Neuro-symbolic Liveness Verification

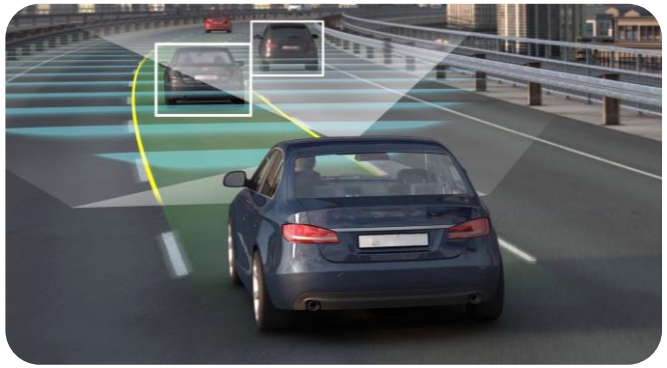
Mirco Giacobbe
Department of Computer Science, University of Oxford

DEWS Seminar, 11 June 2021, Online

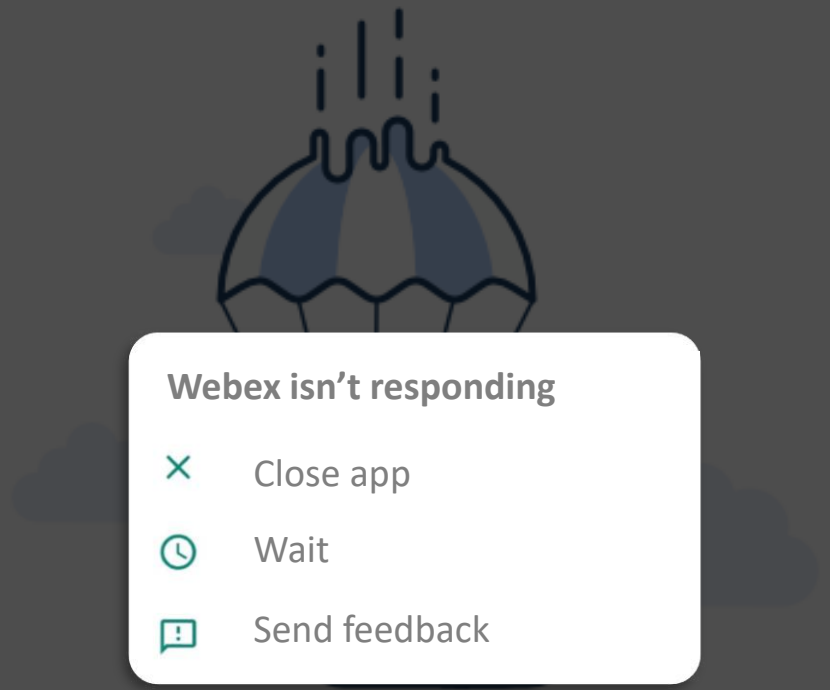
Verifying Software

Software Development

Highly Available Systems



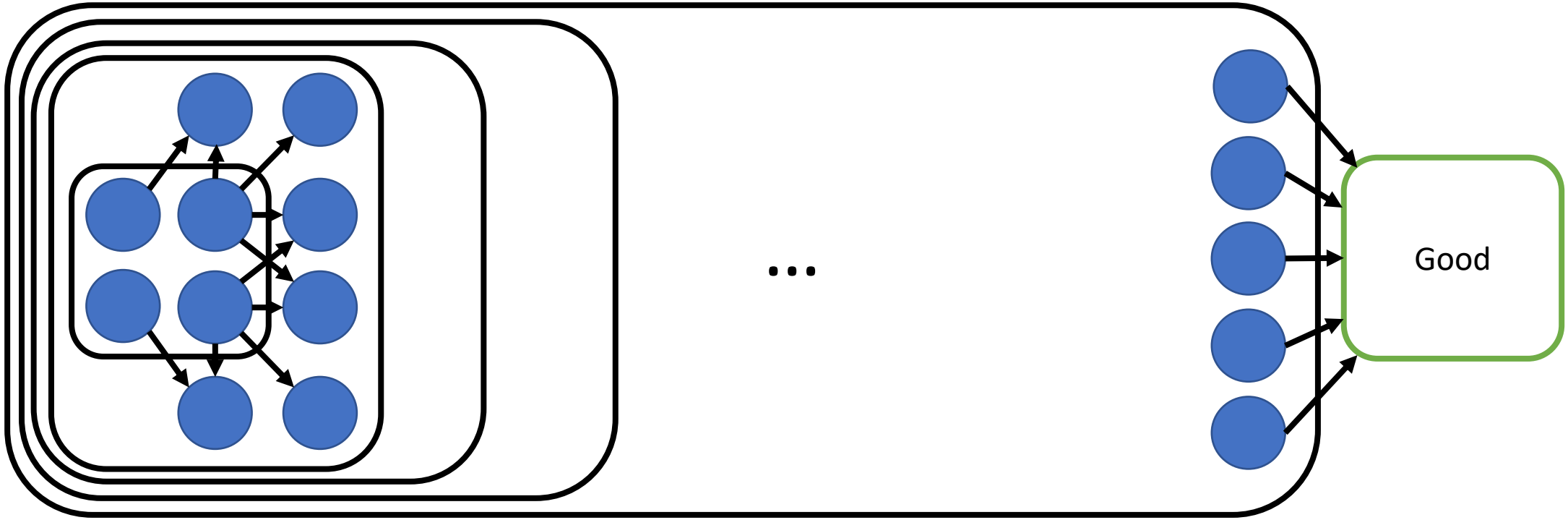
Cyber-physical Systems



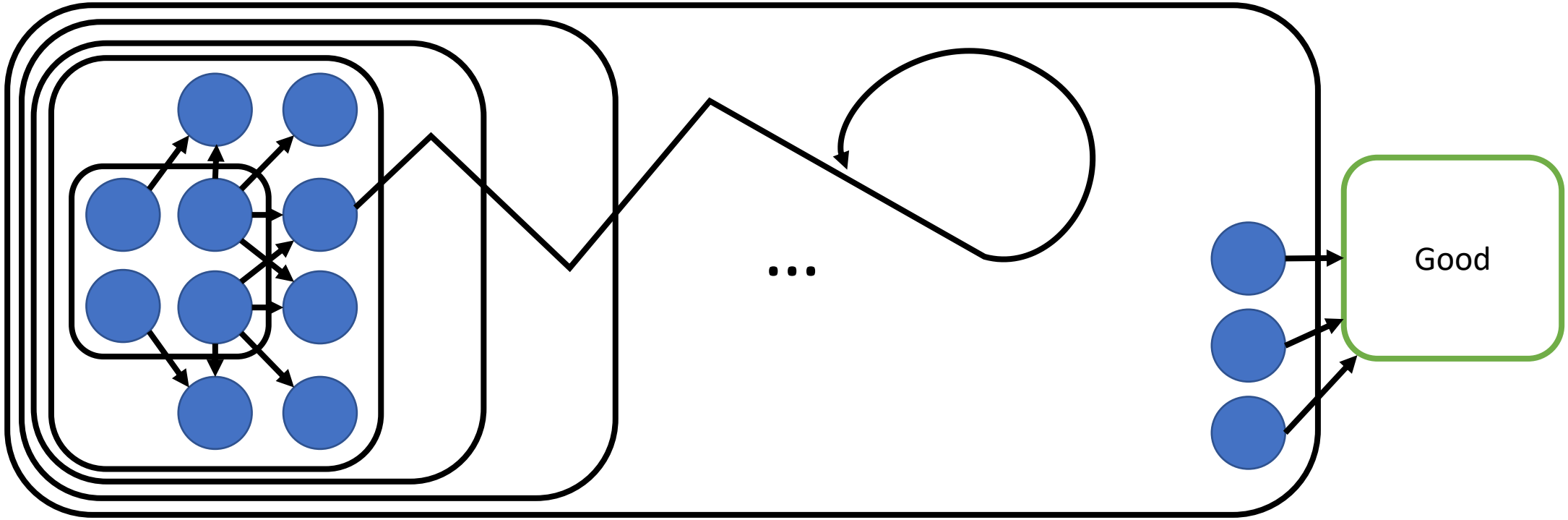
Aaaah! Something went wrong

Brace yourself till we get the error fixed.

Liveness Verification



Liveness Verification



State Transition System (or Program)

$$(S, T)$$

- State space S
- Transition relation $T \subseteq S \times S$
- A run is any sequence $s_0, s_1, s_2, \ldots$ such that $s_i, s_{i+1} \in T$ for all $i \geq 0$

The state transition system is alive if all runs hit a good state

The program terminates if all runs hit a terminal state

Ranking Functions

- Program (S, T)

Function $f : S \to W$ such that

i. $f(s) \succ f(t)$ for all $(s, t) \in T$

ii. Relation $(W, \prec)$ is **well founded**, i.e., $\prec \subseteq W \times W$ s.t. every sequence $w_0 \succ w_1 \succ w_2 \ldots$ is finite (strictly descends into a minimum)

f exists $\Rightarrow$ program terminates

Ranking Functions

- Program (S, T)

Function $f: S \to \mathbb{Z}$ such that

i. $f(s) > f(t)$ for all $(s, t) \in T$

ii. $\boldsymbol{f(s) \geq K}$ for all $s \in S$, for some constant $K \in \mathbb{Z}$

f exists $\Rightarrow$ program terminates

Ranking Functions

- Program (S, T)

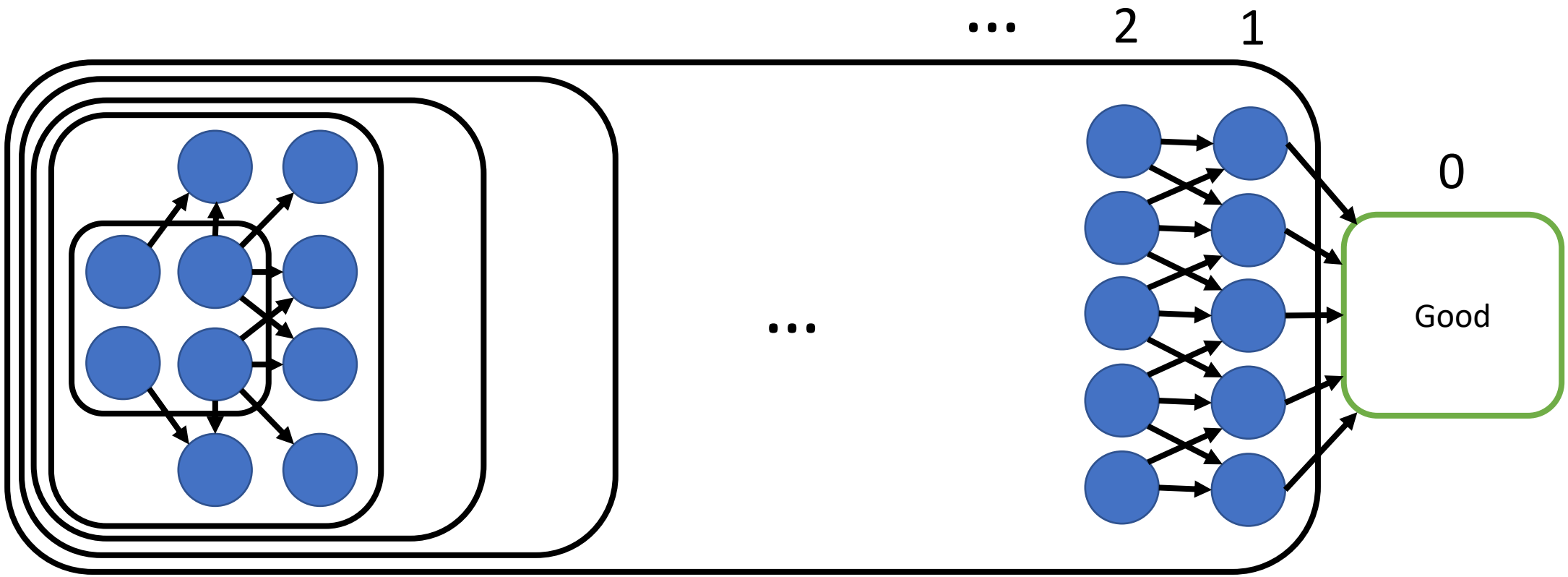
Function $f: S \to \mathbb{Z}$ such that

i. $f(s) > f(t)$ for all $(s, t) \in T$

ii. $f(s) \geq \mathbf{0}$ for all $s \in S$

f exists $\Rightarrow$ program terminates

Liveness



Ranking Functions

- Program (S, T)

Function $f : S \to \mathbb{R}$ such that

i. $f(s) \geq f(t) + \boldsymbol{\epsilon}$ for all $(s, t) \in T$, for some constant $\boldsymbol{\epsilon > 0}$

ii. $f(s) \geq 0$ for all $s \in S$

f exists $\Rightarrow$ program terminates

Termination Analysis (by ranking)

- Program (S, T)

$\exists f : S \to \mathbb{R}$ such that i. and ii. hold true

i. $f(s) \geq f(t) + \epsilon$ for all $(s, t) \in T$, for some constant $\epsilon > 0$

ii. $f(s) \geq 0$ for all $s \in S$

f exists $\Rightarrow$ program terminates

Machine Learning (in a nutshell)

- Input domain X
- Output domain Y

$$\underset{\boldsymbol{f:X\to Y}}{\operatorname{argmin}} \frac{1}{|D|} \sum_{d \in D} \mathcal{L}(d, f)$$

- Dataset D
- Loss function $\mathcal{L}: D \times (X \to Y) \to \mathbb{R}_{\geq 0}$

Machine Learning (in a nutshell)

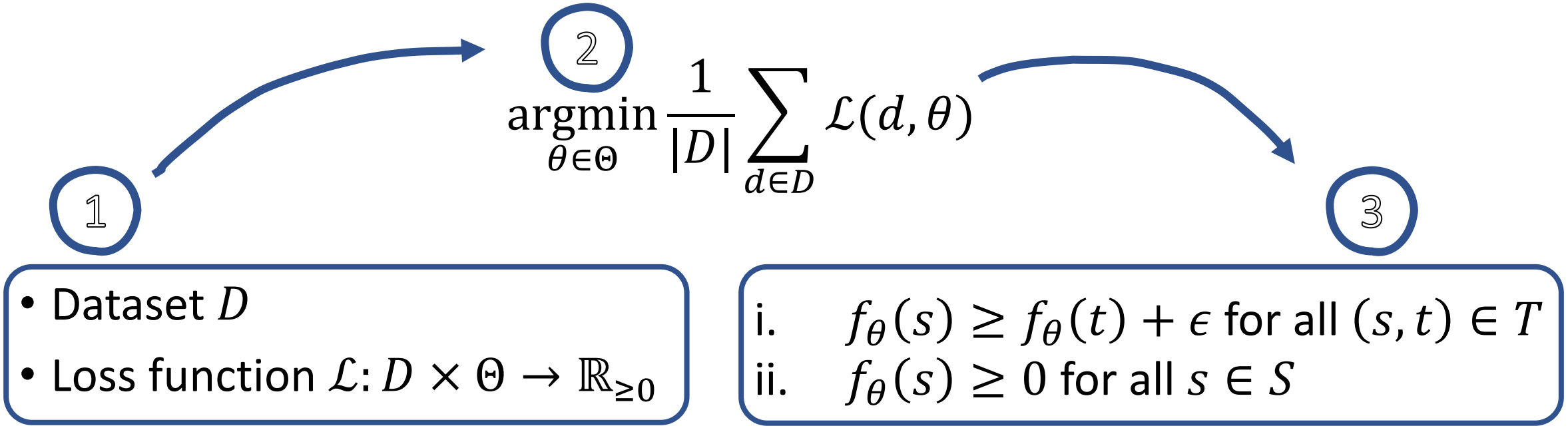
- Input domain X
- Output domain Y
- Parameterised model $f_\theta : X \to Y$ with parameter $\theta \in \Theta$

$$\underset{\boldsymbol{\theta \in \Theta}}{\operatorname{argmin}} \frac{1}{|D|} \sum_{d \in D} \mathcal{L}(d, \theta)$$

- Dataset D
- Loss function $\mathcal{L}: D \times \Theta \to \mathbb{R}_{\geq 0}$

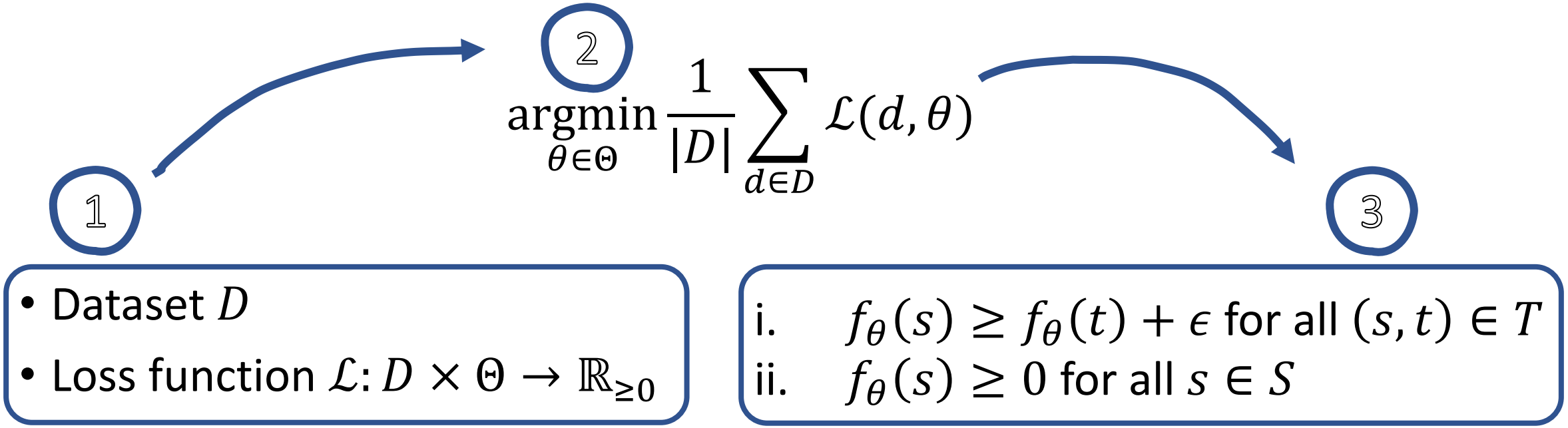
Learning a Ranking Function

- Program (S, T)
- Parameterised model $f_\theta : S \to \mathbb{R}$ with parameter $\theta \in \Theta$

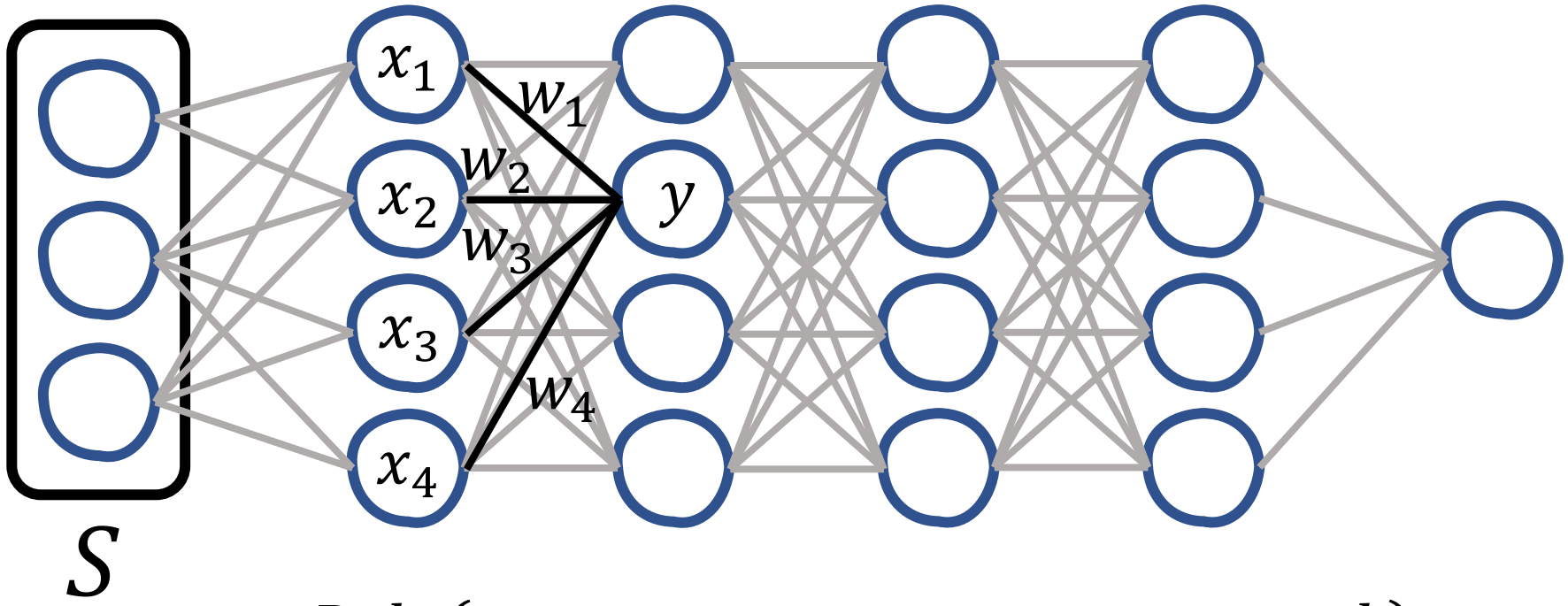


Neural Termination Analysis

- Program (S, T)
- **Neural Network** $f_\theta : S \to \mathbb{R}$ with parameter $\theta \in \Theta$



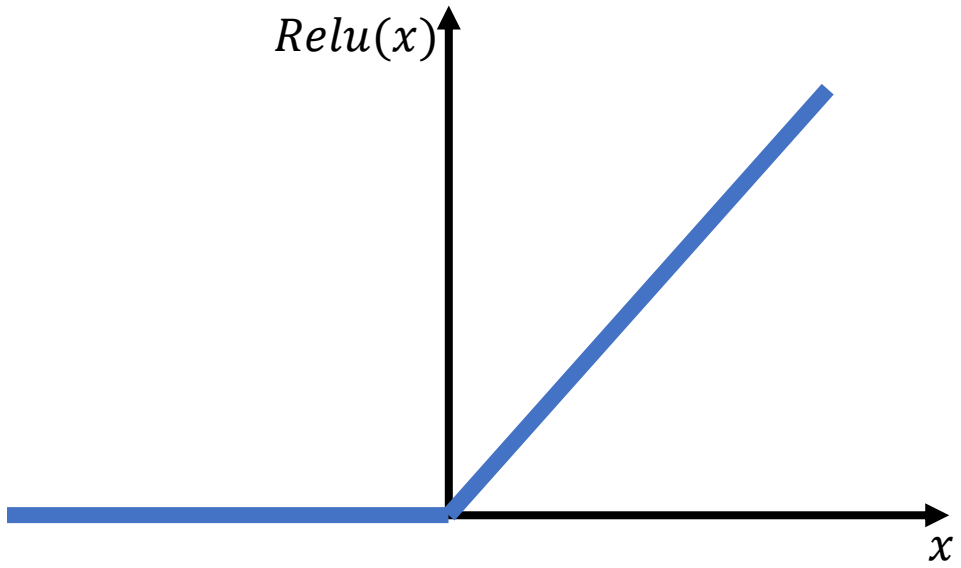
Neural Network $f_\theta : S \to \mathbb{R}$



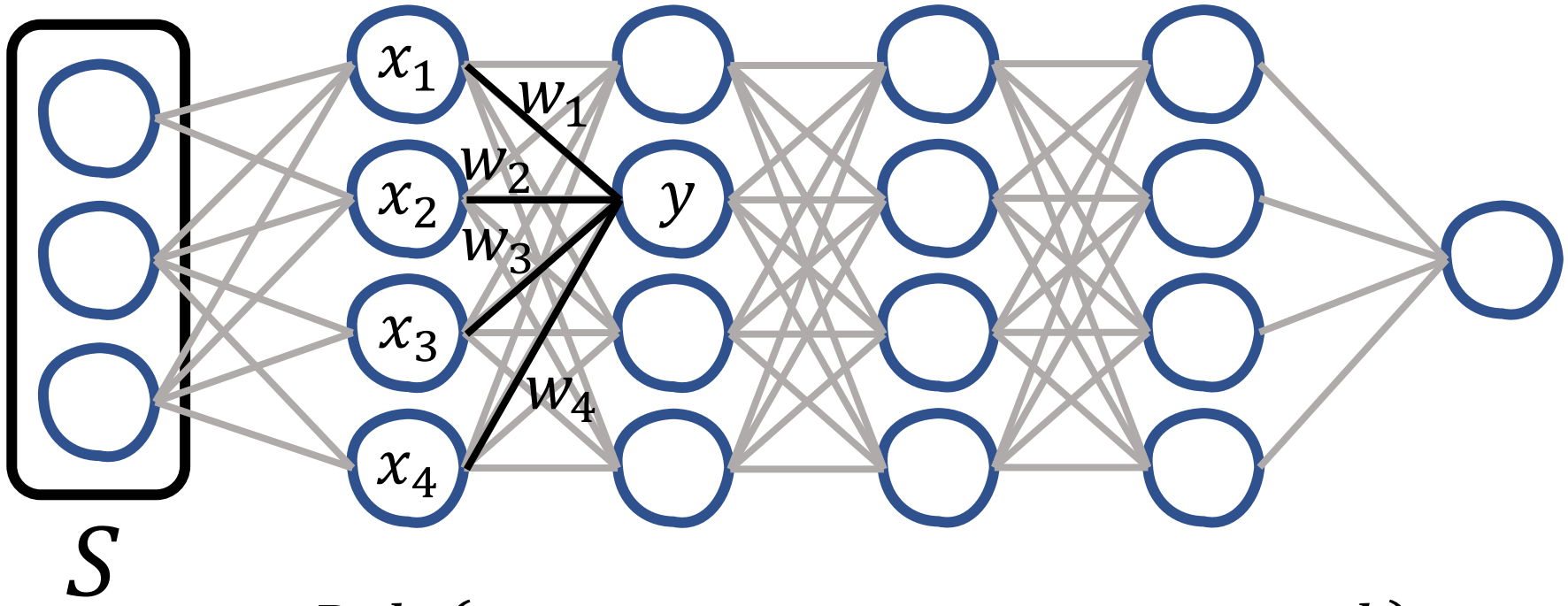
$$y = Relu(w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b)$$

$$Relu(x) = \max\{x, 0\}$$

$$Relu(x) = \max\{x, 0\}$$



Neural Network $f_\theta : S \to \mathbb{R}_{\geq 0}$

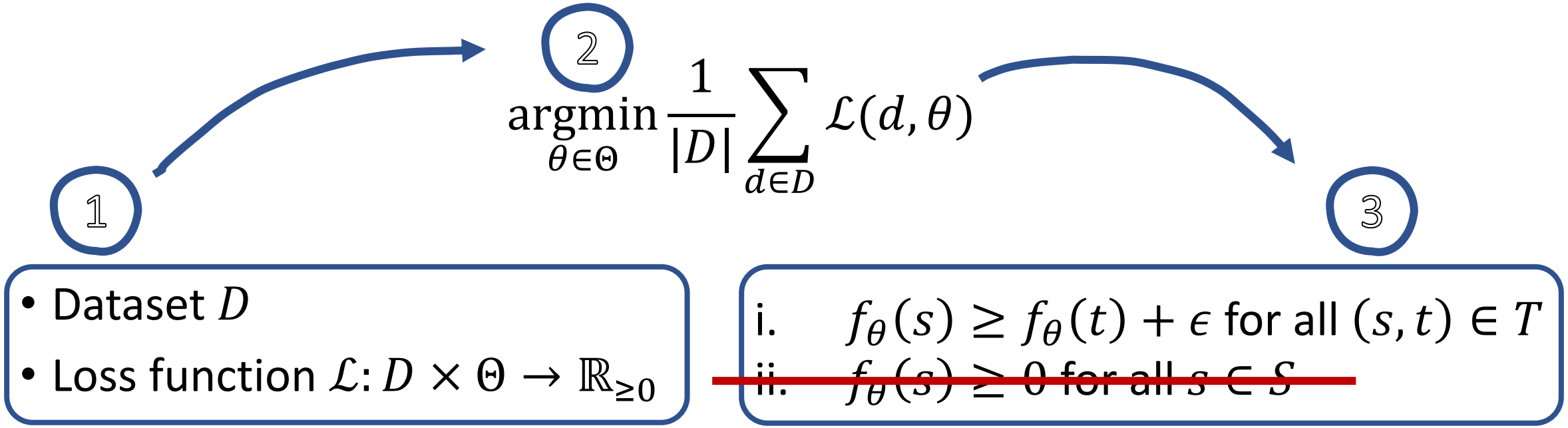


$$y = Relu(w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b)$$

$$Relu(x) = \max\{x, 0\}$$

Neural Termination Analysis

- Program (S, T)
- **Neural Network** $f_\theta : S \to \mathbb{R}_{\geq 0}$ with parameter $\theta \in \Theta$



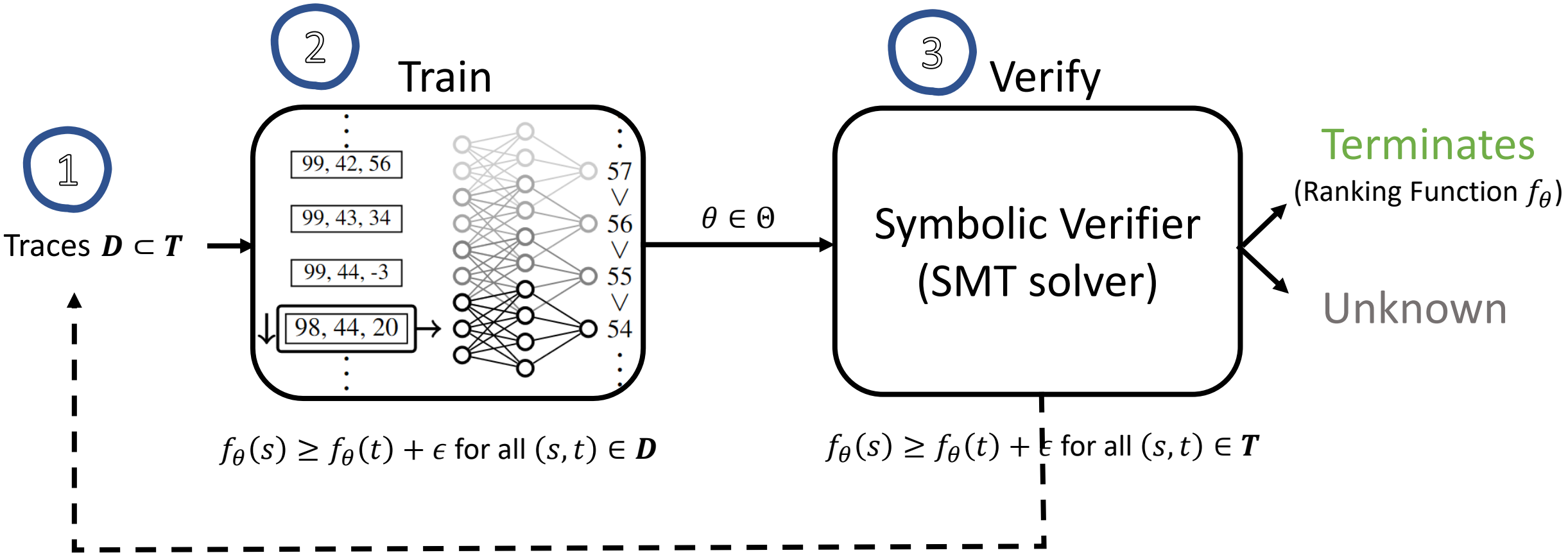
Training a Neural Ranking Function $f_\theta : S \to \mathbb{R}_{\geq 0}$

- Input: dataset $D \subset T$, finite sampling of T
- Output: $f_\theta(s) \geq f_\theta(t) + \epsilon$ for all $(s, t) \in \boldsymbol{D}$

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{|D|} \sum_{(\boldsymbol{s},\boldsymbol{t}) \in \boldsymbol{D}} \mathcal{L}(\boldsymbol{s}, \boldsymbol{t}, \theta)$$

$$\mathcal{L}(s, t, \theta) = \max\{f_\theta(t) - f_\theta(s) + \epsilon, 0\}$$

Train-Verify Framework



Software Termination Analysis

Mirco Giacobbe, Daniel Kroening, Julian Parsert. **Neural Termination Analysis,** 2021

A Simple Example

```
while (i < n) {
  i++;
}
```

1

```
i   n

0   4
1   4
2   4
3   4

50  4

3   81
4   81
5   81
6   81
7   81
8   81
…
```

2

$Relu(n - i)$

3

```
assume i!0 < n!0
R!0 := Relu(n!0 - i!0)
i!1 := i!0 + 1
R!1 := Relu(n!0 - i!1)
assert R!1 < R!0
```

Probabilistic Programs

Alessandro Abate, Mirco Giacobbe, Diptarko Roy. **Learning Probabilistic Termination Proofs,** CAV 2021

```
while (red > 0 || blue > 0) {
  p ~ Bernoulli(.01);
  if p == 1 then
    red = red - 1
  else
    blue = blue - 1
}
```

Probabilistic Termination

- Program = Stochastic Process $\{X_t\}_{t \in \mathbb{N}}$
- Stopping time T

```
while guard {
  update with branching
}
```

- Almost-sure termination $\mathbb{P}[T < \infty] = 1$
- Positive almost-sure termination $\mathbb{E}[T] < \infty$

Ranking Supermartingales

- Probabilistic program $\{X_t\}_{t\in\mathbb{N}}$ s.t. $X_t \in \mathbb{R}^n$

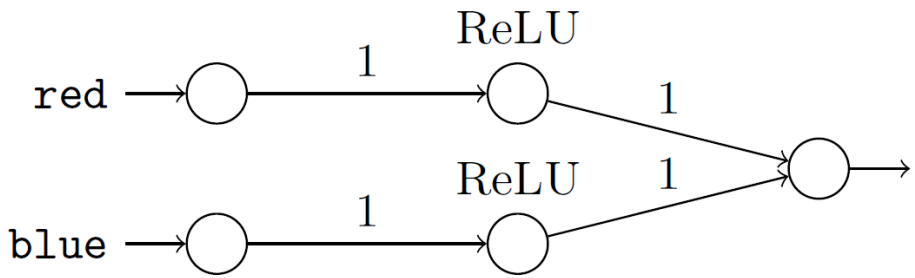
Function η: $\mathbb{R}^n \to \mathbb{R}$ such that

i. $\mathbb{E}[\eta(X_{t+1})|X_t = x] \leq \eta(x) - \epsilon$ for all $x \in \mathbb{R}^n$

~~ii. $\eta(s) \geq 0$ for all $x \in \mathbb{R}^n$~~

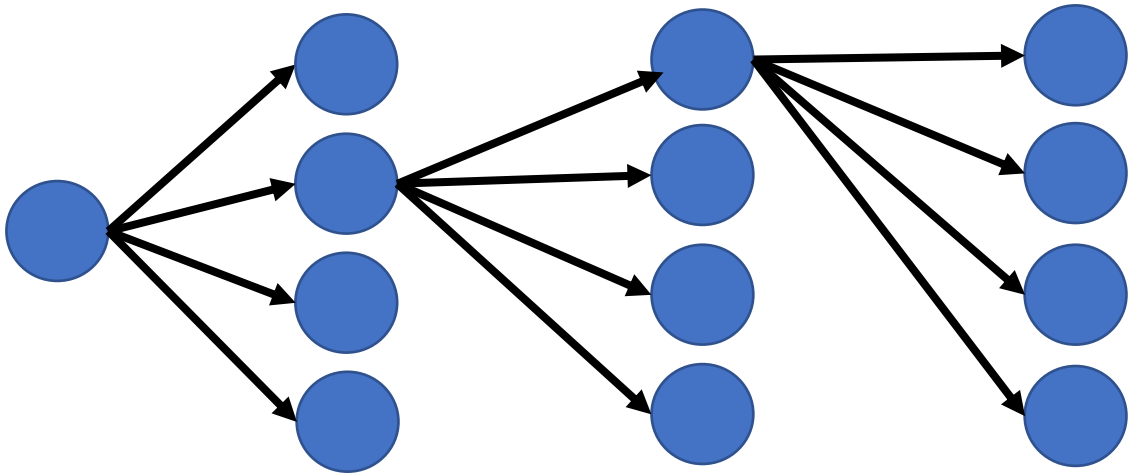
f exists ⇒ program almost-surely terminates in expected finite time

```
while (red > 0 || blue > 0) {
  p ~ Bernoulli(.01);
  if p == 1 then
    red = red - 1
  else
    blue = blue - 1
}
```



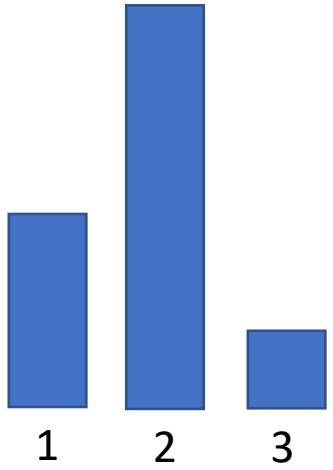
$$Relu(red) + Relu(blue)$$

Sampling from a Probabilistic Program



...

S'



$s \in \mathbb{R}^n$ $S' \subseteq \mathbb{R}^n$

Training a Neural Ranking Supermartingale

- Input: dataset $D \subset \mathbb{R}^n \times \wp(\mathbb{R}^n)$
- Output: $\mathbb{E}[\eta_\theta(X_{t+1})|X_t = x] \leq \eta_\theta(x) - \epsilon$ for all $x \in \mathbb{R}^n$

$$\operatorname*{argmin}_{\theta \in \Theta} \frac{1}{|D|} \sum_{(s,S') \in D} \mathcal{L}(s, S', \theta)$$

$$\mathcal{L}(s, S', \theta) = \max\{\mathbb{E}_{s' \in S'}[\eta_\theta(s')] - \eta_\theta(s) + \epsilon, 0\}$$

Experimental Comparison

Program	Amber [39]	Farkas' lemma [2]	Ab-synth [41]	Succ. rate	Inter.	Train.	Verif.	#iter	NRSM
Hare & Tortoise (d)	0.04	≈0	0.09	10/10	0.61	3.86	0.70	0	SOR
exmini/terminate (d)	—	0.02	oot	10/10	1.75	29.35	7.67	2	SOR
aaron2 (d)	0.03	0.02	0.02	10/10	0.04	2.27	0.01	0	SOR
catmouse (c)	0.03	0.02	—	9/10	0.39	12.41	3.68	1	SOS
counterex1c (d)	—	0.02	0.22	8/10	1.00	6.71	0.02	0	SOR
easy1 (d)	0.12	0.01	0.05	10/10	1.12	5.55	1.27	0	SOR
easy2 (c)	0.04	0.02	—	10/10	1.55	6.79	0.18	0	SOS
ndecr (d)	0.04	0.02	0.03	10/10	1.18	5.63	0.02	0	SOR
random1d (c)	0.05	0.02	—	10/10	1.14	4.86	0.79	0	SOS
rsd (d)	error	0.01	oot	10/10	1.14	6.18	2.04	0	SOR
speedFails1 (d)	0.07	0.01	0.04	10/10	0.45	4.09	0.67	0	SOR
speedPldi2 (d)	—	0.02	0.40	9/10	1.36	7.85	0.02	0	SOR
speedPldi3 (d)	—	0.02	0.36	8/10	2.58	30.70	2.12	1	SOR
speedPldi4 (d)	—	0.02	0.17	10/10	0.68	5.07	0.04	0	SOR
speedSingleSingle (c)	0.03	0.02	—	10/10	0.39	2.85	0.51	0	SOS
speedSingleSingle2 (d)	—	0.02	0.15	10/10	0.83	7.30	0.04	0	SOR
wcet0 (d)	—	0.02	0.10	10/10	1.45	5.64	0.09	0	SOR
wcet1 (d)	—	0.02	0.10	10/10	0.85	4.31	0.09	0	SOR
probfact (d)	—	—	n/a	10/10	0.49	6.12	0.16	0	SOR
probfact2 (d)	—	—	n/a	10/10	0.45	5.89	0.23	0	SOR
marbles (d)	—	—	n/a	10/10	0.84	10.83	0.91	0	SOR
marbles3 (d)	—	—	n/a	10/10	0.40	70.14	7.87	2	SOR
crwalk (c)	—	—	—	10/10	0.53	3.06	1.56	1	SOS
crwalk2 (c)	—	—	—	10/10	1.32	3.11	0.75	1	SOS
expdistrw (c)	n/a	—	—	10/10	0.05	1.53	0.01	0	SOS
expdistrw2 (c)	n/a	—	—	10/10	4.92	3.15	1.03	1	SOS
gaussrw (c)	—	—	—	10/10	10.30	3.45	0.75	0	SOS
gaussrw2 (c)	—	—	—	9/10	15.46	4.91	5.33	0	SOS
slicedcauchy (c)	—	—	—	10/10	0.02	3.31	0.01	0	SOR
slicedcauchy2 (c)	—	—	—	10/10	0.01	2.16	0.03	0	SOR

[39] Amber: Moosbrugger, Bartocci, Katoen, Kovács**: Automated termination analysis of polynomial probabilistic programs**. ESOP 2021

[2] Farkas: Agrawal, Chatterjee, **Novotný Lexicographic ranking supermartingales: an efficient approach to termination of probabilistic programs** POPL 2018

[41] Ngo, Carbonneaux, and Hoffmann**. Bounded expectations: resource analysis for probabilistic programs.** *ACM SIGPLAN Notices* 2018 .

Stability Analysis

Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, Andrea Peruffo. **Formal Synthesis of Lyapunov Neural Networks,** IEEE L-CSS 2021

Stability Analysis / Lyapunov Functions

$$\dot{x} = f(x), \qquad x \in \mathbb{R}^n$$

- Polynomial autonomous systems
- Region of interest $I \subseteq \mathbb{R}^n$

i. $\dot{V}(x) = \nabla V(x) \cdot f(x) < 0$ for all $x \in I - \{x_e\}$

~~ii. $V(x) > 0$ for all $x \in A - \{x_e\}$~~

~~iii. $V(x_e) = 0$~~

Training a Lyapunov Neural Network $V_\theta : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$

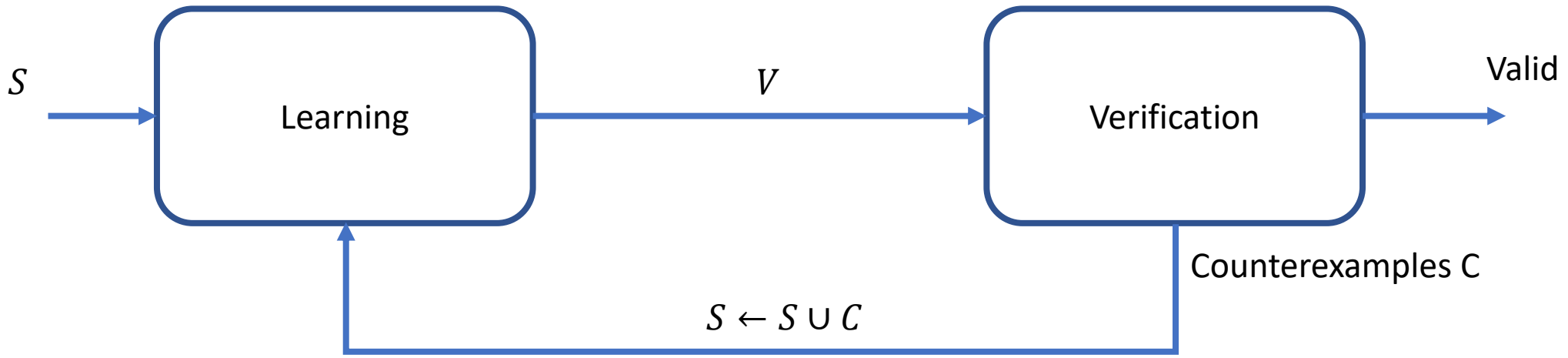
- Input: dataset $D \subset I$, finite sampling from region of interest
- Output: $\dot{V}(x) = \nabla V(x) \cdot f(x) < 0$ for all $x \in D$

$$\underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{|D|} \sum_{x \in D} \mathcal{L}(x, \theta)$$

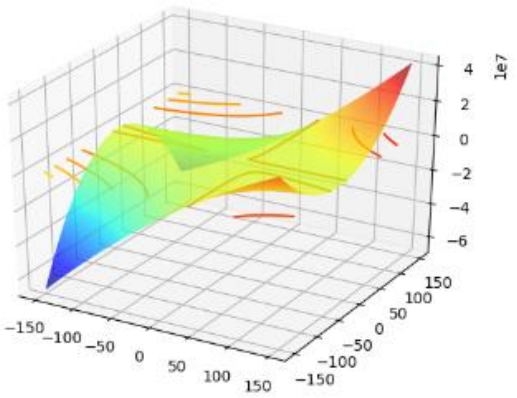
$$\mathcal{L}(x, \theta) = \max\{\nabla V(x) \cdot f(x) + \delta, 0\}$$

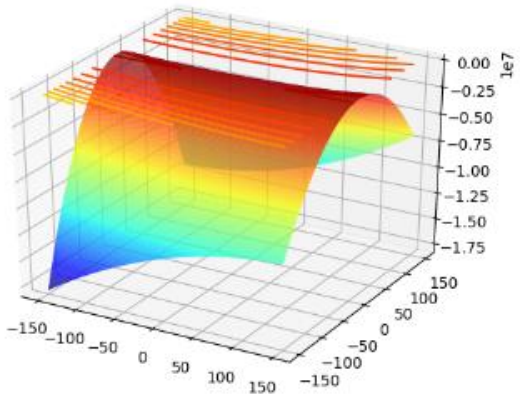
small $\delta > 0$

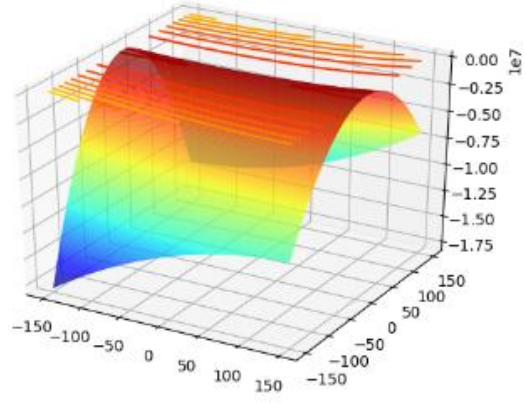
Counterexample-guided Synthesis Loop

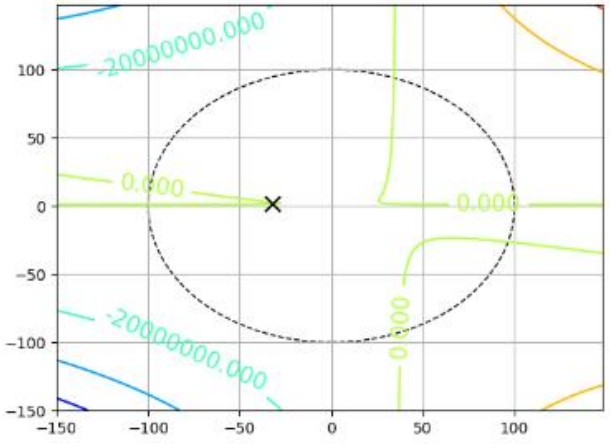


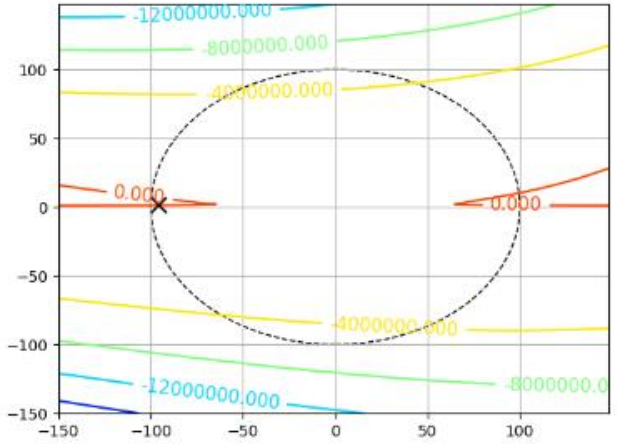
Counterexample-guided Synthesis Loop

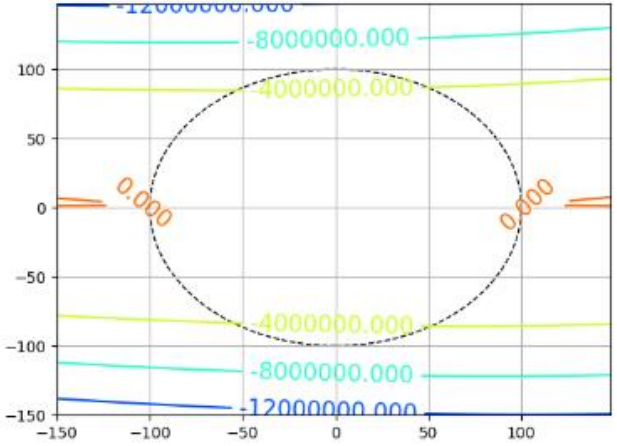






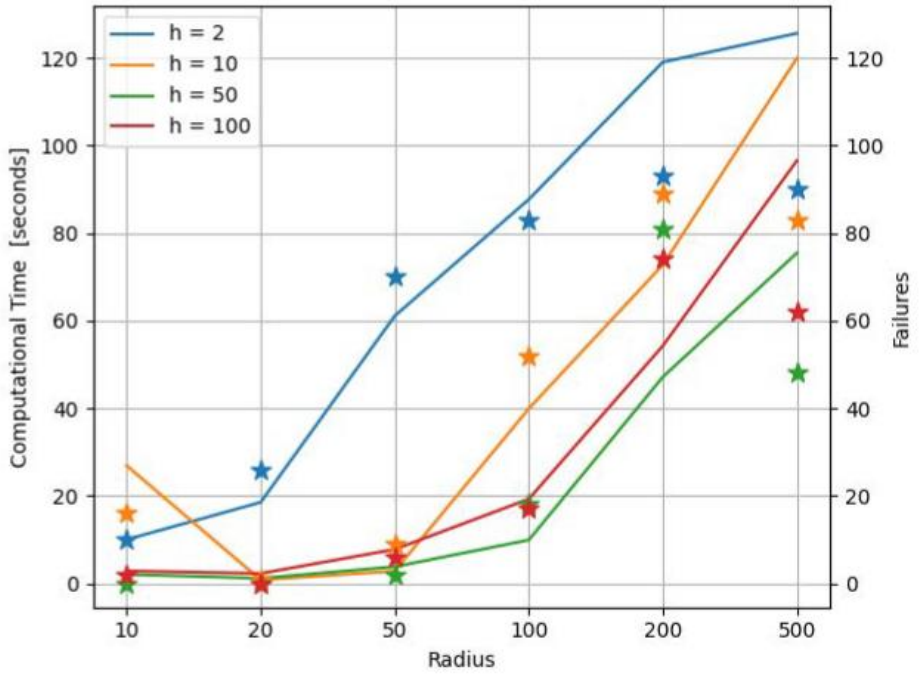






Experimental Comparison

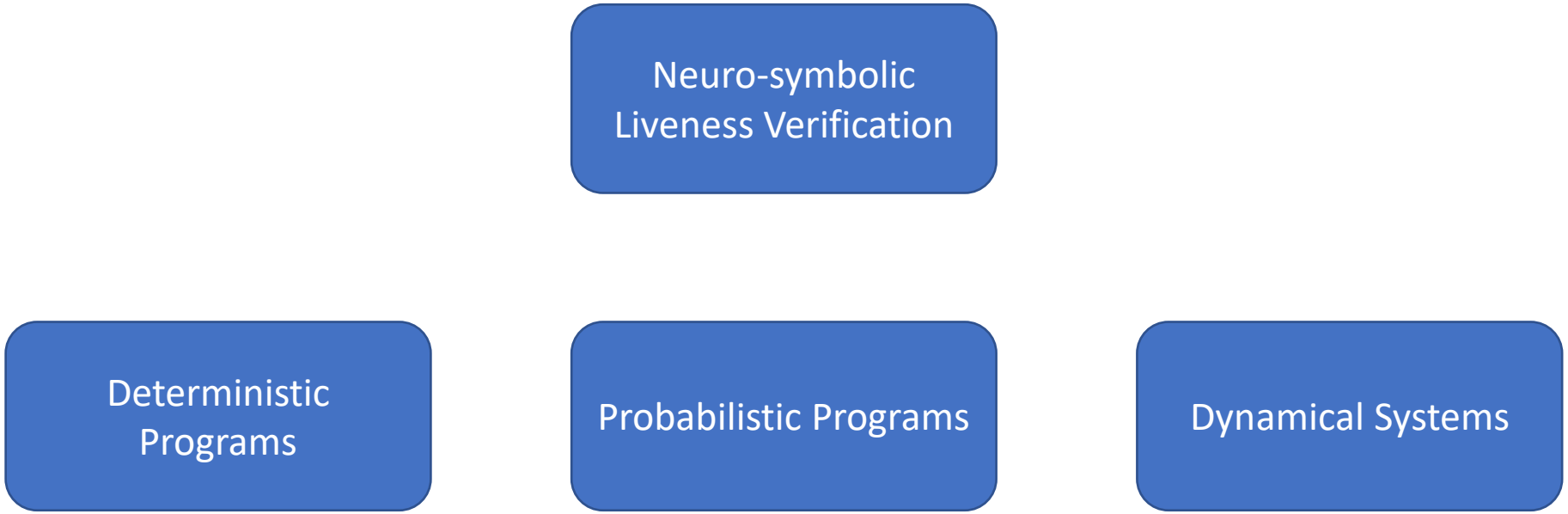
LNN Time	LNN r	NLC[2] Time	NLC Domain	CBS[3] Time	CBS r	SOS Time	SOS r
12.01	500	6.28	1	0.22	1	6.67	800
0.29	100	5.45	1	0.30	1	7.76	25
0.32	1000	54.12	1	2.22	1	11.80	oot
33.27	1000	37.80	1	0.42	1	9.65	oot



NLC: Chang, Roohi, Gao **Neural Lyapunov Control**, NeurIPS 2019.

CBS: Ahmed, Peruffo, Abate. **Automated and Sound Synthesis of Lyapunov Functions with SMT Solver**s, TACAS 2020

Learning Proof Witnesses from Examples



(Neural)	Ranking Function	Ranking Supermartingale	Lyapunov Function

Conclusion

Checking a proof is much easier than **finding** one

- Guess a candidate proof (witness) from examples
- Then check whether it is correct using formal methods

- Mirco Giacobbe, Daniel Kroening, Julian Parsert. **Neural Termination Analysis,** arXiv 2021
- Alessandro Abate, Mirco Giacobbe, Diptarko Roy. **Learning Probabilistic Termination Proofs,** CAV 2021
- Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, Andrea Peruffo. **Formal Synthesis of Lyapunov Neural Networks,** IEEE L-CSS 2021