Probabilistic verification of neural networks

Eric Goubault, Sylvie Putot and Francesc Pifarré-Esquerda

January 29th, SIPTA Seminar

Overview

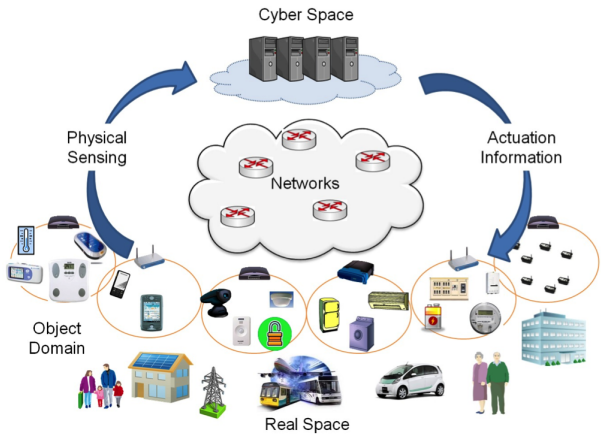
1. Context: neural networks in CPS
2. Probabilistic analysis of CPS
3. A plain DSI approach
4. From interval to zonotopic focal elements
5. From independent input to general dependencies

Context: neural networks in CPS

CPS?

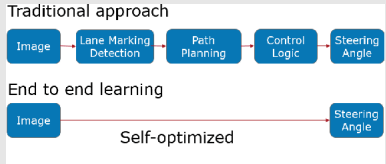
Cyber-Physical Systems = (distributed) control systems

Needs high-confidence!



Example : Autonomous vehicle

Typical approaches

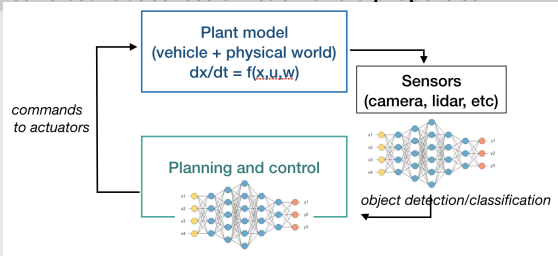


Various levels in between (perception-prediction through learning, control-motion planning through classical model-based methods etc.)

The closed-loop: a time-triggered hybrid system

Uncertain environment and neural network control : robustness or reach-avoid properties

- plant dynamic f,
- state x, control u, disturbance w
- NN controller h, period Δt_u

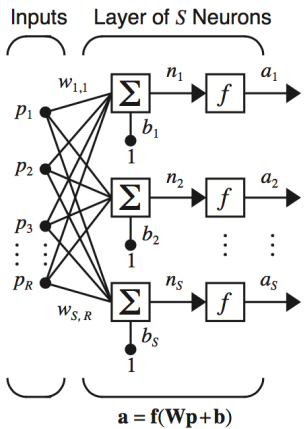


Time-triggered (u computed every $\Delta_u t$) dynamical system with non-linear feedback:

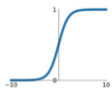
$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), w(t)) \\
x(t_0) &= x_0 \in X_0 \\
u(t) &= u_k = h(y(x(\tau_k))), \text{ for } t \in [\tau_k, \tau_{k+1}), \text{ with } \tau_k = t_0 + k\Delta t_u, \ \forall k \geq 0
\end{aligned}
$$

Feedforward neural network controlled system

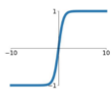
Each layer consists in a linear transform followed by a non linear activation function:



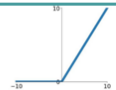
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



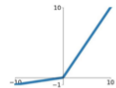
tanh
$\tanh(x)$



ReLU
$\max(0, x)$

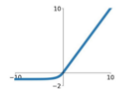


Leaky ReLU
$\max(0.1x, x)$



Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

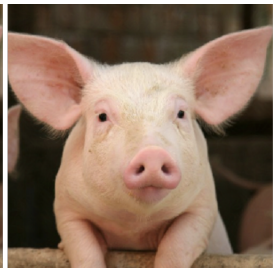


There is a need for safe design!

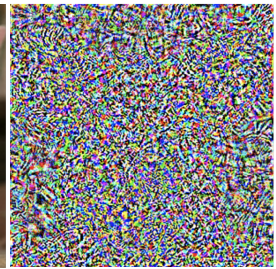
Extra difficulty, mostly with data-driven algorithms (neural nets)

Linked to adversarial attacks

classified as pig (91% confidence)



classified as airliner (99% confidence)



(difference between the two images, /0.005)

- Robustness properties: how big $\epsilon > 0$ can be so that if $||image' - image|| < \epsilon$, image' is classified as image
- Sometimes we have specifications (e.g. ACAS Xu)
- Or we can prove general properties of the neural net in the control loop, possibly probabilistically (uncertainties etc)

Reachability-based verification

Safety verification, temporal properties

- Compute enveloppes of all possible trajectories
- If these enveloppes do not intersect with unsafe sets of states, then the system is safe
- We may then want to prove some additional temporal properties

Method

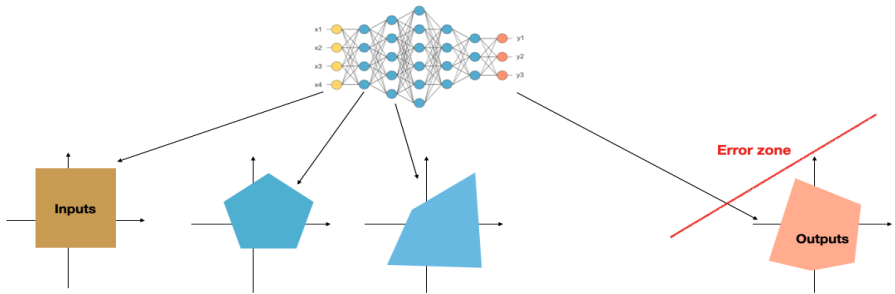
- Replace a large number of simulations by set-based simulation, using “tractable sets" such as boxes, zonotopes etc.
- Not possible to compute exact envelopes : compute inner and outer approximations
- Based on (local) functional approximations of solutions, e.g. Taylor models

Probabilistic analysis of CPS

Reachability Analysis for Neural Network Verification

Robustness and input/output properties:

- Need to be proved for (possibly large) sets of network inputs
- Can be specified as preconditions/postconditions expressed in linear arithmetic



Qualitative verification: property proven true or unknown

Quantitative Neural Network Verification

Motivation

- Provide additional information on property satisfaction compared to SAT/UNKNOWN
- Exploit knowledge of probabilistic information on inputs
  - can be probabilistic but imprecisely known, e.g.:
    - Gaussian variable $\mathcal{N}(\mu, \sigma^2)$ with uncertain mean $\mu \in [\underline{\mu}, \overline{\mu}]$ and variance $\sigma^2 \in [\underline{\sigma^2}, \overline{\sigma^2}]$
    - Uniform variable $\mathcal{U}(a, b)$ with uncertain range (a and b uncertain)
  - example: noise due to sensor $V + \varepsilon$ with $V \in [a, b]$, ε a random variable

With respect to most closely related work: Quantitative verification for neural networks using Probstars, Tran, H.D., Choi, S., Okamoto, H., Hoxha, B., Fainekos, G., Prokhorov, D., HSCC 2023

- inputs are arbitrary distributions (extending the Gaussian distribution hypothesis)
- our approach gives fully guaranteed probability bounds

Problem Statement: propagating imprecise probabilities

Problem (Probability bounds analysis)

Given a ReLU network f and a constrained probabilistic input set

$$\mathcal{X} = \{X \in \mathbb{R}^{h_0} \mid CX \leq d \wedge \underline{F}(x) \leq \mathbb{P}(X \leq x) \leq \overline{F}(x), \forall x\}$$

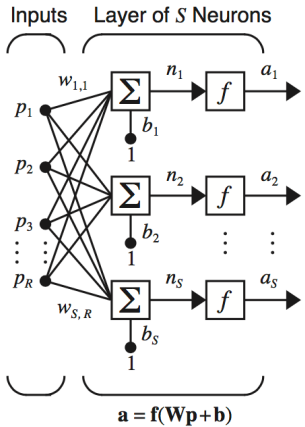
where $\underline{F}$ and $\overline{F}$ are two cumulative distribution functions, compute a constrained probabilistic output set $\mathcal{Y}$ guaranteed to contain $\{f(X), X \in \mathcal{X}\}$.
For $X \in \mathbb{R}^n$, we note $\mathbb{P}(X \leq x) := \mathbb{P}(X_1 \leq x_1 \wedge X_2 \leq x_2 \ldots \wedge X_n \leq x_n)$

Problem (Quantitative property verification)

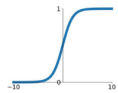
Given a ReLU network f, a constrained probabilistic input set $\mathcal{X}$ and a linear safety property $Hy \leq w$, bound the probability of the network output vector y satisfying this property.

Feedforward ReLU neural network

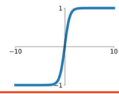
Each layer consists in a linear transform followed by a non linear activation function:



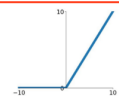
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



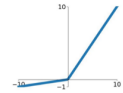
tanh
$\tanh(x)$



ReLU
$\max(0, x)$

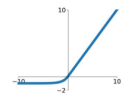


Leaky ReLU
$\max(0.1x, x)$



Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

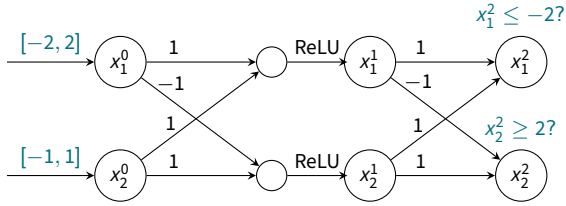


We focus, in the first part, on the classical ReLU activation function

Toy illustrating example: 2-layers ReLU network

$$A_1 = A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1. \end{bmatrix}, b_1 = b_2 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}.$$

$$\begin{aligned} x^1 &= \sigma(A_1 x^0 + b_1) = \sigma(x_1^0 - x_2^0, x_1^0 + x_2^0) \\ x^2 &= A_2 x^1 + b_2 \end{aligned}$$



Property:

- Qualitative: if $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$, does output satisfy $x_1^2 \leq -2 \wedge x_2^2 \geq 2$?
- Quantitative:
  - $\mathbb{P}\left(x_1^2 \leq -2 \wedge x_2^2 \geq 2 \;\middle|\; x_1^0 \in \mathcal{U}(-2, 2) \wedge x_2^0 \in \mathcal{U}(-1, 1)\right)$?
  - $\mathbb{P}\left(x_1^2 \leq -2 \wedge x_2^2 \geq 2 \;\middle|\; x_1^0 \in \mathcal{N}(0, [0.5, 0.66]) \wedge x_2^0 \in \mathcal{N}([0, 1], 0.33)\right)$?

Outline

- Imprecise probabilities: P-boxes and Dempster-Shafer Interval Structures (DSI)
  - Representations of sets of probability distributions
  - Generalize both probabilistic and non deterministic (interval) computations
- ReLU neural network analysis by DSI
- Mitigating the wrapping effect of intervals using zonotopes
  - Probabilistic Zonotopes
  - Zonotopic Dempster-Shafer Structures (DSZ)
- Evaluation

A plain DSI approach

Representation of imprecise probabilities: P-box

Definition (P-box for a real-valued random variable X)

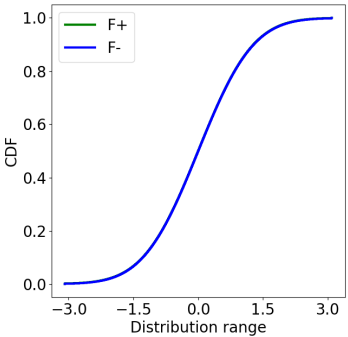
Given two (lower and upper) CDF (Cumulative Distribution Functions) $\underline{F}$ and $\overline{F}$ from $\mathbb{R}$ to $\mathbb{R}^+$ s.t. $\forall x \in \mathbb{R}, \underline{F}(x) \leq \overline{F}(x)$, the p-box $[\underline{F}, \overline{F}]$ represents the set of probability distributions for X s.t.

$$\forall x \in \mathbb{R}, \underline{F}(x) \leq \mathbb{P}(X \leq x) \leq \overline{F}(x).$$

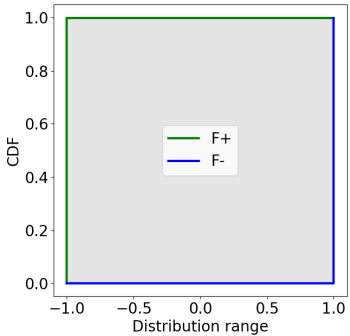
- Ferson S, Kreinovich V, Ginzburg L, Myers D, Sentz K, Constructing probability boxes and Dempster–Shafer structures. Tech. Rep. SAND2002-4015, 2003
- Williamson and Downs, Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds, Journal of Approximate Reasoning, 1990

P-box examples (Julia library ProbabilityBoundsAnalysis.jl)l

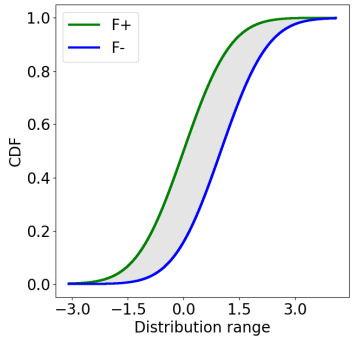
Sets of probability distributions on X (CDF form) such that $\forall x, F^{-}(x) \leq \mathbb{P}(X \leq x) \leq F^{+}(x)$:



normal(0,1)



makepbox(interval(-1,1))



normal(interval(0,1),1)

Generalize probabilistic and non deterministic (interval) information

Dempster-Shafer Interval structures (DSI)

A discrete version of P-boxes:

- Focal elements $t \in T$ (sets of values, here Intervals) with probability $w : T \to \mathbb{R}^+$

$t \in T$	[-1,0.25]	[-0.5,0.5]	[0.25,1]	[0.5,1]	[0.5,2]	[1,2]
$w(t)$	0.1	0.2	0.3	0.1	0.1	0.2

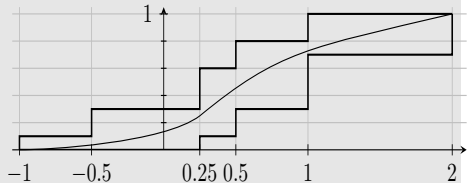
- Represents the set of probability distributions P on X such that:

$\forall x \in [-1, -0.5],\ P(X \leq x) \leq 0.1\,,$

$\forall x \in [-0.5, 0.25],\ P(X \leq x) \leq 0.1 + 0.2\,,$

$\forall x \in [0.25, 0.5],\ 0.1 \leq P(X \leq x) \leq 0.1 + 0.2 + 0.3\,,$

etc.

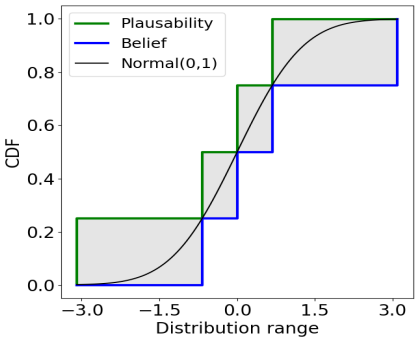


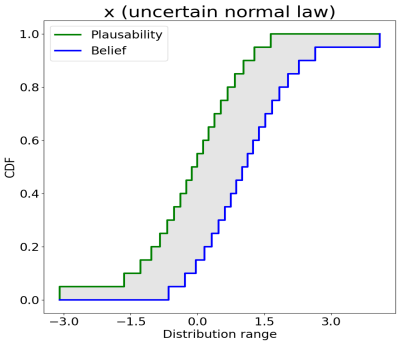
$$\sum_{t \in T, t \subseteq S} w(t) \leq P(S) \leq \sum_{t \in T, t \cap S \neq \emptyset} w(t)$$

From P-boxes to Dempster-Shafer Interval structures

Given a P-box $(\underline{F}, \overline{F})$

- Take lower and upper approximation by stair functions
- Deduce focal elements (intervals) and weights

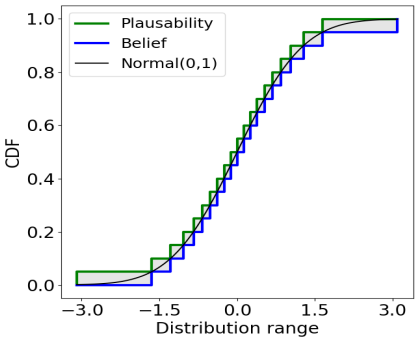


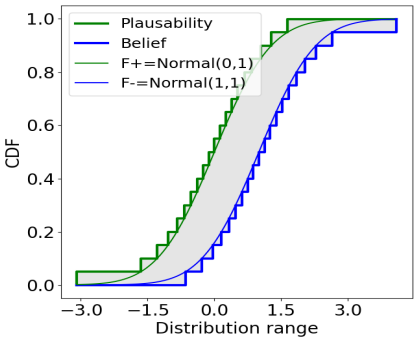


From P-boxes to Dempster-Shafer Interval structures

Given a P-box $(\underline{F}, \overline{F})$

- Take lower and upper approximation by stair functions
- Deduce focal elements (intervals) and weights

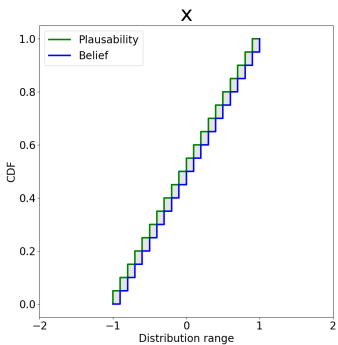


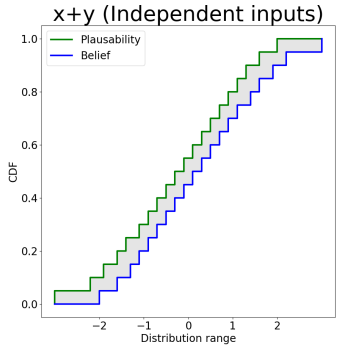


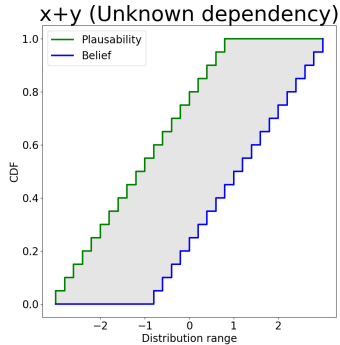
Arithmetic on DSI structures

DSI structures can be propagated through arithmetic operations:

- 2 cases: independent inputs / unknown dependency
- relying on interval arithmetic / Frechet inequalities
- conservative approximations



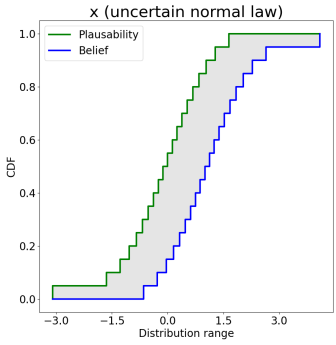


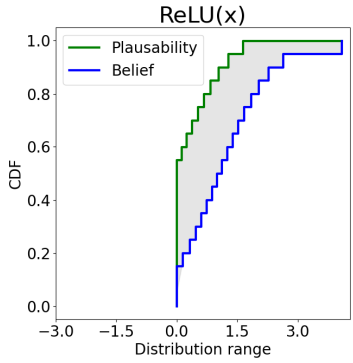


ReLU

Lemma (ReLU of a DSI)

Given X represented by the DSI $\{\langle \boldsymbol{x_i}, w_i \rangle\ ,\ i \in [1, n]\}$*, then the CDF of* $Y = \sigma(X) = \max(0, X)$ *is included in the DSI* $\{\langle \boldsymbol{y_i}, w_i \rangle\ ,\ i \in [1, n]\}$ *with* $y_i = [\max(0, \underline{x_i}), \max(0, \overline{x_i})]$.





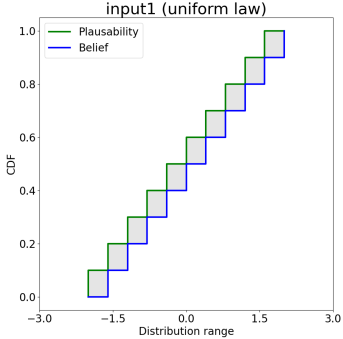
ReLU neural network analysis by DSI

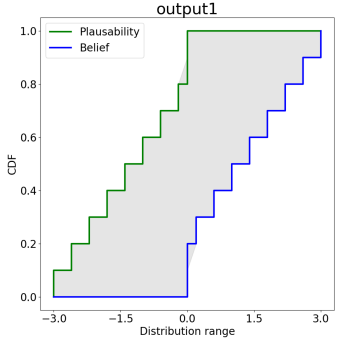
Input: d^0 a h_0-dimensional vector of DSI

1: **for** $k = 0$ to $L - 1$ **do**
2: **for** $l = 1$ to h_{k+1} **do**
3: $d_l^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} a_{lj}^k d_j^k + b_l^k)$ ▷ *Affine transform and ReLU - Dependency graph useful for choosing the right DSI operations (indep. or unknown dep.) in affine transforms*
4: **end for**
5: **end for**
6: **return** $(d^L, cdf(Hd^L, w))$ ▷ *Vector of DSI for the output layer and probability bounds for property* $Hz \leq w$

Illustration on the toy example

Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Uniform law on inputs





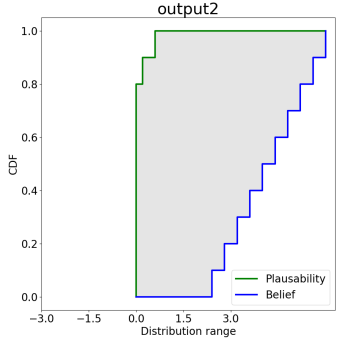
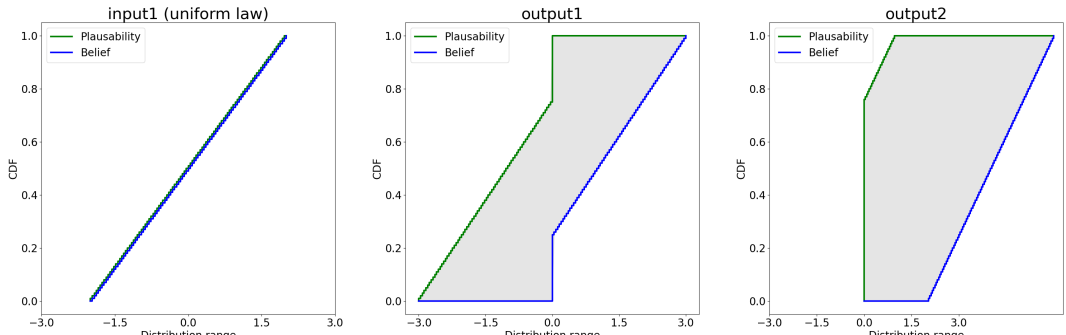


Illustration on the toy example

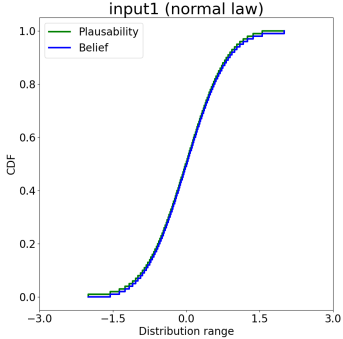
Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Uniform law on inputs

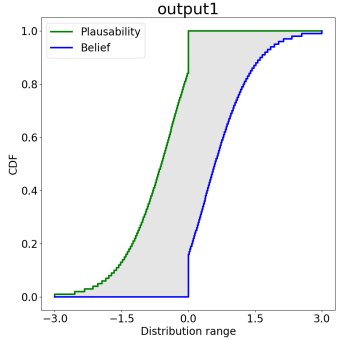


Finer discretization refines the approximation but the ranges are unchanged

Illustration on the toy example

Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Normal law on inputs





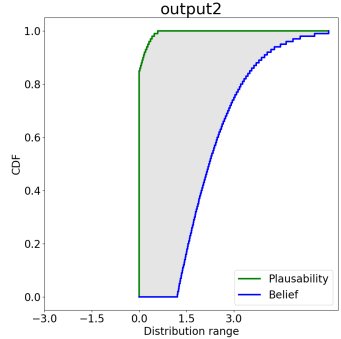
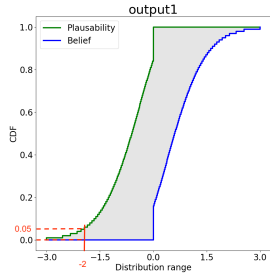
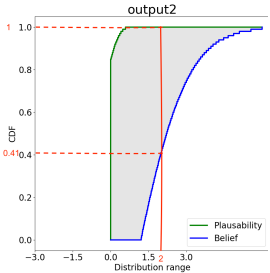
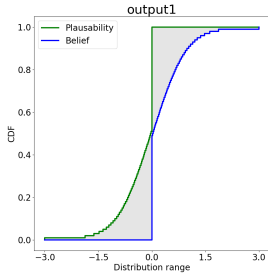


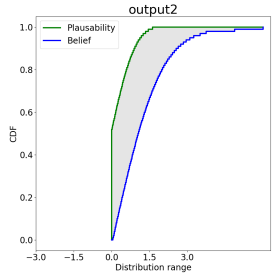
Illustration on the toy example

Unknown dependency on inputs vs independent inputs





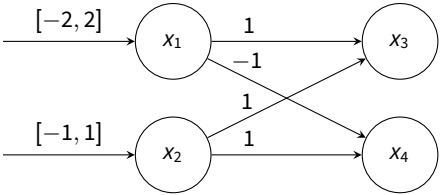




$\mathbb{P}(z_1 \leq -2) \in [0, 0.05]$ $\mathbb{P}(z_2 \geq 2) \in [0, 0.59]$ $\mathbb{P}(z_1 \leq -2) \in [0, 0.01]$ $\mathbb{P}(z_2 \geq 2) \in [0, 0.2]$

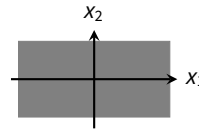
From interval to zonotopic focal elements

Wrapping effect: example of the first affine layer



Initial domain:

$$-2 \leq x_1 \leq 2$$
$$-1 \leq x_2 \leq 1$$

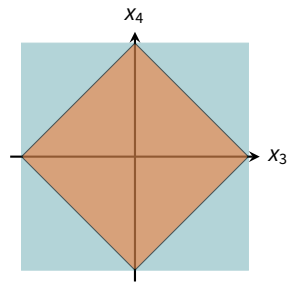


Exact domain:

$$x_3 = x_1 - x_2$$
$$x_4 = x_1 + x_2$$
$$x_1, x_2 \in [-1, 1]$$

Using Intervals/Boxes:

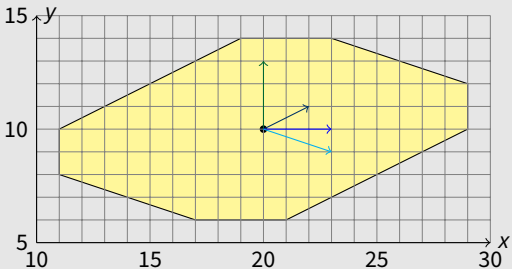
$$-3 \leq x_3 \leq 3$$
$$-3 \leq x_4 \leq 3$$
$$x_1, x_2 \in [-1, 1]$$



The optimal affine transformers for boxes are not exact. Zonotope transformers are !

Zonotopes?

Sub-polyhedric domain of zonotopes (center-symmetric with center-symmetric faces)



Affine image of (higher-dim) hypercube, here:

$$\begin{cases} x = 20 + 4\epsilon_1 + 2\epsilon_3 + 3\epsilon_4 \\ y = 10 - 2\epsilon_1 + \epsilon_2 - \epsilon_4 \end{cases}$$

for $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4 \in [-1, 1]$.

See E. Goubault, S. Putot: "A zonotopic framework for functional abstractions". Formal Methods Syst. Des. 2015 ; one of the bases of FLUCTUAT

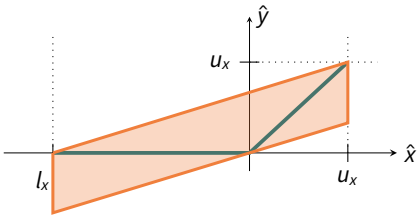
Zonotopes and neural network reachability analysis

Definition (Zonotope)

An n-dimensional zonotope $\mathcal{Z}$ with center $c \in R^n$ and a vector $\Gamma = \begin{bmatrix} g_1 \dots g_p \end{bmatrix} \in \mathbb{R}^{n,p}$ of p generators $g_j \in \mathbb{R}^n$ for $j = 1, \dots, p$ is defined as $\mathcal{Z} = \langle c, \Gamma \rangle = \{c + \Gamma\varepsilon \mid \|\varepsilon\|_\infty \leq 1\}$.

Zonotopes are closed under affine transformations: for $A \in \mathbb{R}^{m,n}$ and $b \in \mathbb{R}^m$ we define $A\mathcal{Z} + b = \langle Ac + b, A\Gamma \rangle$ as the m-dimensional resulting zonotope.

RELU transformer: conservative approximation



When $l_x \leq 0$ and $u_x \geq 0$:

$$\hat{y} = \lambda\hat{x} - \frac{\lambda l_x}{2} - \frac{\lambda l_x}{2}\varepsilon_{new} \text{ with } \lambda = \frac{u_x}{u_x - l_x}.$$

Two solutions for zonotopic probabilistic NN analysis

Probabilistic zonotopes (or probabilistic affine forms)

- Zonotopic network analysis starting from the support of input distribution
- Probabilistic interpretation: noise symbols are DSI instead of intervals
- inspired from [Adje et al 2013] A. Adjé, O. Bouissou, J. Goubault-Larrecq, E. Goubault, S. Putot: Static Analysis of Programs with Imprecise Probabilistic Inputs. VSTTE 2013: 22-47

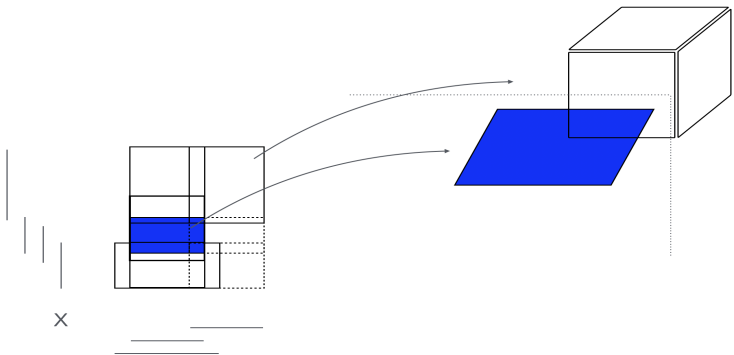
Dempster-Shafer Zonotopic structures (DSZ)

- Dempster-Shafer structures with zonotopic focal elements
- A refinement of probabilistic zonotopes, which fully exploits the DSI input discretization in the NN analysis
- In part 1, restricted to independent inputs

Principle of the zonotopic analysis

Propagate zonotopes through the different linear and ReLU layers

- Starting with DSIs on each input neuron (supposed independent for now)
- And producing initial zonotopes $\mathcal{Z}^0_{\dots}$ product of all focal elements of the h_0 input DSIs



NN analysis by DSZ (independent inputs)

Input: d^0 a h_0-dimensional vector of DSI

1: $d^0_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}^0_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle, (i_1, \dots, i_{h_0}) \in [1,n]^{h_0} \right\} \leftarrow$ dsi-to-dsz(d^0)
2: **for** $k = 0$ to $L-1$ **do**
3: **for** $(i_1, i_2, \dots, i_{h_0}) \in [1,n]^{h_0}$ **do**
4: $\mathcal{Z}^{k+1}_{i_1 \dots i_{h_0}} \leftarrow \sigma(A^k \mathcal{Z}^k_{i_1 \dots i_{h_0}} + b^k)$ ▷ Independent zonotopic analyzes (can be done in parallel)
5: **end for**
6: **end for**
7: $d^L_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}^L_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle, (i_1, \dots, i_{h_0}) \in [1,n]^{h_0} \right\}$
8: $d^L \leftarrow$ dsz-to-dsi($d^L_{\mathcal{Z}}$)
9: **return** (d^L,cdf($(Hd^L_{\mathcal{Z}}, w)$)) ▷ Property bounds computed by direct evaluation of the CDF on the zonotopic focal elements

Implementation and Evaluation

Julia implementation

- available from `https://github.com/sputot/DSZAnalysis` or `https://doi.org/10.5281/zenodo.12519084`.
- uses the LazySets and the NeuralVerification package for zonotopic NN analysis
- uses the ProbabilityBoundsAnalysis package for P-boxes / DSI analysis

Examples and evaluation

- Toy example
- ACAS Xu airplanes collision avoidance example
- Rocket lander example

Comparing DSI, Prob. Zonotopes and DSZ: toy example

Table 1: Probability bounds for the toy example, independent inputs.

Law	DSI			Prob. Zono.			DSZ		
(#FE)	$\mathbb{P}(x_1^2 \leq -2)$	$\mathbb{P}(x_2^2 \geq 2)$	time	$\mathbb{P}(x_1^2 \leq -2)\mathbb{P}(x_2^2 \geq 2)$		time	$\mathbb{P}(x_1^2 \leq -2)$	$\mathbb{P}(x_2^2 \geq 2)$	time
$U(2)$	$[0, 0.5]$	$[0, 1]$	$< e^{-3}$	$[0, 0.5]$	$[0, 1]$	$< e^{-3}$	$[0, 0.25]$	$[0, 0.5]$	$< e^{-3}$
$U(10)$	$[0, 0.2]$	$[0, 0.7]$	e^{-3}	$[0, 0.3]$	$[0, 0.8]$	e^{-3}	$[0, 0.03]$	$[0.2, 0.3]$	$< e^{-3}$
$U(10^2)$	$[0, 0.07]$	$[0.05, 0.52]$	0.022	$[0, 0.26]$	$[0, 0.76]$	0.013	$[0, 0.0014]$	$[0.25, 0.26]$	0.026
$U(10^3)$	$[0, 0.063]$	$[0.062, 0.502]$	2.4	$[0, 0.251]$	$[0, 0.751]$	1.2	$[0, 3.e^{-6}]$	$[0.25, 0.251]$	3
$N(10)$	$[0, 0.017]$	$[0, 0.277]$	e^{-3}	$[0, 0.1]$	$[0, 1]$	e^{-3}	$[0, 0.01]$	$[0, 0.1]$	$< e^{-3}$
$N(10^2)$	$[0, 0.004]$	$[0, 0.186]$	0.022	$[0, 0.07]$	$[0, 0.94]$	0.013	$[0, 4.e^{-4}]$	$[0.06, 0.07]$	0.026
$N(10^3)$	$[0, 0.004]$	$[0.003, 0.182]$	2.4	$[0, 0.067]$	$[0, 0.934]$	1.2	$[6e^{-5}, 1.1e^{-4}]$	$[0.066, 0.067]$	3

- For independent inputs, DSZ always more precise.
- In our paper “A Zonotopic Dempster-Shafer Approach to the Quantitative Verification of Neural Networks", FM 2024, detailed calculation for the 3 approaches in the case of 2 focal elements.

Comparisons to the state of the art

[Tran et al 23] Quantitative Verification for Neural Networks using ProbStars, Tran et al, HSCC 2023

Examples

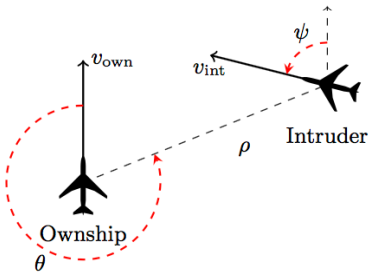
- ACAS Xu airplanes collision avoidance
- Rocket lander

Inputs and configuration

- Bounded (vector) inputs in [lb,ub], components follow independent Gaussian distributions with $\mu = (ub + lb)/2$ and $\sigma = (ub - m)/3$
- Timings and results given for [Tran et al 23] are from their paper:
  - parallelized (between 1 and 8 cores) and on a slightly stronger computer than ours
  - we reproduced a few analyzes: our machine runs ProbStar approx 7 to 10 times slower than their results with 1 core, approx 1.5 to 3.5 with 4 and 8 cores

ACAS Xu: collision avoidance systems for civil aircrafts (FAA)

- Produces aicraft advisory (clear-of-conflict, weak right, weak left, strong right, etc.)
- Array of 45 DNNs by discretizing τ and a_{prev} ; each has 5 inputs (ρ, θ, ψ, v_{own} and v_{int}) and 5 outputs (score for each advisory).
- Fully connected ReLU feeedforward networks with 5 inputs, 6 hidden layers, 5 outputs



Properties: $P_2 : y_1 > y_2 \wedge y_1 > y_3 \wedge y_1 > y_4 \wedge y_1 > y_5$
$P_3/P_4 : y_1 < y_2 \wedge y_1 < y_3 \wedge y_1 < y_4 \wedge y_1 < y_5$

Reluplex : An Efficient SMT Solver for Verifying Deep Neural Networks, Katz et al, CAV 2017.

Comparing DSZ and ProbStars Prob. bounds on ACAS Xu

- (Manual) Input discretization: [5, 80, 50, 6, 5] for P_2, [5, 20, 1, 6, 5] for P_3 and P_4

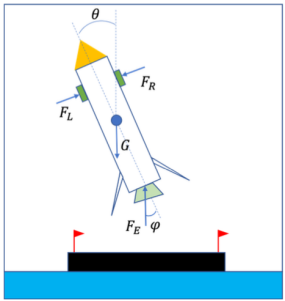
Prop	Net	DSZ		Probstar $p_f = e^{-5}$		Probstar $p_f = 0$	
		$\mathbb{P}$	time	$\mathbb{P}$	time	$\mathbb{P}$	time
2	1-6	[0, 0.01999]	46.4	[2.8e-06,0.05283]	206.7	1.87224e-05	1424
2	2-2	[0.00423 0.0809]	47.9	[0.0195,0.094]	299.0	0.0353886	2102.5
2	2-9	[0, 0.0774684]	51.0	[0.000255,0.107]	504.5	0.000997678	4561.2
2	3-1	[0.0165, 0.08787]	43.8	[0.0305, 0.07263]	202.7	0.044535	1086.4
2	3-6	[0.0167, 0.1111]	52.4	[0.02078,0.1069]	452.0	0.0335763	5224.4
2	3-7	[6e-05, 0.1361]	43.7	[0.002319,0.075]	331.1	0.00404731	2598
2	4-1	[1e-05, 0.05353]	40.9	[0.00104,0.07162]	305.3	0.00231247	1870.7
2	4-7	[0.0129, 0.1056]	44.4	[0.02078,0.1081]	418.9	0.04095	3407.8
2	5-3	[0, 0.03939]	40.0	[1.59e-09,0.0326]	139.7	1.81747e-09	418.8
3	1-7	[1, 1]	0.25	[0.9801,0.9804]	4.7	0.976871	3.6
4	1-9	[1, 1]	0.2	[0.9796,0.98]	3.6	0.989244	3.6

Comparing DSZ and ProbStars Prob. bounds on ACAS Xu

- (Manual) Input discretization: [5, 80, 50, 6, 5] for P_2, [5, 20, 1, 6, 5] for P_3 and P_4
- Basic (sensitivity-based) discretization refinement algo: P_2, net 1-6, refining [1,1,1,1,1] to ensure width(prob interval)$\leq$ 0.05: 112s and [5, 81, 38, 5, 5], probability in [0, 0.0276]

Prop	Net	DSZ		Probstar $p_f = e^{-5}$		Probstar $p_f = 0$	
		$\mathbb{P}$	time	$\mathbb{P}$	time	$\mathbb{P}$	time
2	1-6	[0, 0.01999]	46.4	[2.8e-06,0.05283]	206.7	1.87224e-05	1424
2	2-2	[0.00423 0.0809]	47.9	[0.0195,0.094]	299.0	0.0353886	2102.5
2	2-9	[0, 0.0774684]	51.0	[0.000255,0.107]	504.5	0.000997678	4561.2
2	3-1	[0.0165, 0.08787]	43.8	[0.0305, 0.07263]	202.7	0.044535	1086.4
2	3-6	[0.0167, 0.1111]	52.4	[0.02078,0.1069]	452.0	0.0335763	5224.4
2	3-7	[6e-05, 0.1361]	43.7	[0.002319,0.075]	331.1	0.00404731	2598
2	4-1	[1e-05, 0.05353]	40.9	[0.00104,0.07162]	305.3	0.00231247	1870.7
2	4-7	[0.0129, 0.1056]	44.4	[0.02078,0.1081]	418.9	0.04095	3407.8
2	5-3	[0, 0.03939]	40.0	[1.59e-09,0.0326]	139.7	1.81747e-09	418.8
3	1-7	[1, 1]	0.25	[0.9801,0.9804]	4.7	0.976871	3.6
4	1-9	[1, 1]	0.2	[0.9796,0.98]	3.6	0.989244	3.6

Rocket lander



- feedforward neural networks with 9 inputs, 3 outputs, and 5 hidden layers with 20 ReLU neurons per layer
- P_1: when $-20^\circ \leq \theta \leq -6^\circ, \omega < 0\,, \phi' \leq 0, F_S' \leq 0$, the output action should be $\phi < 0$ or $F_S < 0$: the agent should prevent the rocket from tilting to the right. (P_2 similar)

Neural Network Repair with Reachability Analysis. Yang et al, FORMATS 2022

Quantitative Verification for Neural Networks using ProbStars, Tran et al, HSCC 2023

Comparing DSZ and ProbStars Prob. bounds: rocket lander

- Input discretization: [7, 12, 10, 17, 9, 7, 1, 1, 2, 1, 1]

Prop	Net	DSZ		Probstar $p_f = 1e-5$		Probstar $p_f = 0$	
		$\mathbb{P}$	time	$\mathbb{P}$	time	$\mathbb{P}$	time
1	0	[0, 0.03387]	77.8	[4.15e-09, 0.06748]	1158.6	7.978e-08	5903.7
2	0	[0, 0.01352]	83.7	[0,0.1053]	2216	0	13132.7
1	1	[0, 0.01985]	80.5	[0,0.0536]	1229.7	8.68e-08	5163.9
2	1	[0, 0.00055]	69.1	[0, 0.0161751]	448.5	0	1495.6

From independent input to general dependencies

Multi-dimensional distributions

Univariate limitations:

- In the DSI approach we propagate a **vector of marginal DSIs**.
- This **ignores dependence** between features.
- NN layers (especially affine layers) also **create strong dependencies** between neurons and intermediate variables.

Goal

Track a **set of joint distributions** on $X = (X_1, \ldots, X_n)$ compatible with the marginals, and soundly propagate it through the network to obtain tighter bounds on safety properties.

This will allow for the verification of multivariate properties on the outputs of a NN.

Copulas

Separating marginals and dependence

For a random vector $X = (X_1, \ldots, X_n)$ with joint CDF F_X and marginals F_{X_i}, Sklar's theorem gives

$$F_X(x_1, \ldots, x_n) = C\big(F_{X_1}(x_1), \ldots, F_{X_n}(x_n)\big),$$

where $C : [0,1]^n \to [0,1]$ is a **copula encoding dependence independently from the marginals**.

Input model for propagation:

- We take **imprecise marginals** to encode uncertainty: each $F_{X_i} \in [\underline{F}_{X_i}, \overline{F}_{X_i}]$ (P-box / DSI).
- We take **precise dependence** and given by a known copula between the input feature's DSIs.

Copula volumes on DSI's quantiles

Each marginal DSI induces a **discretization of the unit interval via its quantiles**. A focal element of the DSIs corresponds to a quantile interval on the CDF space: $[\alpha^i_{j-1}, \alpha^i_j]$.

The copula assigns a **probability volume** to each multivariate rectangle from joining focal elements of different marginals:

$$R = [\alpha^1_{j_1-1}, \alpha^1_{j_1}] \times \cdots \times [\alpha^n_{j_n-1}, \alpha^n_{j_n}]$$

which represents the joint weight of the corresponding DSI focal elements. Collectively, these joint probability masses form a measure on rectangles that captures the joint dependence structures compatible with the marginal DSIs.

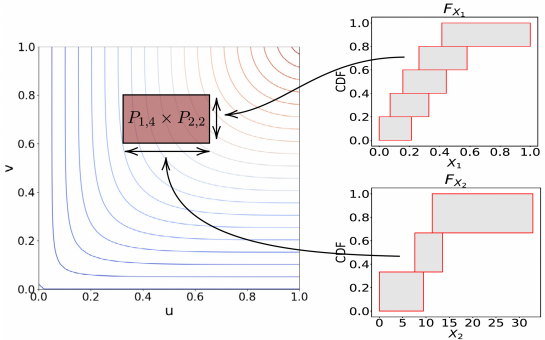


Figure 1: Diagram for the calculation of the copula volume of two coupled DSIs. Figure from Gray et al.

Gray, A., Forets, M., Schilling, C., Ferson, S., Benet, L.: Verified propagation of imprecise probabilities in non-linear ODEs. International Journal of Approximate Reasoning 164, 109044 (Jan 2024).

DSI + copula arithmetic: affine

Marginal affine propagation

Let $Y = AX + b$ with $A \in \mathbb{R}^{m \times n}$. Each marginal $Y_i = a_i X + b_i$ is obtained by affine arithmetic on the input DSIs. For each component X_i, let $d_{X_i} = \{\langle \mathbf{x}_{j_i}^i, w_{j_i}^i \rangle\}$. New focal intervals arise from affine combinations of input intervals.

Combinations of focal elements are indexed by $(j_1, \ldots, j_n)$ and induce a quantile rectangle in copula space. The output focal element $\sum a_i \mathbf{x}_{j_i}^i + b_i$ gets its probability given by the copula volume $V_X(R)$.

DSI + copula arithmetic: affine

Affine push-forward

Each input quantile grid cell (R) maps to exactly one output focal interval per coordinate. This defines a memory index map for where each rectangle comes from for each component $L_k : \prod_i \{1, \dots, N_i\} \to \{1, \dots, M_k\}, \quad L(\mathbf{J}) = (L_1(\mathbf{J}), \dots, L_m(\mathbf{J}))$. $\mathbf{J} = (j_1, \dots, j_n)$ a multi-index for convenience.

We have an output quantile grid cell given by the marginals $U_\ell = [\gamma^1_{\ell_1 - 1}, \gamma^1_{\ell_1}] \times \cdots \times [\gamma^m_{\ell_m - 1}, \gamma^m_{\ell_m}]$ for which we push-forward the volume through the memory map as $V_{C_S}(U_\ell) = \sum_{J:\, L(\mathbf{J}) = \ell} V_{C_X}(R_\mathbf{J})$. The copula is reconstructed by cumulative sums at output grid-points $u^* = (\gamma^1_{\ell_1}, \dots, \gamma^m_{\ell_m})$ as $C_S(u^*) = \sum_{J:\, L_k(J) \le \ell_k \ \forall k} m_J$.

Takeaways

- Dependence is propagated by pushing forward probability mass on quantile cells, not analytically.
- The memory map L records how each input quantile contributes to output marginals, keeping track of the dependencies between layers.
- We formally recover a discrete copula. Exactness is guaranteed on the quantile grid. Outside the grid, copula evaluations would interpolation.

DSI + copula arithmetic: activations

Component-wise activations

Let $Y_i = \sigma(X_i)$ for $i = 1, \ldots, n$ and a monotone **injective** activation σ (e.g. sigmoid, tanh, leaky ReLU...). Then $d_{Y_i} = \sigma(d_{X_i})$ where the activation is applied to both bounds of each focal interval.

DSI + copula arithmetic: ReLU

Why ReLU is special

- Not injective: all $x \leq 0$ are mapped to 0. This breaks introduces probability mass atoms which break to uniqueness of the copula below them.
- The amount of probability mass given to the atoms can not be precisely determined with imprecise marginals.

ReLU-aware merging of focal elements

- DSI focal elements containing negative points or 0 are merged into a single focal element at 0, with weights summed.
- Focal elements entirely in $[0, +\infty)$ are propagated normally by push-forward.
- Focal elements crossing 0 are conservatively split into a negative part (merged at 0) and a positive part (pushed forward).

This merger gives an outer approximation of the marginals that guarantees that the copula will only be evaluated at well defined points of its domain.

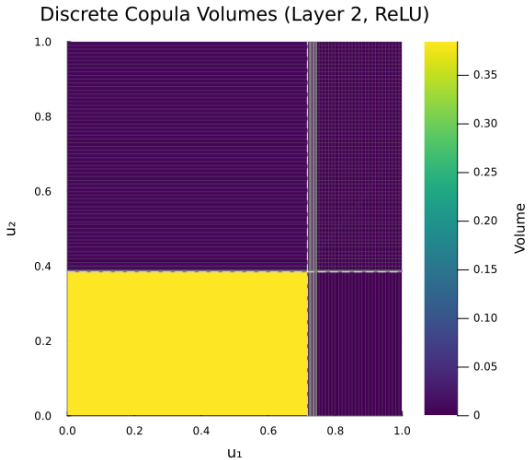


Figure 2: Output push-forwarded copula volumes after propagation through Layer 2 with a discretization size of 100.

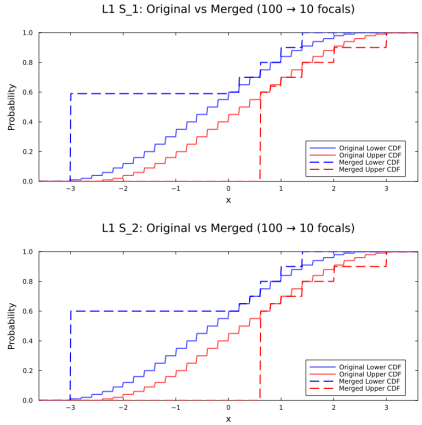


Figure 3: ReLU aware merging of focal elements after the first affine operation of the toy example.

DSI + copula arithmetic: activation

Activation push-forward

Let $Z = \sigma(X)$ with σ acting coordinate-wise and monotone. Each marginal DSI is propagated by image of focal intervals under σ_i, possibly followed by quantile-preserving merges (e.g. ReLU).

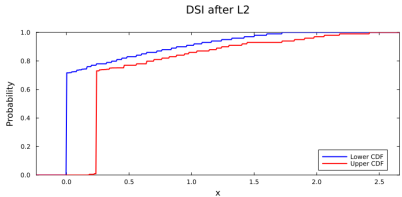
Each input quantile grid cell (R) maps to exactly one output focal interval per coordinate. This defines a memory index map $L_i : \{1, \dots, N_i\} \to \{1, \dots, M_i\}$ and $L(\mathbf{J}) = (L_1(j_1), \dots, L_n(j_n))$, with $\mathbf{J} = (j_1, \dots, j_n)$.

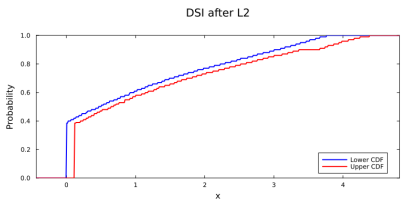
For an output quantile grid cell $U_{\boldsymbol{\ell}} = [\gamma^1_{\ell_1 - 1}, \gamma^1_{\ell_1}] \times \cdots \times [\gamma^n_{\ell_n - 1}, \gamma^n_{\ell_n}]$, we push forward the volume as $V_{C_Z}(U_{\boldsymbol{\ell}}) = \sum_{\mathbf{J}:\, L(\mathbf{J}) = \boldsymbol{\ell}} V_{C_X}(R_{\mathbf{J}})$. The copula is reconstructed at grid points $u^* = (\gamma^1_{\ell_1}, \dots, \gamma^n_{\ell_n})$ by $C_Z(u^*) = \sum_{\mathbf{J}:\, L_i(j_i) \le \ell_i \ \forall i} m_{\mathbf{J}}$.

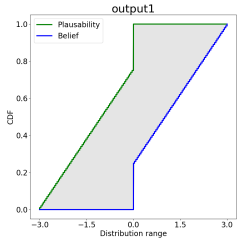
Takeaways

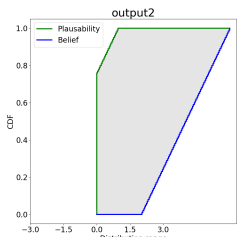
- Coupled DSI propagation handles arbitrary monotone injective activations naturally.
- The memory map L records how marginal quantile intervals are mapped by σ, including intermediate ReLU mergers.
- We recover a discrete copula, exact on the quantile grid. Non-injectivity is handled by conservatively coarsening the output quantile grid by mergers.

Toy example: coupled vs uncoupled DSIs









Toy example: copulas

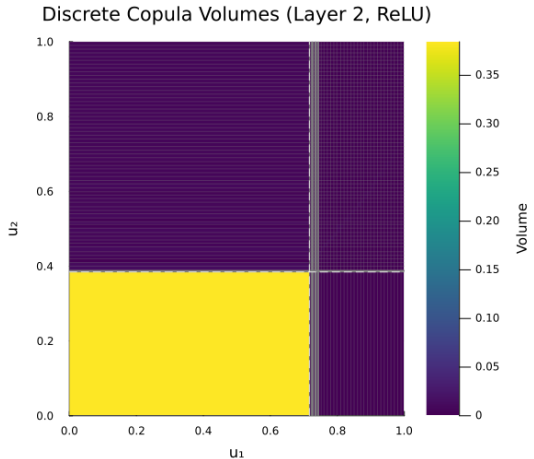


Figure 4: Output push-forwarded copula volumes after propagation through Layer 2 with a discretization size of 100.

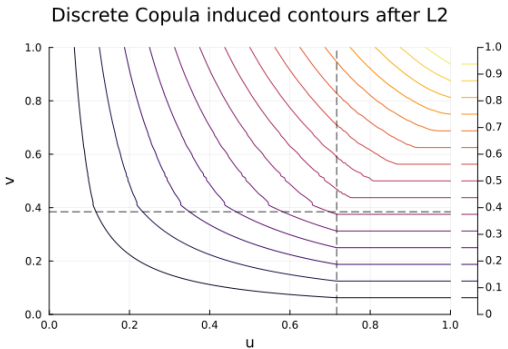


Figure 5: Reconstruction of the push-forwarded copula contours after propagation through Layer 2 with a discretization size of 100.

Toy example: benchmark

Table 2: Probability bounds for property $\mathbb{P}(x_2^2 \geq 2)$ on the toy example.

Law	Un-C. DSI		C. DSI		DSZ	
(#FE)	bound	time	bound	time	bound	time
$U(2)$	$[0, 1]$	$< e^{-3}$	$[0, 1]$	$< e^{-3}$	$[0, 0.5]$	$< e^{-3}$
$U(10)$	$[0, 0.7]$	$< e^{-3}$	$[0.2, 0.35]$	$< e^{-3}$	$[0.2, 0.3]$	$< e^{-3}$
$U(100)$	$[0.05, 0.52]$	0.016	$[0.23, 0.27]$	0.022	$[0.25, 0.26]$	0.015
$U(1000)$	$[0.062, 0.502]$	1.551	$[0.248, 0.252]$	3.43	$[0.25, 0.251]$	1.58
$N(10)$	$[0, 0.27]$	$< e^{-3}$	$[0, 0.98]$	$< e^{-3}$	$[0, 0.1]$	$< e^{-3}$
$N(100)$	$[0, 0.186]$	0.024	$[0.05, 0.23]$	0.017	$[0.06, 0.07]$	0.014
$N(1000)$	$[3e^{-3}, 0.182]$	1.586	$[0.064, 0.093]$	3.91	$[0.066, 0.067]$	1.69

Strengths and limitations compared to zonotopes

Strengths:

- Naturally extends to arbitrary (injective) activation functions.
- Recovers a copula between the marginals at each verification step, useful for interpretability purposes.
- Performance on par with DSZ.
- Allows for the evaluation of multivariate safety properties or joint probability for conjunctions of properties on the marginals.

Limitations:

- Interval affine arithmetic still suffers from the wrapping effect, which becomes worse with networks depth.
- Very conservative over-approximation required for ReLU to keep the copulas well-defined.
- Computationally more expensive than propagating zonotopes.

Real example: ETCS

Neural networks trained to replicate braking behavior of the European Train Control System. Sigmoid networks of increasing number of layers (1-8) trained on the same data.

Features:

- x_1: position of the first train.
- x_2: position of the second train.
- v: speed of the moving train.

One train is assumed to be stopped for simplicity.

Outputs:

- y_0: advise not to brake.
- y_1: advise to brake.

Braking property characterized by

$$y_0 \leq y_1 \iff y_0 - y_1 \leq 0.$$

Verification scenarios:

- Scenario A: unconstrained features. Ranges are: $x_1, x_2 \in [0, 50000], v \in [0, 150]$.
- Scenario B: imposing an order on the trains. Same ranges as scenario A with constraint $x_1 > x_2$.
- Scenario C: violation of safety distance between trains, and therefore crash, impossible. No reason to advise braking under ETCS rules. Modeled by $x_1 = 15000, x_2 = 35000$ and $v \in [25, 150]$.
- Scenario D: violation of safety distance between trains, and therefore braking, unavoidable. NNs should always advise braking under ETCS rules. Modeled by $x_1 \in [0, 800], x_2 \in [0, 800]$ and $v \in [25, 150]$.

Grundt, D., Jurj, S.L., Hagemann, W., Kröger, P., Fränzle, M.: Verification of Sigmoidal Artificial Neural Networks using iSAT. Electronic Proceedings in Theoretical Computer Science 361, 45–60 (Jul 2022).

Real example: ETCS

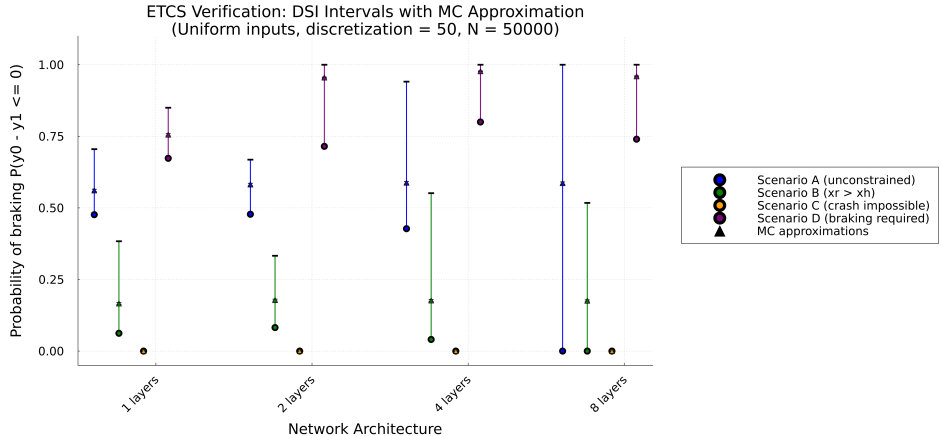


Figure 6: Quantitative verification results for probability of braking for the ETCS NNs obtained by propagating coupled DSI and modelling scenarios by precise uniform distributions.

Real example: ETCS

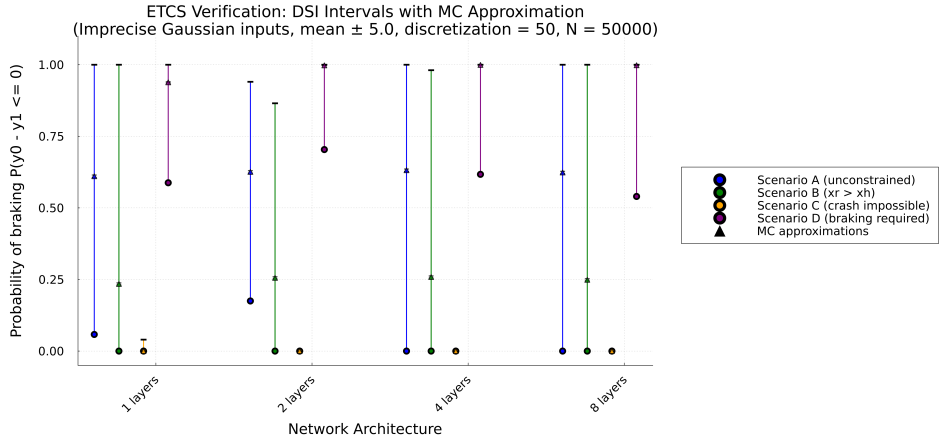


Figure 7: Quantitative verification results for probability of braking for the ETCS NNs obtained by propagating coupled DSI and modeling scenarios by imprecise Gaussian distributions.

Real example: ETCS

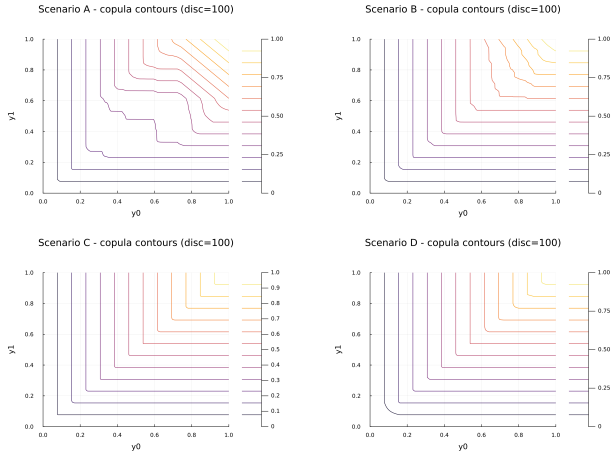


Figure 8: Copula contours of propagating the precise uniform modeling of scenarios A-D through the ETCS NN with 8 layers.

Future directions

- Extend to imprecise dependencies by using imprecise copulas.
- Extending copula advantages to the zonotopic domain for more precise arithmetic.
- Explore applications to robustness certification.
- Derive explicit input-output structures useful for sensitivity analysis.

Any questions? (`{goubault,putot,francesc.pifarre-esquerda@lix.polytechnique.fr}@lix.polytechnique.fr`)