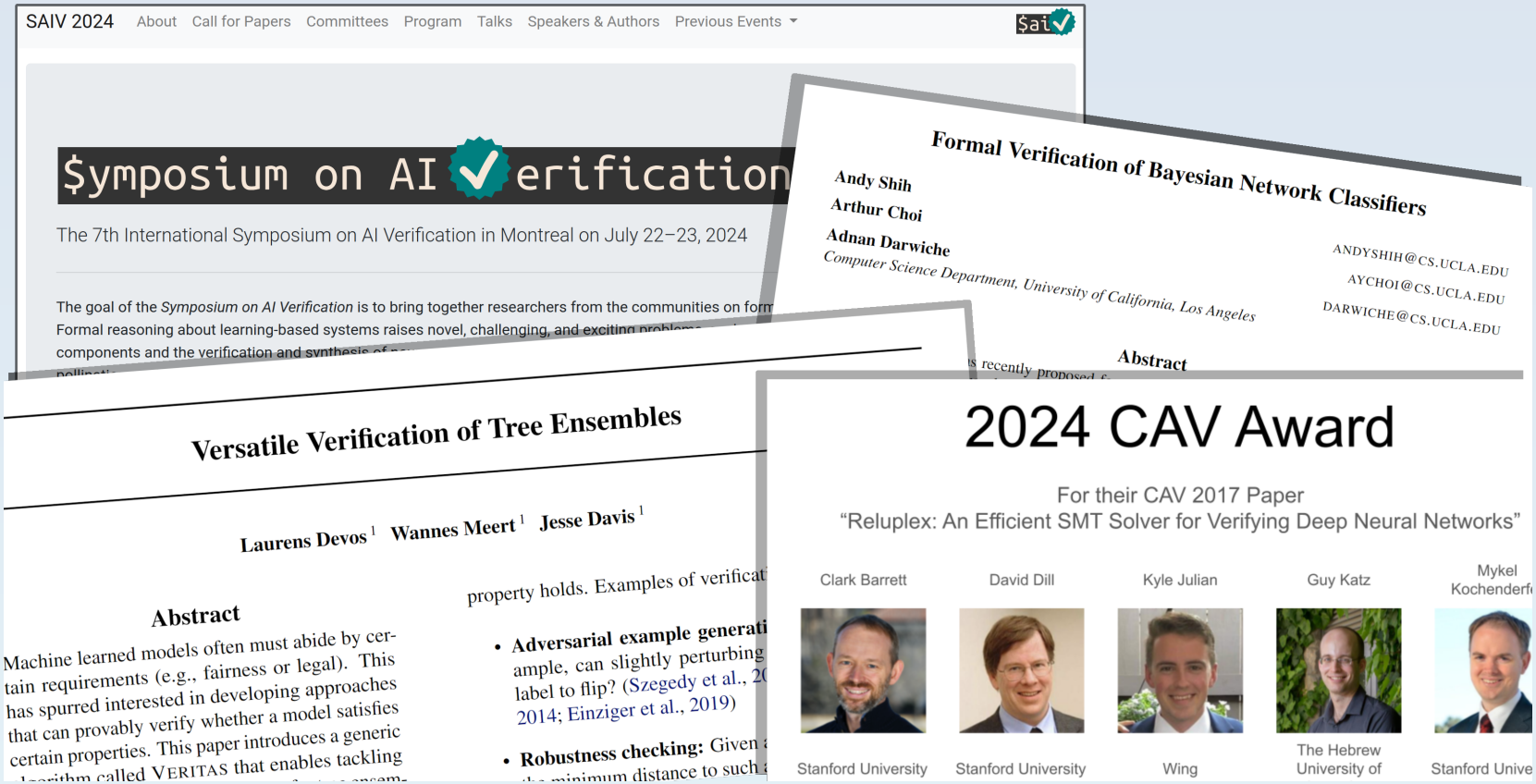
- Vibrant research field at the intersection of ML and FV, however

Formal verification



- Vibrant research field at the intersection of ML and FV, however
- **ad-hoc** approaches for specific model families and properties

Formal verification



- Vibrant research field at the intersection of ML and FV, however
- **ad-hoc** approaches for specific model families and properties
- **deterministic** verification routines.

This project in a nutshell

This project in a nutshell

- A **unified perspective** on ML verification

[D1] **probabilistic**

[D2] model / property **agnostic**, ...

[D3] ... **shared representation** formalism

This project in a nutshell

- A **unified perspective** on ML verification

[D1] **probabilistic**

[D2] model / property **agnostic**, ...

[D3] ... **shared representation** formalism

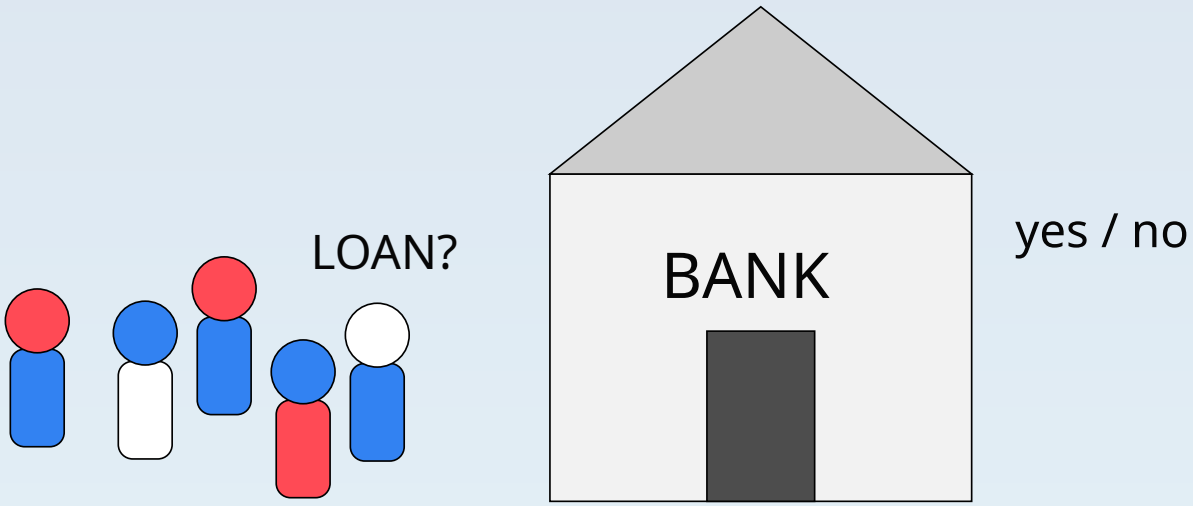
- **Weighted Model Integration** (WMI):
  - **continuous** / **discrete** probabilistic inference framework
  - with **algebraic** / **logical** constraints

Why [D1] probabilistic?

Why [D1] probabilistic?

- Inherently probabilistic properties

E.g. fairness definitions based on population models $P(\mathbf{x})$



Demographic parity: $\dfrac{P(\text{loan} \mid \bullet)}{P(\text{loan} \mid \neg \bullet)} \simeq 1$

Why [D1] probabilistic?

- Inherently probabilistic properties

  E.g. fairness definitions based on population models $P(\mathbf{x})$

- Support for probabilistic models

  GANs, VAEs, Bayesian NNs, PGMs, programs...

Why [D1] probabilistic?

- Inherently probabilistic properties

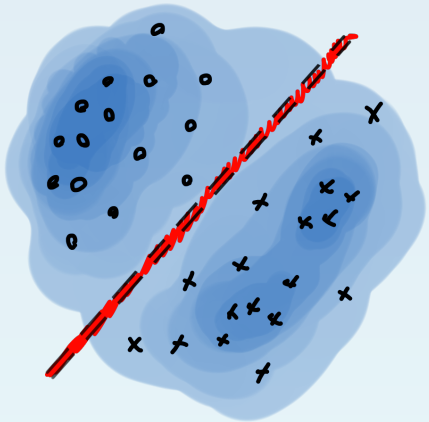
  E.g. fairness definitions based on population models $P(\mathbf{x})$

- Support for probabilistic models

  GANs, VAEs, Bayesian NNs, PGMs, programs...

- Hard guarantees difficult to enforce (globally)

$$(d(\mathbf{x}, \mathbf{x}') < k) \implies (f(\mathbf{x}) = f(\mathbf{x}'))$$



Why [D2] property/model-agnostic?

Why [D2] property/model-agnostic?

- Theoretical perspective on probabilistic ML verification
  - Bridging progress in the literature
  - Characterizing the complexity of the PFV tasks

Why [D2] property/model-agnostic?

- Theoretical perspective on probabilistic ML verification
  - Bridging progress in the literature
  - Characterizing the complexity of the PFV tasks
- An interface between:
  - ad-hoc algorithms
  - general-purpose PFV tools

Why [D3] shared a representation?

Why [D3] shared a representation?

Certified ML models as part of the requirement

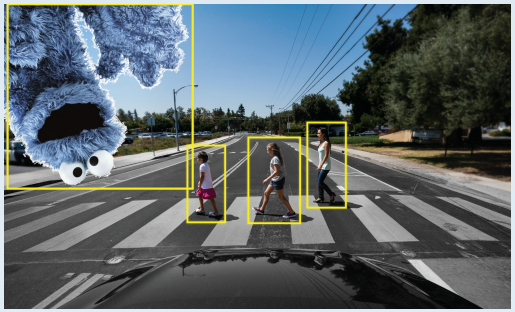
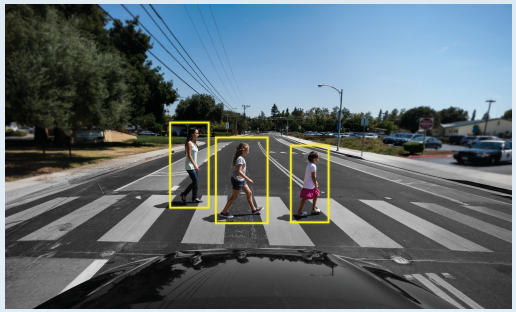
sometimes hard to define manually (e.g. in computer vision)

Why [D3] shared a representation?

Certified ML models as part of the requirement

sometimes hard to define manually (e.g. in computer vision)

- Complex **priors** $P(\mathbf{x})$:

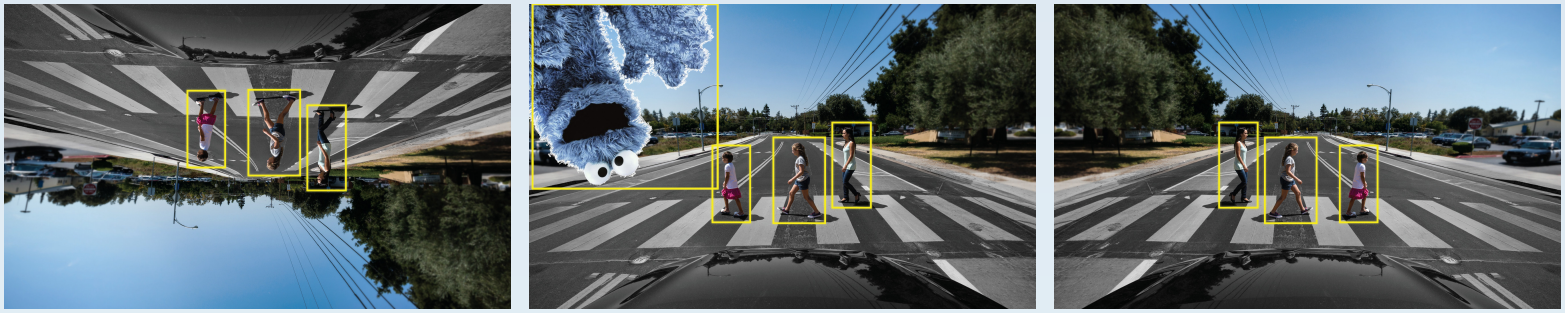




Why [D3] shared a representation?

Certified ML models as part of the requirement

sometimes hard to define manually (e.g. in computer vision)

- Complex **priors** $P(\mathbf{x})$:



- High-level **properties**:

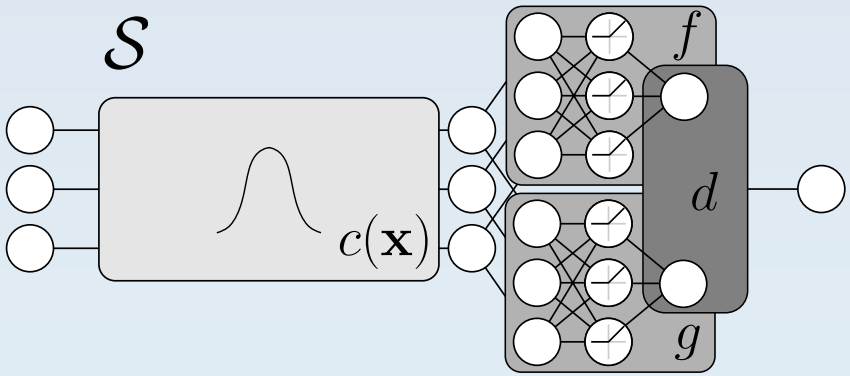
$$\text{pedestrian}(\boxed{}) \implies (f(\boxed{}) = \text{decelerate})$$

The task

The task

Given a **system** $\mathcal{S}$:

$$\mathbf{y} \sim P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$$

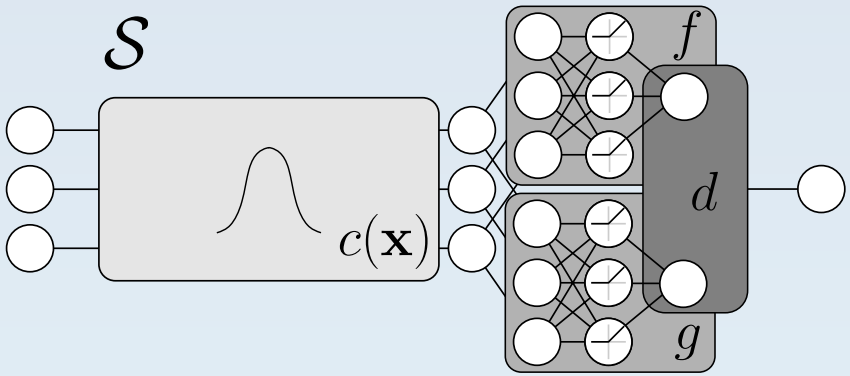


The task

Given a **system** $\mathcal{S}$:

$$\mathbf{y} \sim P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$$

and a **requirement** $\mathcal{R}$:



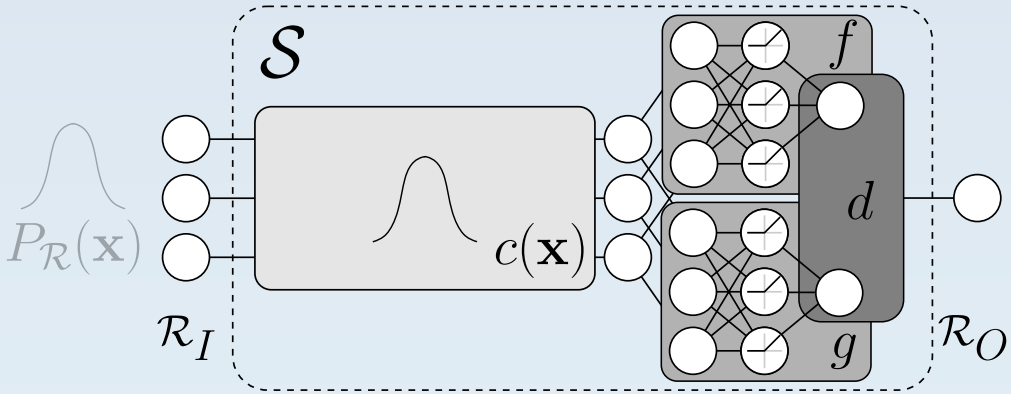
The task

Given a **system** $\mathcal{S}$:

$$\mathbf{y} \sim P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$$

and a **requirement** $\mathcal{R}$:

- a *precondition* $\mathcal{R}_I$
- a *postcondition* $\mathcal{R}_O$



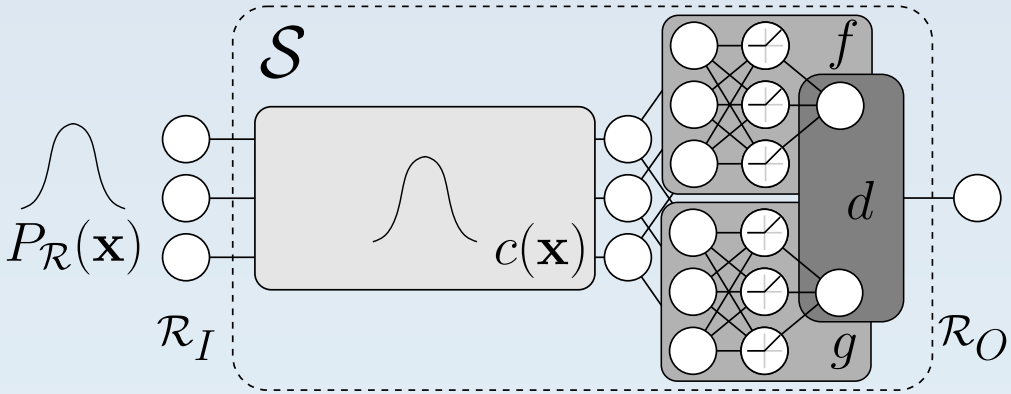
The task

Given a **system** $\mathcal{S}$:

$$\mathbf{y} \sim P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$$

and a **requirement** $\mathcal{R}$:

- a *precondition* $\mathcal{R}_I$
- a *postcondition* $\mathcal{R}_O$
- a *prior* $P_{\mathcal{R}}(\mathbf{x})$



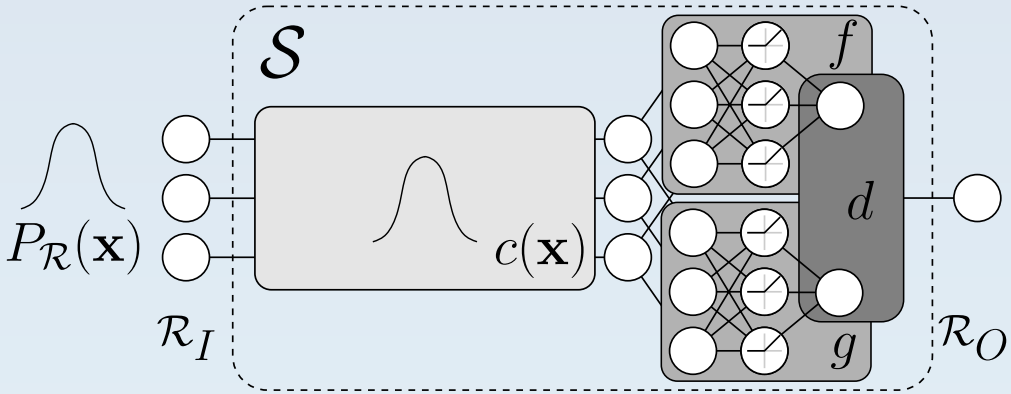
The task

Given a **system** $\mathcal{S}$:

$$\mathbf{y} \sim P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$$

and a **requirement** $\mathcal{R}$:

- a *precondition* $\mathcal{R}_I$
- a *postcondition* $\mathcal{R}_O$
- a *prior* $P_{\mathcal{R}}(\mathbf{x})$



Compute (for $k \in [0, 1]$):

$$P(\mathcal{R}_O \mid \mathcal{R}_I) > k \ ?$$

with $\quad \mathbf{x}, \mathbf{y} \sim P(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} P_{\mathcal{R}}(\mathbf{x}) \cdot P_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x})$

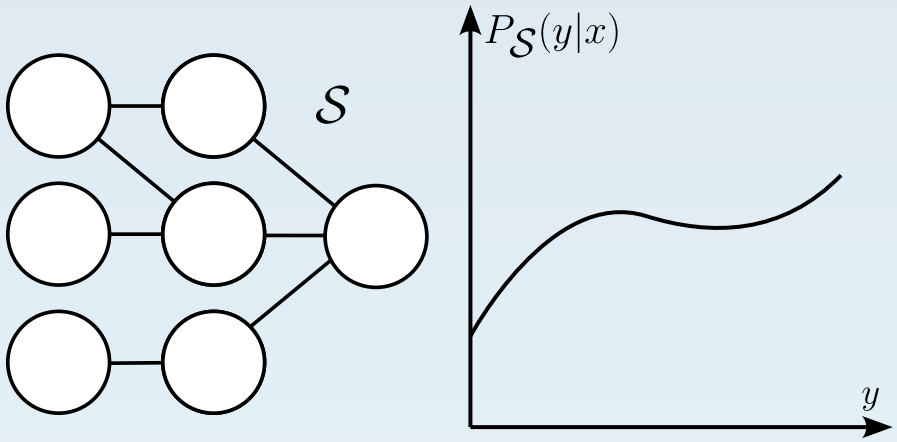
The task

In other words..

The task

In other words..

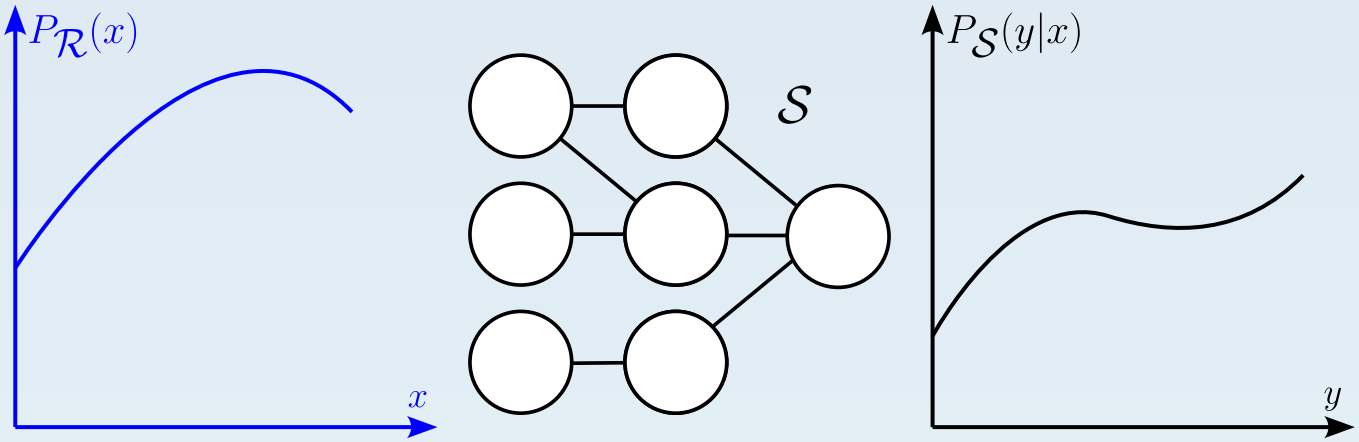
- Is our probabilistic **system**



The task

In other words..

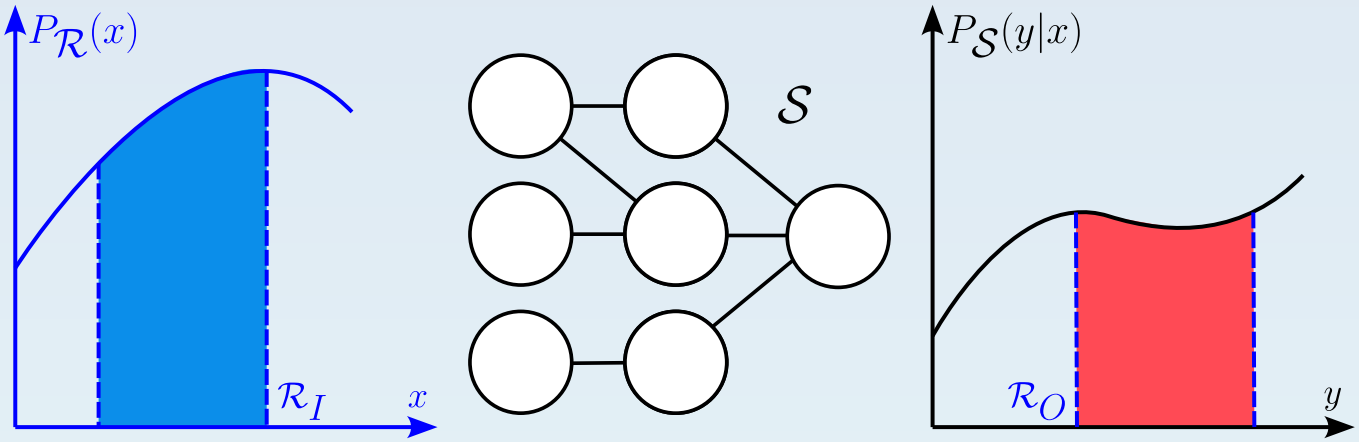
- Is our probabilistic **system**
- once deployed in an uncertain environment



The task

In other words..

- Is our probabilistic **system**
- once deployed in an uncertain environment
- satisfying $\mathcal{R}_O$ with $P > k$ when $\mathcal{R}_I$ holds?

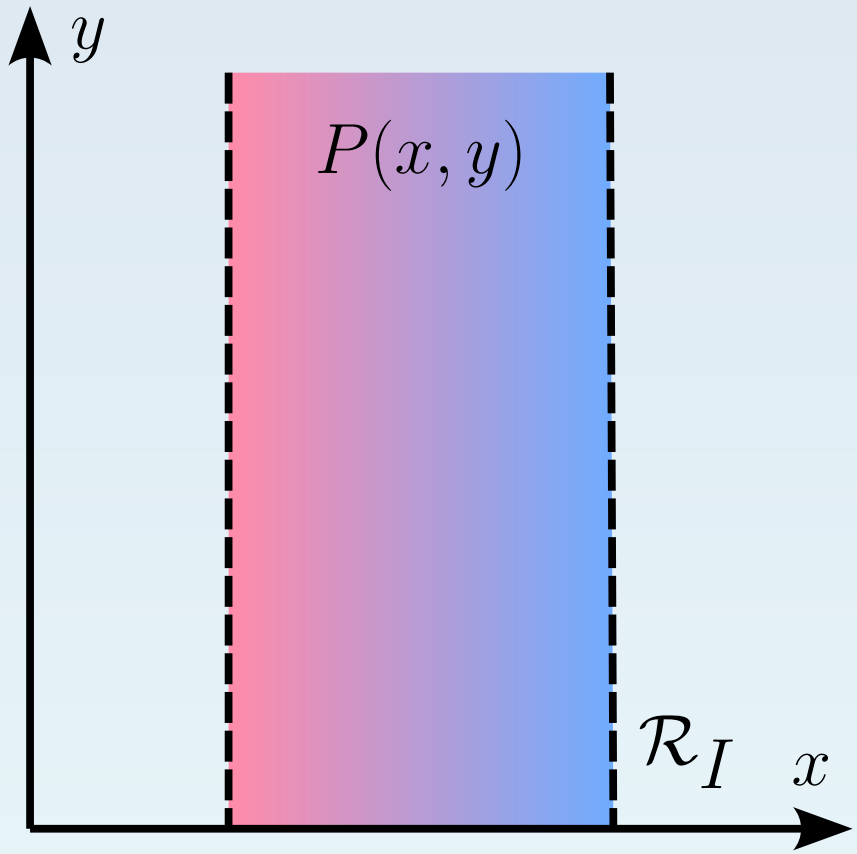


The task

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) =$$

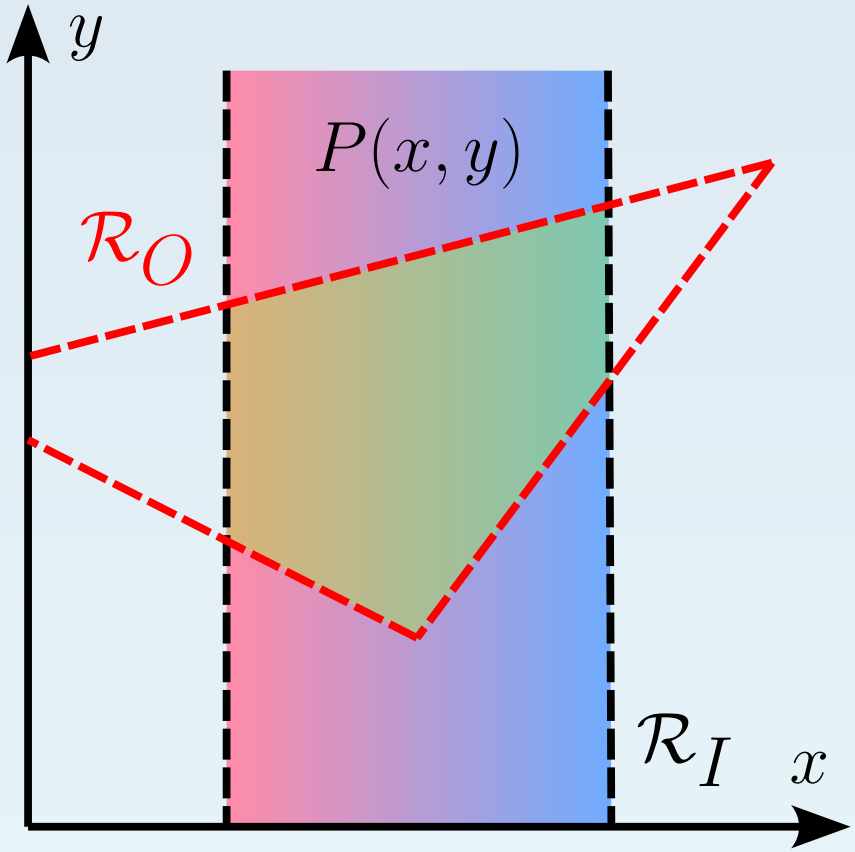
The task

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{}{P(\mathcal{R}_I \mid \mathcal{S})}$$



The task

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$



How do we compute **that**..?

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

$$\#SAT(A \vee B) = 3$$

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

$$\#SAT(A \vee B) = 3$$

- **WMC**: **weighted** generalization of #SAT:

$$WMC(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} m(\mu, w) =$$

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

$$\#SAT(A \vee B) = 3$$

- **WMC**: **weighted** generalization of #SAT:

$$WMC(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} m(\mu, w) = \sum_{\mu \models \Delta} \prod_{\ell \in \mu} w(\ell)$$

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

$$\#SAT(A \vee B) = 3$$

- **WMC**: **weighted** generalization of #SAT:

$$WMC(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} m(\mu, w) = \sum_{\mu \models \Delta} \prod_{\ell \in \mu} w(\ell)$$

$$w(A) = 3,\ w(B) = 2,\ w(\ell) = 1$$

Weighted Model Counting (WMC)

- **#SAT**: counting variant of SAT

$$\#SAT(\Delta) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} 1$$

$$\#SAT(A \vee B) = 3$$

- **WMC**: **weighted** generalization of #SAT:

$$WMC(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta} m(\mu, w) = \sum_{\mu \models \Delta} \prod_{\ell \in \mu} w(\ell)$$

$$w(A) = 3,\ w(B) = 2,\ w(\ell) = 1$$

$$WMC(A \vee B, w) = \underbrace{3 \cdot 2}_{A \wedge B} + \underbrace{3 \cdot 1}_{A \wedge \neg B} + \underbrace{1 \cdot 2}_{\neg A \wedge B} = 11$$

Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

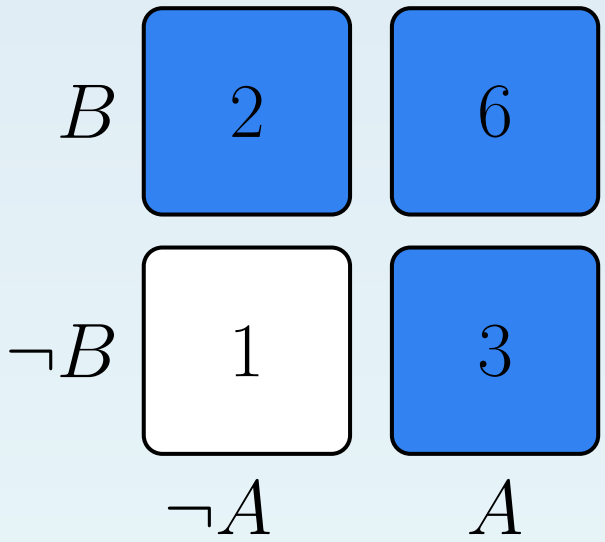
$$P(A \mid A \vee B) = \frac{WMC(A \wedge (A \vee B), w)}{WMC(A \vee B, w)} =$$

Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

$$P(A \mid A \vee B) = \frac{WMC(A \wedge (A \vee B), w)}{WMC(A \vee B, w)} = \frac{}{11}$$

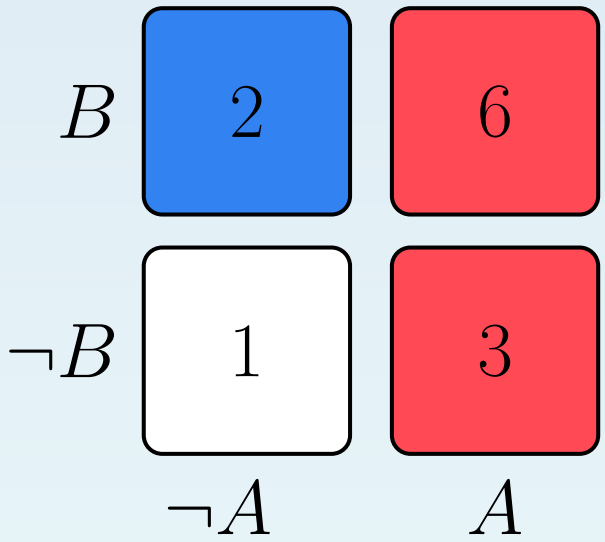


Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

$$P(A \mid A \vee B) = \frac{WMC(A \wedge (A \vee B), w)}{WMC(A \vee B, w)} = \frac{9}{11}$$



Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

- arbitrary formulas Δ, Γ

Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

- arbitrary formulas Δ, Γ
- arbitrary non-negative functions w

$$Z = WMC(\top, w)$$

Weighted Model Counting (WMC)

- WMC enables **probabilistic inference** over logical domains

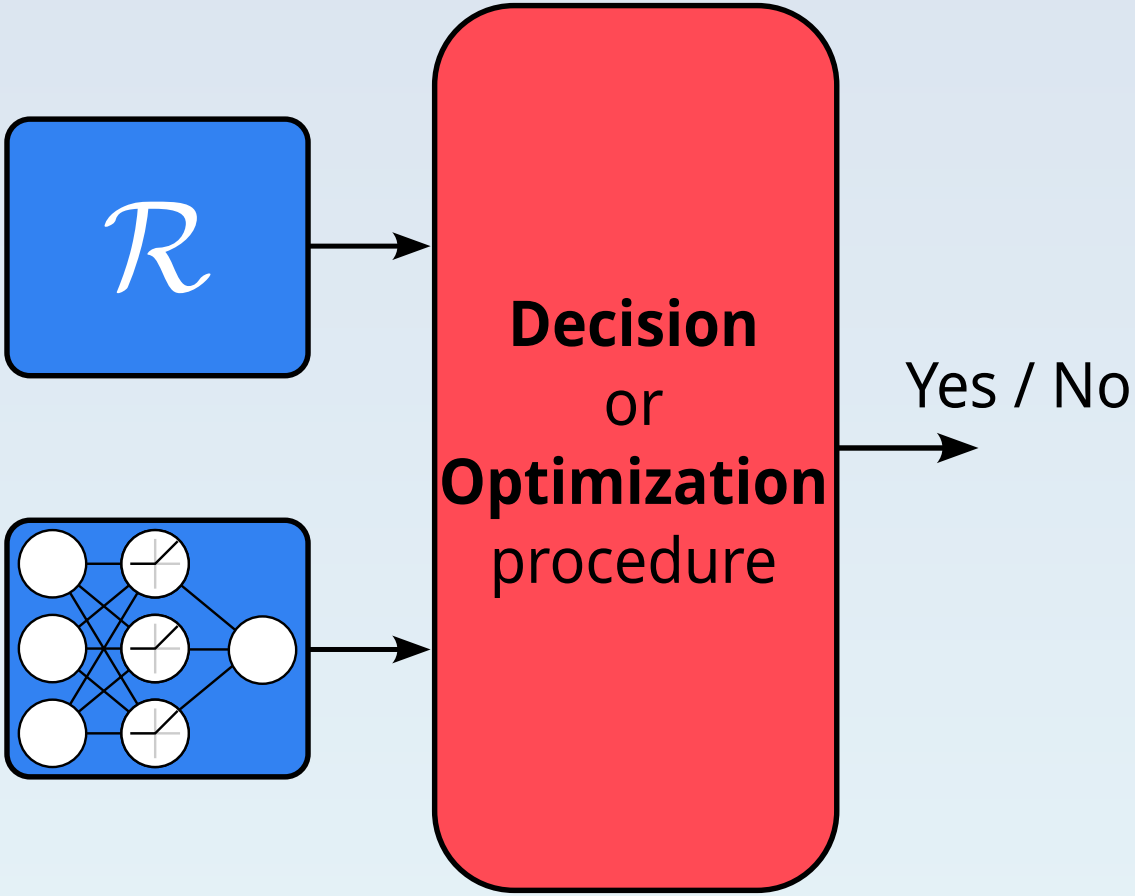
$$P(\Delta \mid \Gamma; w) = \frac{WMC(\Delta \wedge \Gamma, w)}{WMC(\Gamma, w)}$$

- arbitrary formulas Δ, Γ
- arbitrary non-negative functions w

$$Z = WMC(\top, w)$$

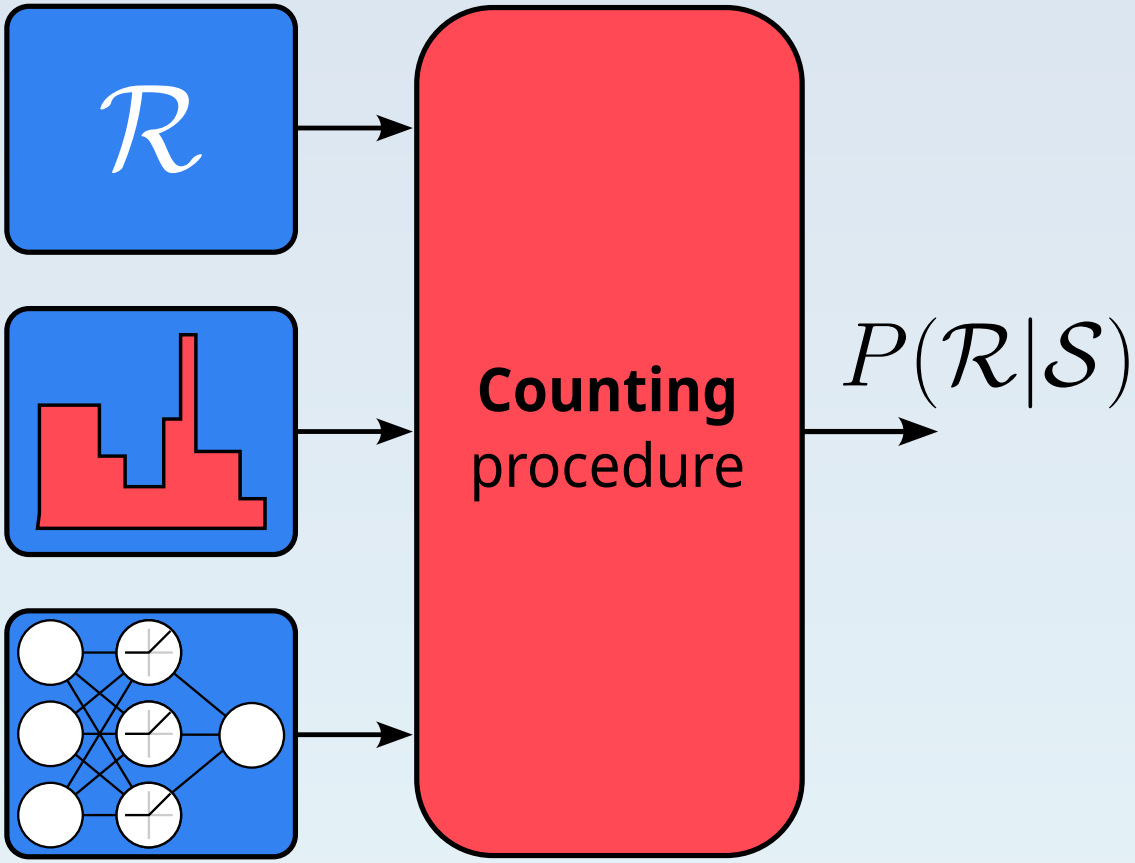
SOTA technique in PGMs, probabilistic logic programs and DBs, etc.

Counting-based verification



"Quantitative verification of NNs and its security applications" [Baluta et al. 2019]

Counting-based verification



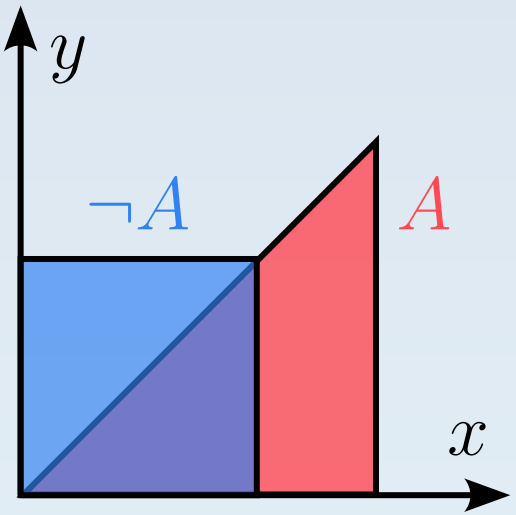
"Quantitative verification of NNs and its security applications" [Baluta et al. 2019]

WMC $\rightarrow$ **Weighted Model Integration**

WMC $\rightarrow$ **Weighted Model Integration**

- Propositional logic $\rightarrow$ **Satisfiability Modulo LRA** (SMT-LRA)

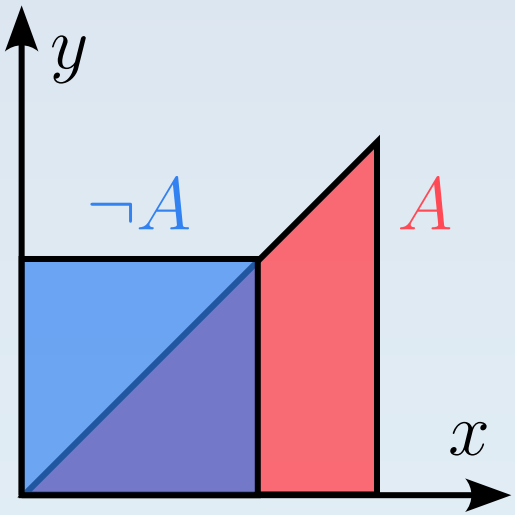
ex. $\chi = (0 \leq x) \wedge (A \rightarrow ((y \leq x) \wedge (x \leq 3)))$
$\wedge\, (0 \leq y) \wedge (\neg A \rightarrow ((x \leq 2) \wedge (y \leq 2)))$



WMC $\rightarrow$ **Weighted Model Integration**

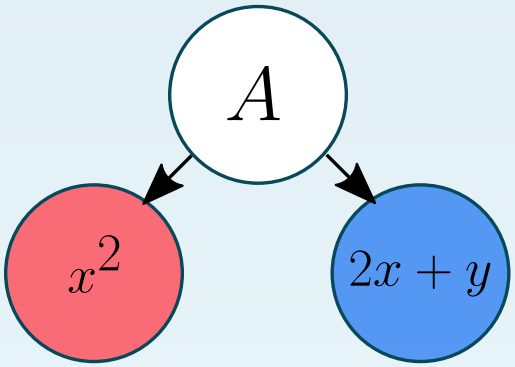
- Propositional logic $\rightarrow$ **Satisfiability Modulo LRA** (SMT-LRA)

ex. $\chi = (0 \leq x) \wedge (A \rightarrow ((y \leq x) \wedge (x \leq 3))) \wedge (0 \leq y) \wedge (\neg A \rightarrow ((x \leq 2) \wedge (y \leq 2)))$



- Discrete w $\rightarrow$ **Density functions**

ex. $w(x, y, A) = \begin{cases} x^2 & \text{if } A \\ 2x + y & \text{if } \neg A \end{cases}$



Weighted Model Integration (WMI)

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta \wedge \chi} m(\mu, w)$$

Weighted Model Integration (WMI)

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta \wedge \chi} m(\mu, w)$$

- χ: explicit notion of **support** of w

Weighted Model Integration (WMI)

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta \wedge \chi} m(\mu, w)$$

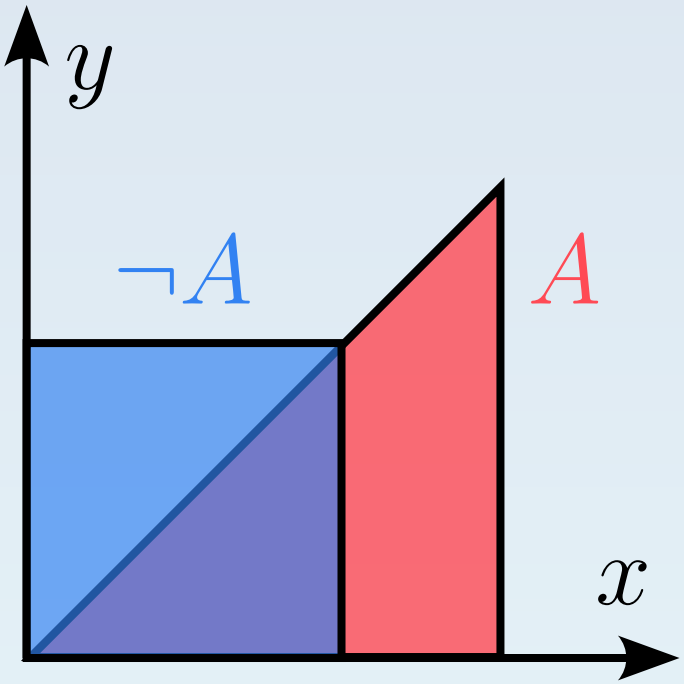
- χ: explicit notion of **support** of w
- μ: conjunction of linear inequalities $\rightarrow$ **convex polytopes**

Weighted Model Integration (WMI)

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \sum_{\mu \models \Delta \wedge \chi} m(\mu, w)$$
$$= \sum_{\mu \models \Delta \wedge \chi} \int_{\mu} w(\mathbf{x}) d\mathbf{x}$$

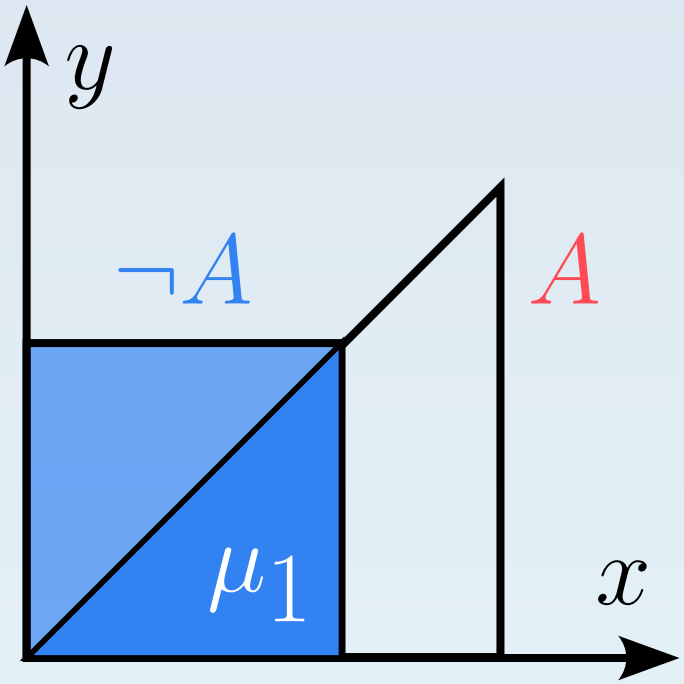
- χ: explicit notion of **support** of w
- μ: conjunction of linear inequalities $\rightarrow$ **convex polytopes**
- Computing the **probability mass** of μ: integration of w in μ

Weighted Model Integration (WMI)



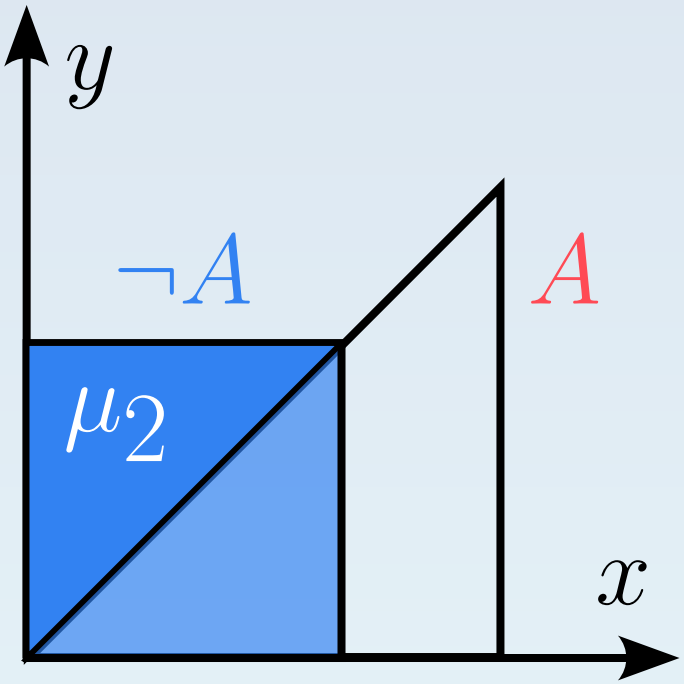
$$Z = WMI(\top, w)$$

Weighted Model Integration (WMI)



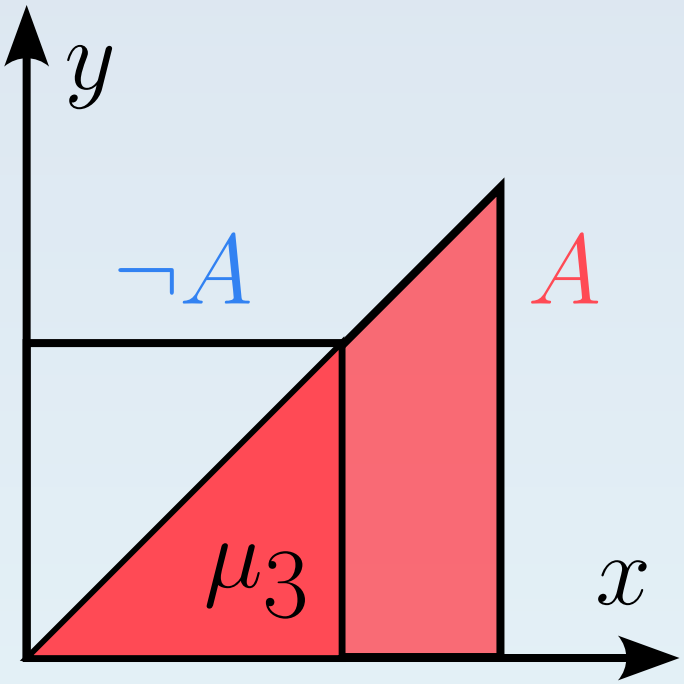
$$Z = WMI(\top, w)$$
$$= \int_0^x \int_0^2 2x + y \; dx \; dy$$

Weighted Model Integration (WMI)



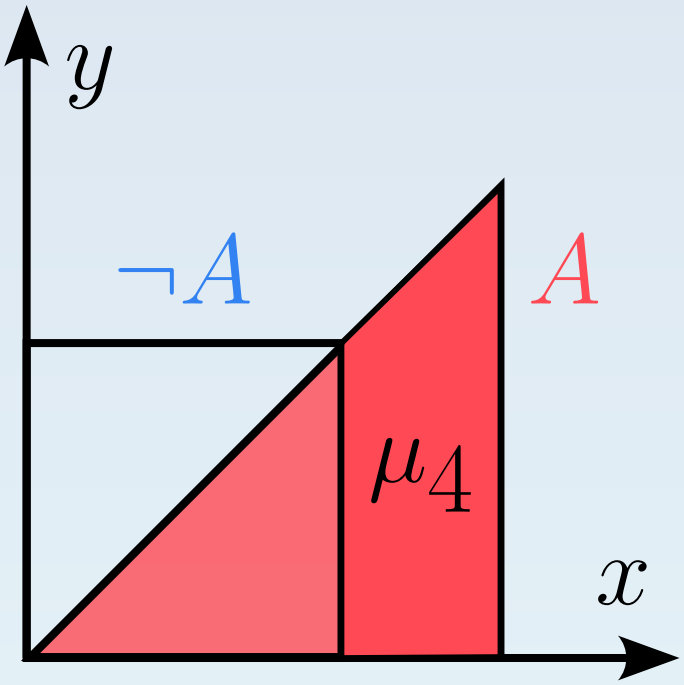
$$
\begin{aligned}
Z &= WMI(\top, w) \\
&= \int_0^x \int_0^2 2x + y \; dx \; dy \\
&+ \int_x^2 \int_0^2 2x + y \; dx \; dy
\end{aligned}
$$

Weighted Model Integration (WMI)



$$
\begin{aligned}
Z &= WMI(\top, w) \\
&= \int_0^x \int_0^2 2x + y \; dx \; dy \\
&+ \int_x^2 \int_0^2 2x + y \; dx \; dy \\
&+ \int_0^x \int_0^2 x^2 \; dx \; dy
\end{aligned}
$$

Weighted Model Integration (WMI)



$$
\begin{aligned}
Z &= WMI(\top, w) \\
&= \int_0^x \int_0^2 2x + y \; dx \; dy \\
&+ \int_x^2 \int_0^2 2x + y \; dx \; dy \\
&+ \int_0^x \int_0^2 x^2 \; dx \; dy \\
&+ \int_0^x \int_2^3 x^2 \; dx \; dy
\end{aligned}
$$

Weighted Model Integration (WMI)

$$P(\Delta \mid \Gamma) = \frac{WMI(\Delta \wedge \Gamma, w)}{WMI(\Gamma, w)}$$

Weighted Model Integration (WMI)

$$P(\Delta \mid \Gamma) = \frac{WMI(\Delta \wedge \Gamma, w)}{WMI(\Gamma, w)}$$

- SOTA inference with algebraic/logical constraints

Weighted Model Integration (WMI)

$$P(\Delta \mid \Gamma) = \frac{WMI(\Delta \wedge \Gamma, w)}{WMI(\Gamma, w)}$$

- SOTA inference with algebraic/logical constraints
- Large number of different solving paradigms:
  - Partial SMT enumeration
  - Knowledge compilation
  - Hashing-based approximations
  - ...

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

$$= \frac{\int [\![\mathcal{R}_I \wedge \mathcal{R}_O]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}{\int [\![\mathcal{R}_I]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}$$

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

$$= \frac{\int [\![\mathcal{R}_I \wedge \mathcal{R}_O]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}{\int [\![\mathcal{R}_I]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}$$

$$= \frac{WMI(\Delta_I \wedge \Delta_O, w)}{WMI(\Delta_I, w)}$$

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

$$= \frac{\int [\![\mathcal{R}_I \wedge \mathcal{R}_O]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}{\int [\![\mathcal{R}_I]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}$$

$$= \frac{WMI(\Delta_I \wedge \Delta_O, w)}{WMI(\Delta_I, w)}$$

where:

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

$$= \frac{\int [\![\mathcal{R}_I \wedge \mathcal{R}_O]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}{\int [\![\mathcal{R}_I]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}$$

$$= \frac{WMI(\Delta_I \wedge \Delta_O, w)}{WMI(\Delta_I, w)}$$

where:

- Δ_I and Δ_O are SMT-LRA encodings of $\mathcal{R}_I$ and $\mathcal{R}_O$

WMI-based verification

$$P(\mathcal{R}_O \mid \mathcal{R}_I, \mathcal{S}) = \frac{P(\mathcal{R}_O, \mathcal{R}_I \mid \mathcal{S})}{P(\mathcal{R}_I \mid \mathcal{S})}$$

$$= \frac{\int [\![\mathcal{R}_I \wedge \mathcal{R}_O]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}{\int [\![\mathcal{R}_I]\!] \quad P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})\, d\mathbf{x}\, d\mathbf{y}}$$

$$= \frac{WMI(\Delta_I \wedge \Delta_O, w)}{WMI(\Delta_I, w)}$$

where:

- Δ_I and Δ_O are SMT-LRA encodings of $\mathcal{R}_I$ and $\mathcal{R}_O$
- $w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{S}}(\mathbf{y} \mid \mathbf{x}) \cdot w_{\mathcal{R}}(\mathbf{x})$ encodes $P_{\mathcal{S}}(\mathbf{y}|\mathbf{x}) \cdot P_{\mathcal{R}}(\mathbf{x})$

Systems and environments

Probabilistic modelling

Probabilistic modelling

- **Building blocks**: w **non-negative**, **integrable** over polytopes

Polynomials

- *closed* under $+, \cdot, \int_\Delta$
- *arbitrary* approximation

- $d = 0$: most common
- $d > 0$: polytime $\int$ in d

Probabilistic modelling

- **Building blocks**: w **non-negative**, **integrable** over polytopes

Polynomials

- *closed* under $+, \cdot, \int_{\Delta}$
- *arbitrary* approximation

- $d = 0$: most common
- $d > 0$: polytime $\int$ in d

- **Structured** by means of:
  - Piecewise decomposition ($w \stackrel{\text{def}}{=} \text{ITE}(\Delta;\ w_1;\ w_2)$)
  - Sums and products

Probabilistic modelling

- **Building blocks**: w **non-negative**, **integrable** over polytopes

  Polynomials

  - *closed* under $+, \cdot, \int_{\Delta}$
  - *arbitrary* approximation

  - $d = 0$: most common
  - $d > 0$: polytime $\int$ in d

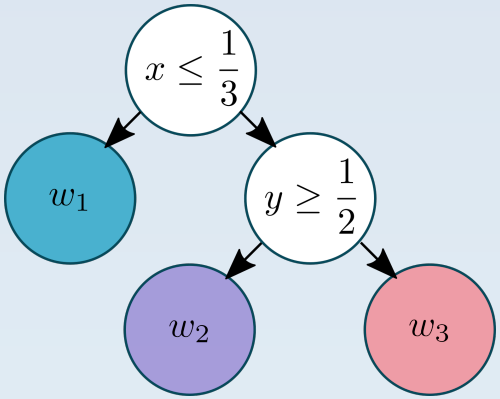
- **Structured** by means of:

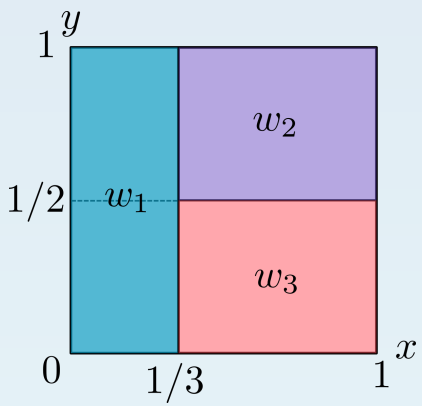
  - Piecewise decomposition ($w \stackrel{\text{def}}{=} \text{ITE}(\Delta;\ w_1;\ w_2)$)
  - Sums and products

- Any determinism / **hard constraints** encoded in χ

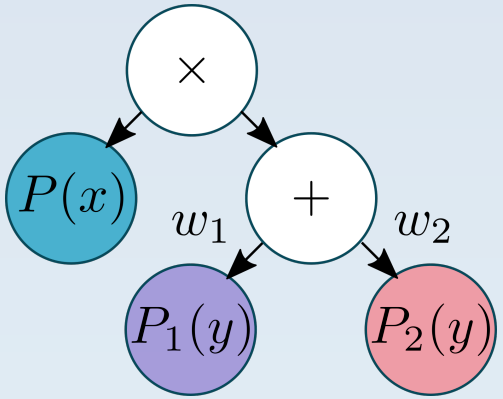
Probabilistic modelling

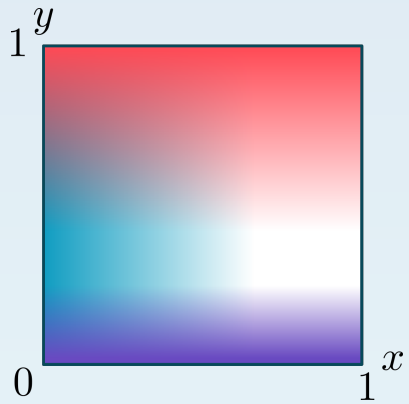
- *Density Estimation Trees*





- *Mixed Sum-Product Networks*





- *Hybrid Bayesian/Markov Nets, cProbLog, ...*

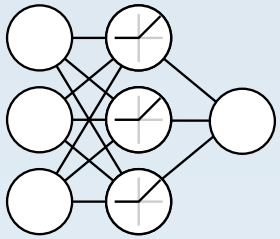
Deterministic modelling

- *Tree-based predictors*

$$\chi_N \stackrel{\text{def}}{=} \mathsf{ITE}(\Delta_{cond};\ \chi_{N_\top};\ \chi_{N_\bot}) \qquad \chi_L \stackrel{\text{def}}{=} (\mathbf{y} = f_L(\mathbf{x}))$$

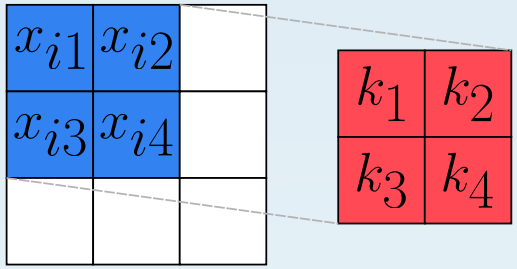
- *Rectified Linear Units*

$$\chi_U \stackrel{\text{def}}{=} (l_U = \mathbf{W}_U \cdot \mathbf{x}_U + b_U) \wedge \mathsf{ITE}(l_U > 0;\ y_U = l_U;\ y_U = 0)$$



- *Convolution / pooling*

$$\chi_i \stackrel{\text{def}}{=} \begin{cases} \sum_{j=1}^{K} k_j x_{ij} & \text{(convolution)} \\ \max\{x_{ij}\}_{j=1}^{K} & \text{(max. pooling)} \\ 1/K \sum_{j=1}^{K} x_{ij} & \text{(avg. pooling)} \end{cases}$$



- *Ensembles of models* $\mathcal{S}_1, ..., \mathcal{S}_K$

$$\chi_{\mathcal{S}} \stackrel{\text{def}}{=} (\mathbf{y} = a(\mathbf{y}_1, ..., \mathbf{y}_K)) \wedge \bigwedge_{i=1}^{K} \chi_{\mathcal{S}_i}(\mathbf{x}, \mathbf{y}_i)$$

- random forests / XGBoost
- CBMs

Properties

Preliminaries

Preliminaries

- ML output:
  - $f(\mathbf{x})$ the output of a regressor
  - $c(\mathbf{x})$ the (positive) output of a binary classifier

Preliminaries

- ML output:
  - $f(\mathbf{x})$ the output of a regressor
  - $c(\mathbf{x})$ the (positive) output of a binary classifier

- Distances:
  - $\|\mathbf{x} - \mathbf{x}'\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^{N} |x_i - x'_i|$
  - $\|\mathbf{x} - \mathbf{x}'\|_\infty \stackrel{\text{def}}{=} \max_{i=1}^{N} |x_i - x'_i|$

  where

$$|v_1 - v_2| \stackrel{\text{def}}{=} \text{ITE}(v_1 < v_2;\ v_2 - v_1;\ v_1 - v_2)$$

$$\max(\{v_1, v_2\}) \stackrel{\text{def}}{=} \text{ITE}(v_1 < v_2;\ v_2;\ v_1)$$

$$\max(\{v_1, v_2\} \cup V) \stackrel{\text{def}}{=} \text{ITE}(v_1 < v_2;\ \max(\{v_2\} \cup V);\ \max(\{v_1\} \cup V))$$

Examples

Examples

- ϵ-*local robustness* around $\mathbf{x}_0$

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}_0\| \leq \epsilon$$

$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c_0)$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y})$$

Examples

- ϵ-*local robustness* around $\mathbf{x}_0$

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}_0\| \leq \epsilon$$
$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c_0)$$
$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y})$$

- *Equivalence* of $\mathcal{S}_1$ and $\mathcal{S}_2$

$$\Delta_I \stackrel{\text{def}}{=} \top$$
$$\Delta_O \stackrel{\text{def}}{=} (c_{\mathcal{S}_1}(\mathbf{x}) \Leftrightarrow c_{\mathcal{S}_2}(\mathbf{x}))$$
$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}_1}(\mathbf{x}, \mathbf{y}) \cdot w_{\mathcal{S}_2}(\mathbf{x}, \mathbf{y})$$

Examples

- ϵ-*local robustness* around $\mathbf{x}_0$

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}_0\| \leq \epsilon$$
$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c_0)$$
$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y})$$

- *Equivalence* of $\mathcal{S}_1$ and $\mathcal{S}_2$

$$\Delta_I \stackrel{\text{def}}{=} \top$$
$$\Delta_O \stackrel{\text{def}}{=} (c_{\mathcal{S}_1}(\mathbf{x}) \Leftrightarrow c_{\mathcal{S}_2}(\mathbf{x}))$$
$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}_1}(\mathbf{x}, \mathbf{y}) \cdot w_{\mathcal{S}_2}(\mathbf{x}, \mathbf{y})$$

- *Demographic parity* $P(c(\mathbf{x}) \mid \mathbf{x} \in \mathcal{M}) / P(c(\mathbf{x}) \mid \mathbf{x} \notin \mathcal{M})$

$$\Delta_I \stackrel{\text{def}}{=} [\neg]\Delta_{\mathcal{M}}$$
$$\Delta_O \stackrel{\text{def}}{=} c(\mathbf{x})$$
$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y})$$

Modelling *hyper-properties*

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

where $[v \leftarrow v']$ substitutes all variable with fresh ones.

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

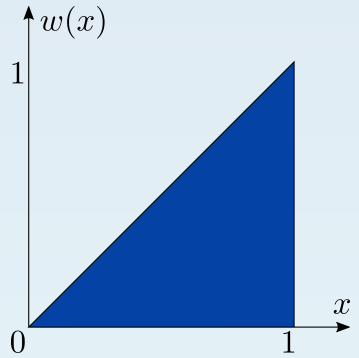
$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

  where $[v \leftarrow v']$ substitutes all variable with fresh ones.

- Example:



$$\chi = (x \in [0, 1]) \qquad w(x) = x$$

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

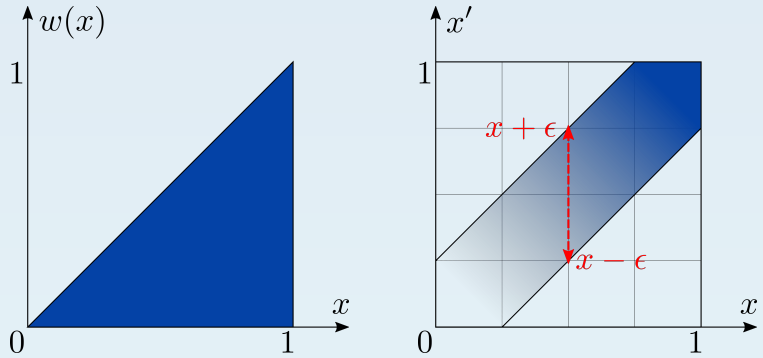
$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

where $[v \leftarrow v']$ substitutes all variable with fresh ones.

- Example:



$$\chi = (x \in [0,1]) \wedge (x' \in [0,1]) \qquad w(x, x') = x \cdot x'$$

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

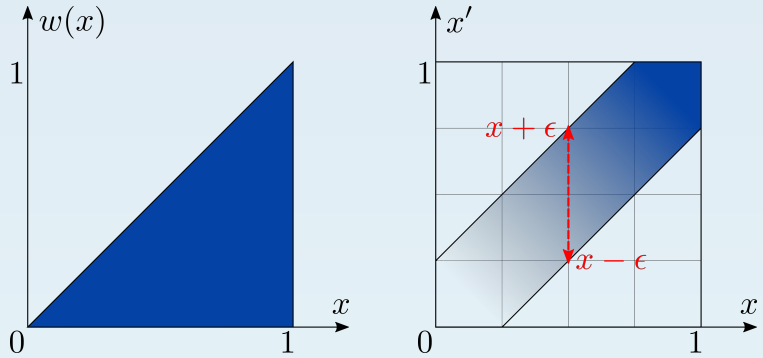
$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

  where $[v \leftarrow v']$ substitutes all variable with fresh ones.

- Example:



$$\chi = (x \in [0, 1]) \wedge (x' \in [0, 1]) \qquad w(x, x') = x \cdot x'$$

"Robustness of NNs: A Probabilistic & Practical Approach" [Mangal et al., 2019]

"Verifying Global Two-Safety Properties in NNs with Confidence" [Athavale et al., 2024]

More examples

More examples

- *Individual fairness*

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}'\| < \epsilon$$

$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathsf{sc}(w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

More examples

- *Individual fairness*

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}'\| < \epsilon$$

$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathsf{sc}(w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

- *Monotonicity*

$$\Delta_I \stackrel{\text{def}}{=} (\mathbf{x} < \mathbf{x}')$$

$$\Delta_O \stackrel{\text{def}}{=} (f(\mathbf{x}) < f(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathsf{sc}(w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

More examples

- *Individual fairness*

$$\Delta_I \stackrel{\text{def}}{=} \|\mathbf{x} - \mathbf{x}'\| < \epsilon$$

$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathsf{sc}(w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

- *Monotonicity*

$$\Delta_I \stackrel{\text{def}}{=} (\mathbf{x} < \mathbf{x}')$$

$$\Delta_O \stackrel{\text{def}}{=} (f(\mathbf{x}) < f(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathsf{sc}(w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

- *Robustness to noise*

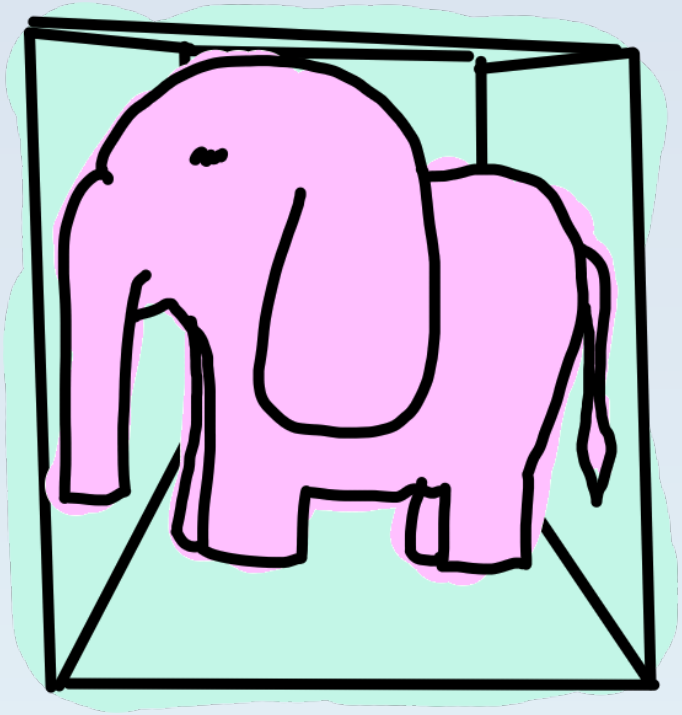
$$\Delta_I \stackrel{\text{def}}{=} (\mathbf{x}' = \mathbf{x} + \mathbf{n})$$

$$\Delta_O \stackrel{\text{def}}{=} (c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}'))$$

$$w(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} w_{\mathcal{R}}(\mathbf{x}) \cdot w_{\mathcal{R}}(\mathbf{n}) \cdot \mathsf{sc}(w_{\mathcal{S}}(\mathbf{x}, \mathbf{y}))$$

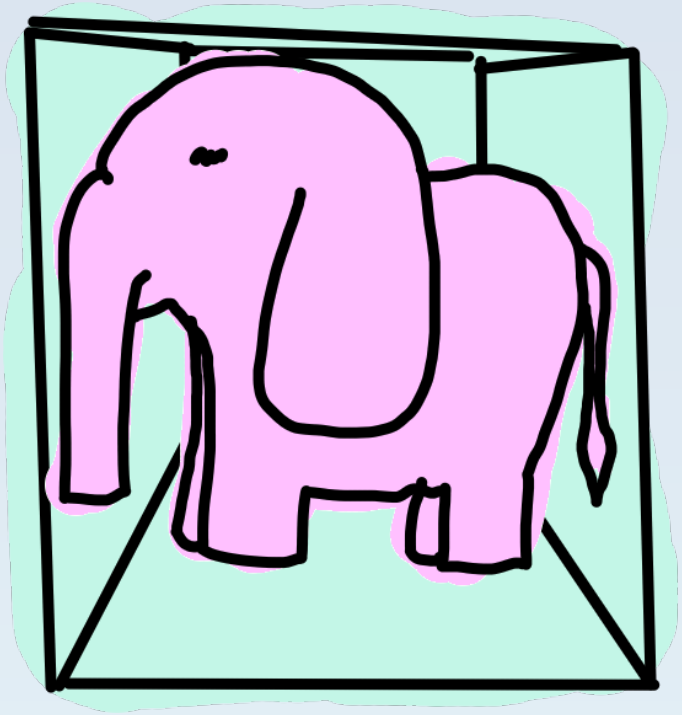
Complexity

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \overbrace{\sum_{\mu \models \Delta \wedge \chi}}^{\textbf{S1}} \underbrace{\int_{\mu} w(\mathbf{x}) d\mathbf{x}}_{\textbf{S2}}$$



Complexity

$$WMI(\Delta, w) \stackrel{\text{def}}{=} \overbrace{\sum_{\mu \models \Delta \wedge \chi}}^{\textbf{S1}} \underbrace{\int_{\mu} w(\mathbf{x}) d\mathbf{x}}_{\textbf{S2}}$$



- **S1** Enumerating μs: as hard as #SAT (#P-complete)
- **S2** Computing $m(\mu, w)$: as hard as volume computation (NP-hard)

Scalability

$$P(\mathcal{R}_O \mid \mathcal{R}_I) > k \ ?$$

Scalability

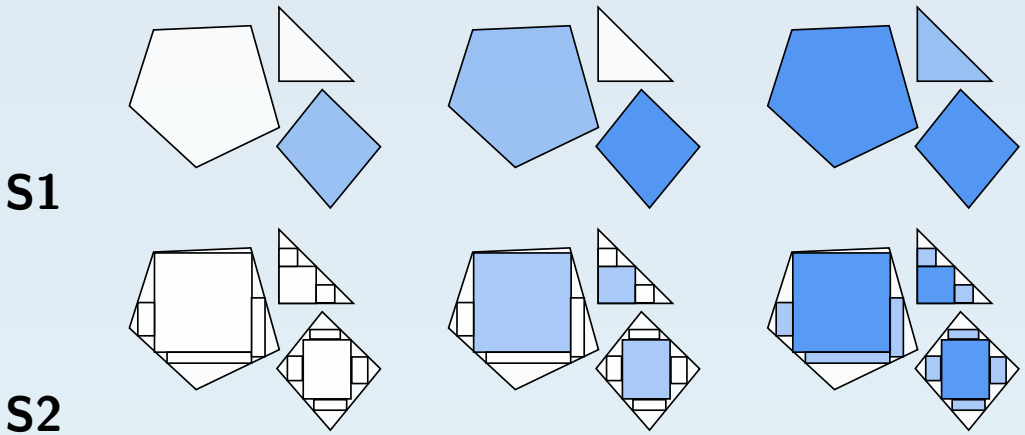
$$P(\mathcal{R}_O \mid \mathcal{R}_I) > k \ ?$$

- LHS doesn't have be computed **exactly**

Scalability

$$P(\mathcal{R}_O \mid \mathcal{R}_I) > k \ ?$$

- LHS doesn't have be computed **exactly**
  - Iteratively tightening bounds $\tilde{P}_1 \leq \tilde{P}_2 \leq \ldots.$ $P(.)$

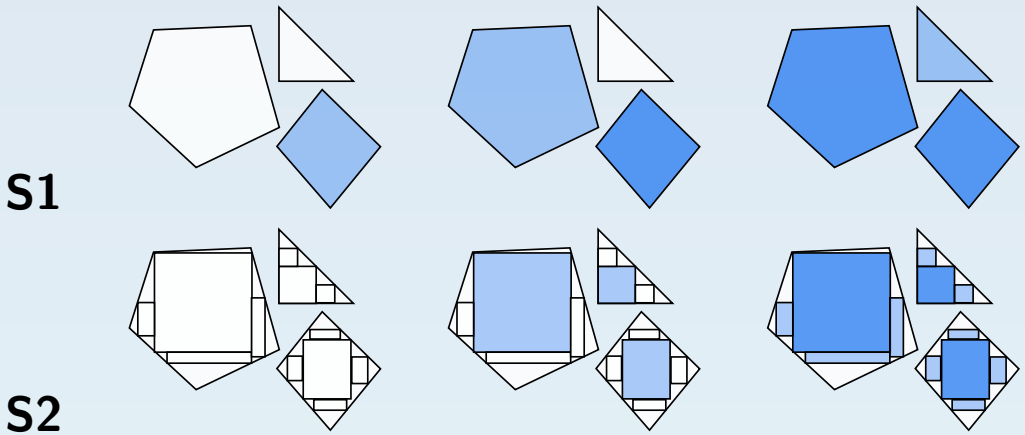


  - Approximating WMI with $\langle \epsilon, \delta \rangle$ guarantees (PAC-style)

Scalability

$$P(\mathcal{R}_O \mid \mathcal{R}_I) > k \ ?$$

- LHS doesn't have be computed **exactly**
  - Iteratively tightening bounds $\tilde{P}_1 \leq \tilde{P}_2 \leq \ldots.$ $P(.)$



  - Approximating WMI with $\langle \epsilon, \delta \rangle$ guarantees (PAC-style)

- Shouldn't be computed with a **general purpose** WMI solver

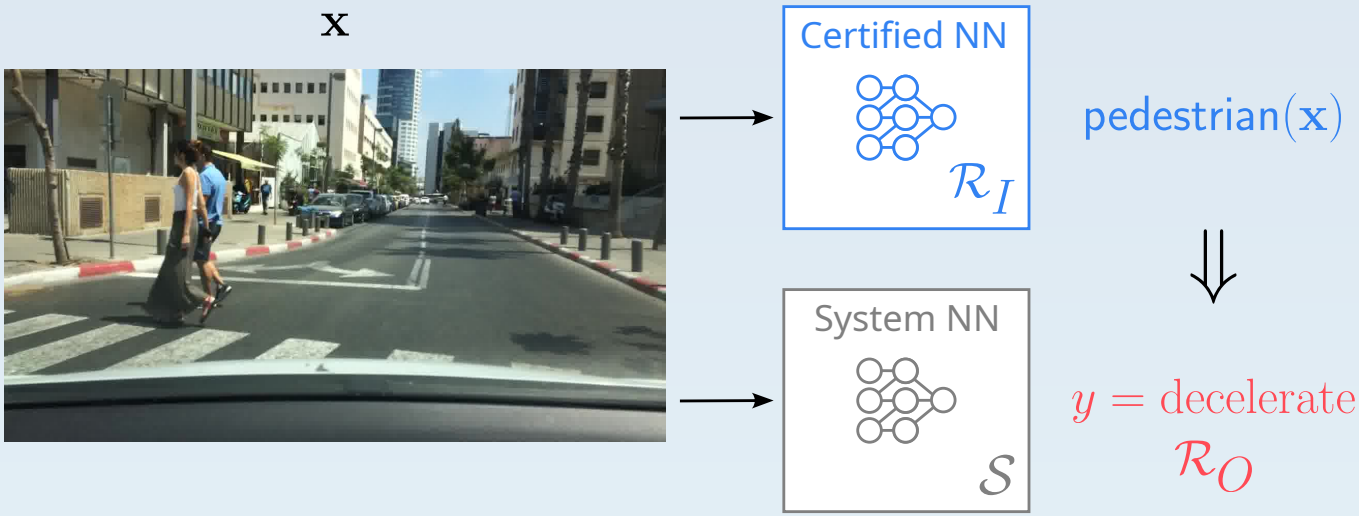
"Fairsquare: probabilistic verification of program fairness" [Albarghouthi et al. 2017]
"Provable Preimage Under-Approximation for Neural Networks" [Zhang et al. 2024]

Research directions beyond scalability

Research directions beyond scalability

- **Semantic properties**

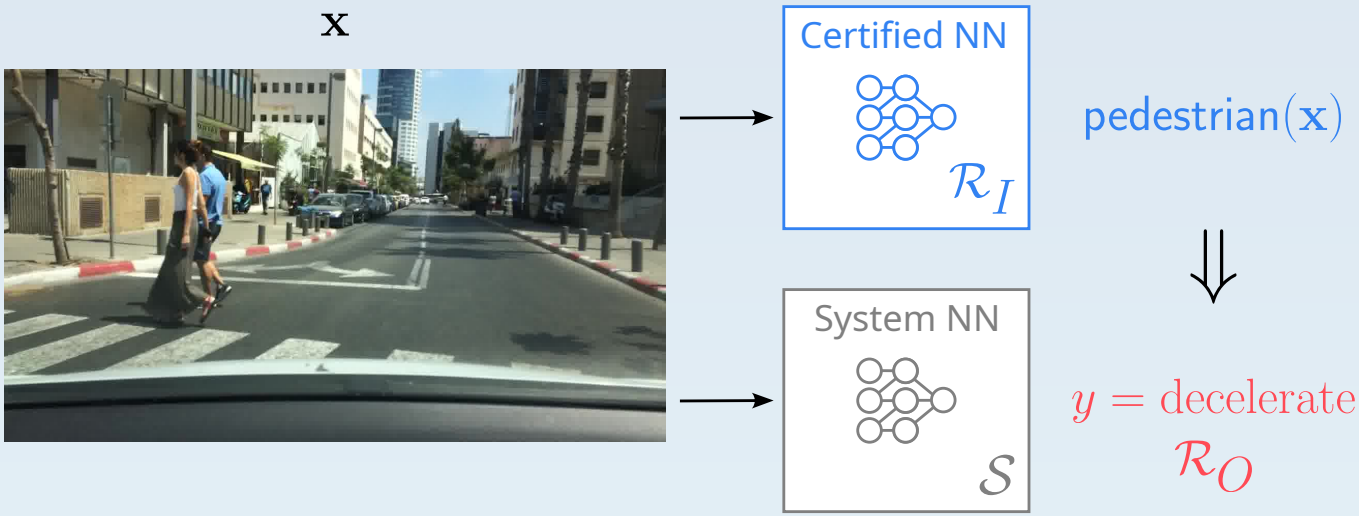
  e.g. probabilistic neuro-symbolic verification



Research directions beyond scalability

- **Semantic properties**

  e.g. probabilistic neuro-symbolic verification



- **Beyond ML verification in isolation**

  sequential models / temporal properties, verifying AI programs

Acknowledgements

Joint work with:

Roberto Sebastiani

Andrea Passerini

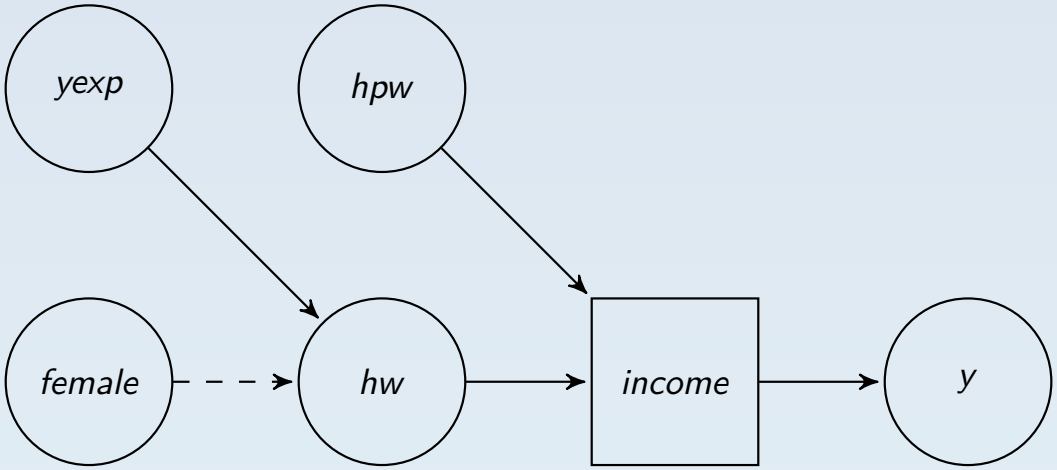
Funded by
the European Union

Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or The European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

GA n°101110960 "Probabilistic Formal Verification for Provably Trustworthy AI - PFV-4-PTAI"

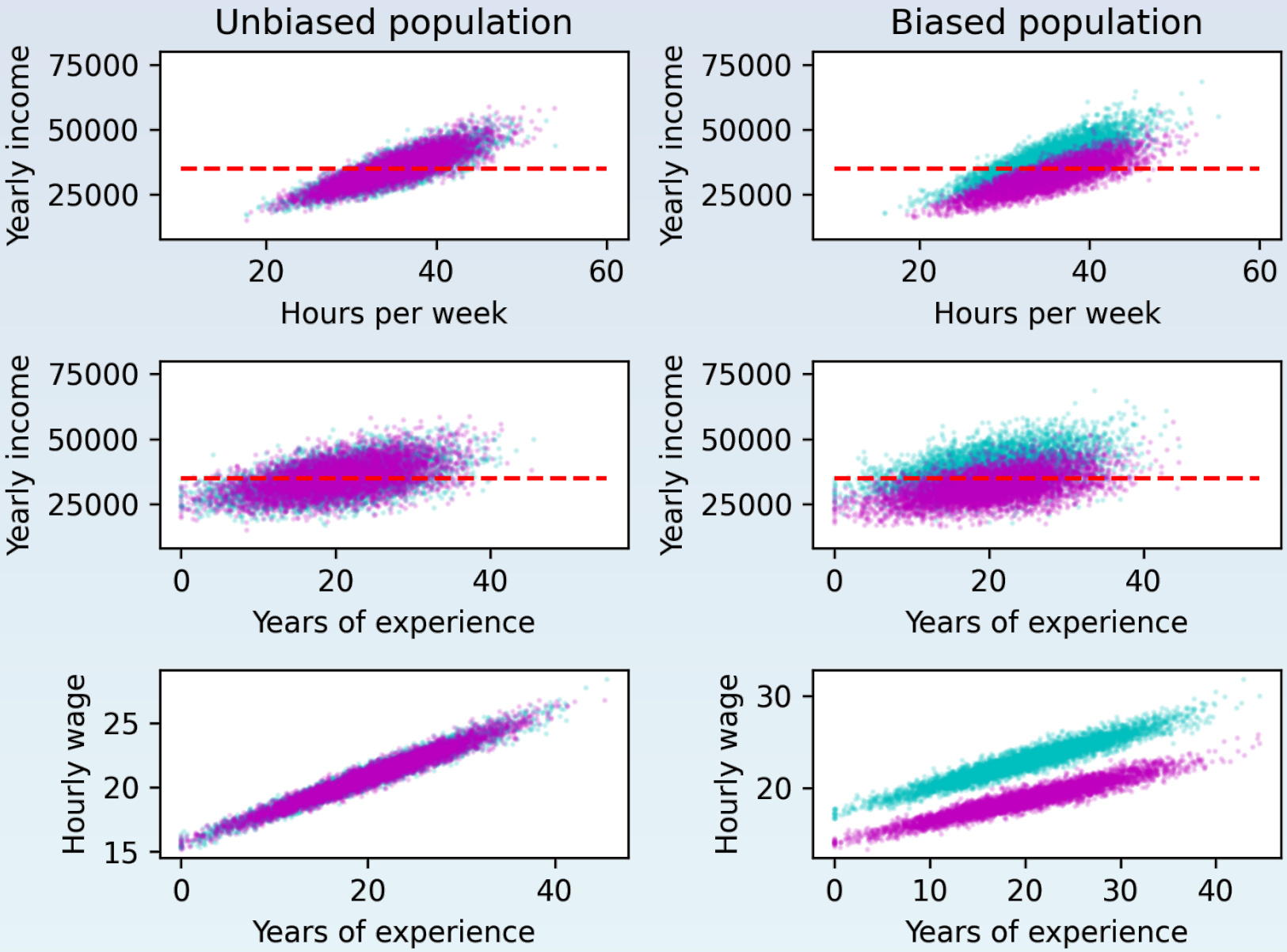
extra slides

Proof-of-concept experiment



- (yexp) *years of work experience*
- (hpw) *working hours per week*
- (hw) *hourly wage*
- (y) *income* $> k$

Proof-of-concept experiment



Proof-of-concept experiment

Proof-of-concept experiment

- Population model **DET**

  $P_{\mathcal{R}}(female, hpw, hw, yexp)$

Proof-of-concept experiment

- Population model **DET**

  $P_{\mathcal{R}}(female, hpw, hw, yexp)$

- System **NN**

  $y = f(hpw, hw, yexp)$

Proof-of-concept experiment

- Population model **DET**

  $P_{\mathcal{R}}(female, hpw, hw, yexp)$

- System **NN**

  $y = f(hpw, hw, yexp)$

- We evaluated:
  - dem.parity
  - monotonicity
  - robustness to noisy *yexp*

Proof-of-concept experiment

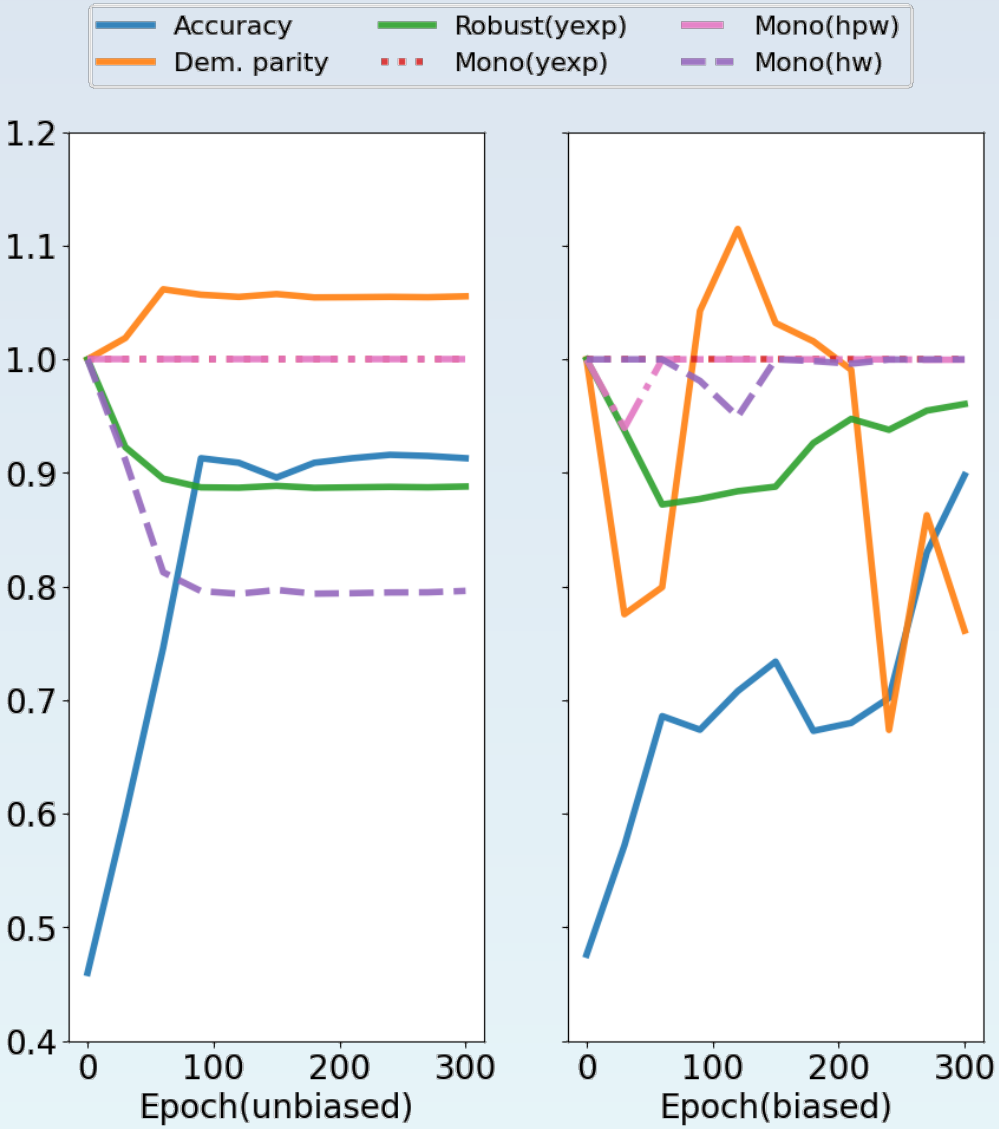
- Population model **DET**

  $P_{\mathcal{R}}(female, hpw, hw, yexp)$

- System **NN**

  $y = f(hpw, hw, yexp)$

- We evaluated:
  - dem.parity
  - monotonicity
  - robustness to noisy *yexp*



Modelling *set-based* properties

Modelling *set-based* properties

- *Demographic parity*

$$\frac{P(c(\mathbf{x}) \mid \mathbf{x} \in \mathcal{M})}{P(c(\mathbf{x}) \mid \mathbf{x} \notin \mathcal{M})} = \left[\frac{WMI(c(\mathbf{x}) \wedge \Delta_{\mathcal{M}}, w)}{WMI(\Delta_{\mathcal{M}}, w)}\right] / \left[\frac{WMI(c(\mathbf{x}) \wedge \neg\Delta_{\mathcal{M}}, w)}{WMI(\neg\Delta_{\mathcal{M}}, w)}\right]$$

if $(\mathbf{x} \in \mathcal{M})$ can be encoded as $\Delta_{\mathcal{M}}$

Modelling *set-based* properties

- *Demographic parity*

$$\frac{P(c(\mathbf{x}) \mid \mathbf{x} \in \mathcal{M})}{P(c(\mathbf{x}) \mid \mathbf{x} \notin \mathcal{M})} = \left[\frac{WMI(c(\mathbf{x}) \wedge \Delta_{\mathcal{M}}, w)}{WMI(\Delta_{\mathcal{M}}, w)}\right] / \left[\frac{WMI(c(\mathbf{x}) \wedge \neg\Delta_{\mathcal{M}}, w)}{WMI(\neg\Delta_{\mathcal{M}}, w)}\right]$$

  if $(\mathbf{x} \in \mathcal{M})$ can be encoded as $\Delta_{\mathcal{M}}$

- *Local robustness*

$$P(c(\mathbf{x}) = c_0 \mid \mathbf{x} \in \mathcal{N}_{\mathbf{x}_0,\epsilon}) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c_0) \wedge \Delta_{\mathcal{N}}, w)}{WMI(\Delta_{\mathcal{N}}, w)}$$

  where $\Delta_{\mathcal{N}}$ encodes a ϵ-ball around $\mathbf{x}_0$ (w/ class c_0):

$$\mathcal{N}_{\mathbf{x}_0,\epsilon} = \{\mathbf{x} \ : \ \|\mathbf{x} - \mathbf{x}_0\| \leq \epsilon\}$$

Modelling *hyper-properties*

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

where $[v \leftarrow v']$ substitutes all variable with fresh ones.

Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

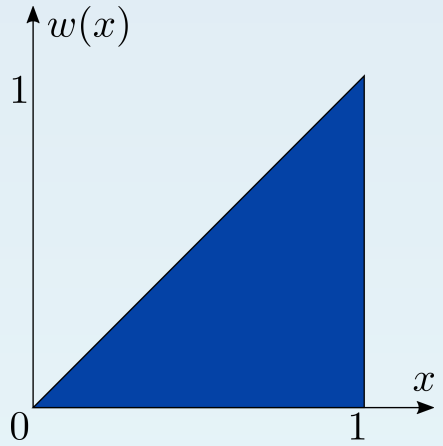
$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

  where $[v \leftarrow v']$ substitutes all variable with fresh ones.

- Example: $\chi = (x \in [0, 1])$ and $w(x) = x$:



Modelling *hyper-properties*

- Require **multiple** instantiations of the same RV:

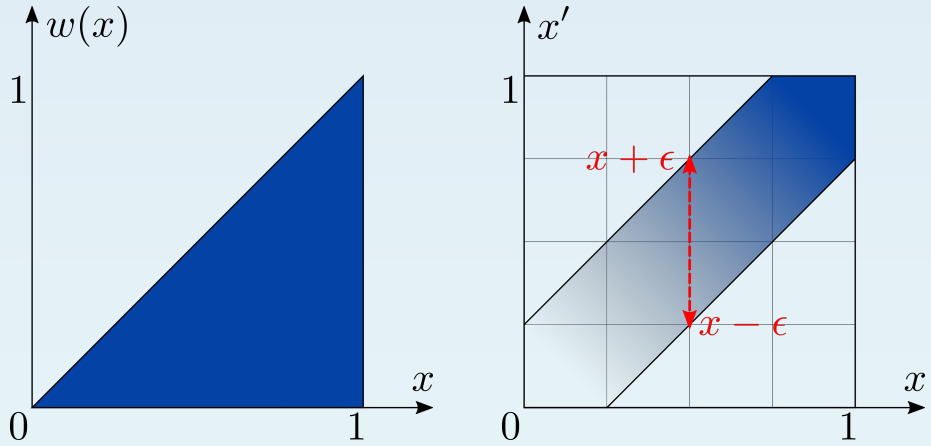
$$P(|x - x'| \leq \epsilon) \qquad x, x' \sim P(x)$$

- **Self-composition**:

$$\mathsf{sc}(\chi) \stackrel{\text{def}}{=} \chi \wedge \chi[v \leftarrow v'], \qquad \mathsf{sc}(w) \stackrel{\text{def}}{=} w \cdot w[v \leftarrow v']$$

  where $[v \leftarrow v']$ substitutes all variable with fresh ones.

- Example: $\chi = (x \in [0, 1])$ and $w(x) = x$:



Modelling *hyper-properties*

Modelling *hyper-properties*

- *Individual fairness*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathsf{sc}(w))}{WMI((\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathsf{sc}(w))}$$

Modelling *hyper-properties*

- *Individual fairness*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}{WMI((\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}$$

- *Probabilistic robustness*

  "Robustness of NNs: A Probabilistic & Practical Approach" [Mangal et al., 2019]

$$P(\|f(\mathbf{x}) - f(\mathbf{x}')\| < k \cdot \|\mathbf{x} - \mathbf{x}'\| \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon)$$

Modelling *hyper-properties*

- *Individual fairness*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}{WMI((\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}$$

- *Probabilistic robustness*

  "Robustness of NNs: A Probabilistic & Practical Approach" [Mangal et al., 2019]

$$P(\|f(\mathbf{x}) - f(\mathbf{x}')\| < k \cdot \|\mathbf{x} - \mathbf{x}'\| \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon)$$

- *Robustness to noise*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \mathbf{x}' = \mathbf{x} + \mathbf{n}) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\mathbf{x}' = \mathbf{x} + \mathbf{n}), w_{\mathcal{R}} \cdot \mathrm{sc}(w_{\mathcal{S}}) \cdot w_N)}{WMI((\mathbf{x}' = \mathbf{x} + \mathbf{n}), w_{\mathcal{R}} \cdot \mathrm{sc}(w_{\mathcal{S}}) \cdot w_N)}$$

  where $w_N(\mathbf{n})$ models the noise distribution

Modelling *hyper-properties*

- *Individual fairness*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}{WMI((\|\mathbf{x} - \mathbf{x}'\| < \epsilon), \mathrm{sc}(w))}$$

- *Probabilistic robustness*

  "Robustness of NNs: A Probabilistic & Practical Approach" [Mangal et al., 2019]

$$P(\|f(\mathbf{x}) - f(\mathbf{x}')\| < k \cdot \|\mathbf{x} - \mathbf{x}'\| \mid \|\mathbf{x} - \mathbf{x}'\| < \epsilon)$$

- *Robustness to noise*

$$P(c(\mathbf{x}) = c(\mathbf{x}') \mid \mathbf{x}' = \mathbf{x} + \mathbf{n}) = \frac{WMI((c(\mathbf{x}) \Leftrightarrow c(\mathbf{x}')) \wedge (\mathbf{x}' = \mathbf{x} + \mathbf{n}), w_{\mathcal{R}} \cdot \mathrm{sc}(w_{\mathcal{S}}) \cdot w_N)}{WMI((\mathbf{x}' = \mathbf{x} + \mathbf{n}), w_{\mathcal{R}} \cdot \mathrm{sc}(w_{\mathcal{S}}) \cdot w_N)}$$

  where $w_N(\mathbf{n})$ models the noise distribution

"Verifying Global Two-Safety Properties in NNs with Confidence" [Athavale et al., 2024]

Other *algebraic* properties

Other *algebraic* properties

- *Monotonicity*

$$P(f(\mathbf{x}) < f(\mathbf{x}') \mid \mathbf{x} < \mathbf{x}')) = \frac{WMI((f(\mathbf{x}) < f(\mathbf{x}')) \wedge (\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}{WMI((\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}$$

Other *algebraic* properties

- *Monotonicity*

$$P(f(\mathbf{x}) < f(\mathbf{x}') \mid \mathbf{x} < \mathbf{x}')) = \frac{WMI((f(\mathbf{x}) < f(\mathbf{x}')) \wedge (\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}{WMI((\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}$$

- *Equivalence* of $\mathcal{S}_1$ and $\mathcal{S}_2$

$$P(c_{\mathcal{S}_1}(\mathbf{x}) = c_{\mathcal{S}_2}(\mathbf{x})) = \frac{WMI(c_{\mathcal{S}_1}(\mathbf{x}) \Leftrightarrow c_{\mathcal{S}_2}(\mathbf{x}), w)}{WMI(\top, w)}$$

"On NN equivalence checking using SMT solvers" [Eleftheriadis et al., 2022]

Other *algebraic* properties

- *Monotonicity*

$$P(f(\mathbf{x}) < f(\mathbf{x}') \mid \mathbf{x} < \mathbf{x}')) = \frac{WMI((f(\mathbf{x}) < f(\mathbf{x}')) \wedge (\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}{WMI((\mathbf{x} < \mathbf{x}'), \mathrm{sc}(w))}$$

- *Equivalence* of $\mathcal{S}_1$ and $\mathcal{S}_2$

$$P(c_{\mathcal{S}_1}(\mathbf{x}) = c_{\mathcal{S}_2}(\mathbf{x})) = \frac{WMI(c_{\mathcal{S}_1}(\mathbf{x}) \Leftrightarrow c_{\mathcal{S}_2}(\mathbf{x}), w)}{WMI(\top, w)}$$

"On NN equivalence checking using SMT solvers" [Eleftheriadis et al., 2022]

- ..and many more!