A Zonotopic Dempster-Shafer Approach to the Quantitative Verification of Neural Networks

Eric Goubault, Sylvie Putot

A Zonotopic Dempster-Shafer Approach to the Quantitative Verification of Neural Networks

Eric Goubault and Sylvie Putot

LIX, CNRS and Institut Polytechnique de Paris, 91128 Palaiseau, France

Abstract. The reliability and usefulness of verification depend on the ability to represent appropriately the uncertainty. Most existing work on neural network verification relies on the hypothesis of either set-based or probabilistic information on the inputs. In this work, we rely on the framework of imprecise probabilities, specifically p-boxes, to propose a quantitative verification of ReLU neural networks, which can account for both probabilistic information and epistemic uncertainty on inputs. On classical benchmarks, including the ACAS Xu examples, we demonstrate that our approach improves the tradeoff between tightness and efficiency compared to related work on probabilistic network verification, while handling much more general classes of uncertainties on the inputs and providing fully guaranteed results.

1 Introduction

Verifying that neural networks satisfy desirable properties has become crucial for ensuring the safety of learning-enabled autonomous systems. However, most existing approaches that provide guarantees on the satisfaction of a specification are designed for adversarial input uncertainties, and offer only qualitative assessments. Complete methods return whether or not the property is satisfied, while sound methods return either that a property is satisfied or that the answer is unknown, due to over-approximation errors. For instance, the analyzes DeepZ [25], DeepPoly [26] and Verinet [11] propagate respectively zonotopes, polyhedra, and symbolic intervals through the layers of a neural network, to ensure that certain specifications are met. In addition to these specifications, many analyzers have considered producing also robustness bounds of networks, as specifically done by CROWN [33], FCROWN [14] and CNN-Cert [3].

In contrast, quantitative verification has been little explored for neural networks, despite providing a better understanding of the system by refining information about property satisfaction. This is especially true for probabilistic verification. Some authors have considered estimating the statistics of the output of neural networks, given a multivariate probabilistic law for its inputs. This approach has been used in particular for assessing the robustness of neural networks [29, 34] and for probabilistically certifying their correctness under adversarial attacks [6, 32, 18, 12]. But these estimates, using improved sampling methods, do not give any guaranteed bounds. Even fewer articles have considered guaranteed probabilistic bounds. In [30], the authors describe the analyzer

PROVEN, which provides probability certificates of neural network robustness when the input perturbation is given by a probabilistic distribution, based on the abstractions computed by Fast-Lin, CROWN and CNN-Cert. For networks with ReLU activation function, methods have been developed in [20] and [19] to find the probability of the output or of the input-output relationships. In [7], the authors consider an ellipsoid input space with Gaussian random variables and compute confidence ellipsoids for the outputs by propagating these ellipsoids in ReLU networks, using semidefinite programming. In [27], the authors consider truncated multivariate Gaussian distribution inputs and abstract them by probabilistic stars (ProbStars), a variation of the star set abstraction [1] recently introduced in the context of reachability analysis. They propagate them in a guaranteed manner in a network, and estimate the probability of violating a safety property on the output by computing each probstar's probability. In a way, ProbStar is a hybrid method, relying on guaranteed set-based computations, but estimating the probabilities in a non guaranteed manner.

The works mentioned above rely on the hypothesis of either set-based or probabilistic information on the inputs. However, in real-world systems, precise models representative of the data are not always available. For instance, several probabilistic models may be plausible for describing of a problem, or a probabilistic model may be known but with uncertain parameters. Therefore, we need to consider both aleatory information and epistemic uncertainty. Imprecise probabilities [28, 2] offer a framework that unifies probabilistic and set-based information. This framework includes a wide variety of mathematical models, among which probability boxes (p-boxes in short) [8], which characterize an uncertain random variable by all probability distributions consistent with lower and upper bounds on its cumulative distribution function (CDF in short). A p-box can be seen as interval bounds on a probability distribution. An Interval-based discrete over-approximation of p-boxes, Interval Demspter-Shafer structure [23] (DSI in short) has been proposed. Algorithms for arithmetic operations on DSI can be derived [8, 31], which can be seen as a unification of standard interval analysis with traditional probability theory, allowing probability bound analysis on arithmetic expressions. It gives the same answer as interval analysis does when only range information is available. And it gives sound bounds on the distribution function, as a sound counterpart of a Monte Carlo simulation, when information is precise enough to fully specify input distributions and their dependencies. However, DSI arithmetic is expensive and suffers from the conservativeness of interval arithmetic, on which it relies. Probabilistic affine forms have been proposed as an alternative [4, 5]. These forms combine affine forms or zonotopes and DSI structures, improving both precision and efficiency.

In this work, we first extend for the analysis of ReLU neural networks, the Interval Dempster Shafer arithmetic in Section 3 and probabilistic affine arithmetic in Section 4 and demonstrate their use on the quantitative verification of a small toy network. We then introduce in Section 5 a new abstraction, Zonotopic Dempster Shafer structures, which exhibits much better computational properties (complexity and tightness of the approximations). This new abstraction is

directly related to the general notion of random sets [22] which generalize one-dimensional Dempster-Shafer structures such as the DSI. Finally, in Section 6 we evaluate our approach and demonstrate that it improves in terms of tradeoff between tightness and efficiency compared to the most closely related work [27] on the ACAS Xu networks and on a neural network controller for a rocket lander benchmark for SpaceX Falcon 9, while being able to handle much more general classes of uncertainties on the inputs and providing fully guaranteed results.

2 Problem statement

We consider an L-layer feedforward ReLU network with input $x^0 \in \mathbb{R}^{h_0}$ and output $y = f(x^0) = x^L \in \mathbb{R}^{h_L}$, with f being the composition of L layers, $f = f^{L-1} \circ \ldots \circ f^0$. The k-th layer of the ReLU network is defined by $f^k : \mathbb{R}^{h_k} \to \mathbb{R}^{h_{k+1}}$ of the form $x^{k+1} = f^k(x^k) = \sigma(A^k x^k + b^k)$, where $A^k \in \mathbb{R}^{h_{k+1} \times h_k}$ is the weight matrix, $b^k \in \mathbb{R}^{h_{k+1}}$ is the bias, and $\sigma(x_j^k) := \max(0, x_j^k)$ is the component-wise ReLU function, where x_j^k is the jth component of $x^k \in \mathbb{R}^{h_k}$.

We are interested in the following two problems, extending, in particular to a larger class of inputs, the quantitative verification properties of [27]:

Problem 1 (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 $\{f(X), X \in \mathcal{X}\}$.

Problem 2 (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.

We will consider through the paper the toy example below to illustrate the different analyzes we propose.

Example 1. We consider the ReLU network defined by the matrices of weights and biases: $A_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1. \end{bmatrix}, b_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, b_2 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$. We take only one ReLU layer and an affine output layer. After the ReLU layer, we note $x^1 = \sigma(A_1 x^0 + b_1) = \sigma(x_1^0 - x_2^0, x_1^0 + x_2^0)$, and after the output layer $x^2 = A_2 x^1 + b_2$.

The problem is to verify the network against the unsafe output set $x_1^2 \leq -2 \wedge x_2^2 \geq 2$ for an input $x^0 = (x_1^0, x_2^0) \in [-2, 2] \times [-1, 1]$. This writes $Hx^2 \leq w$ with $H = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, w = \begin{bmatrix} -2 & -2 \end{bmatrix}$. All details on the analyzes of this example, which results are stated in further sections, are provided in Appendix A.

3 Analysis with Interval Dempster-Shafer structures

Probability-boxes and Interval Dempster-Shafer arithmetic We characterize a real-valued random variable X by its cumulative probability distribution function (CDF in short) $F : \mathbb{R} \to [0, 1]$ defined by $F(x) = \mathbb{P}(X \leq x)$. A p-box [8] is defined by a pair of CDF:

Definition 1 (P-box). *Given two CDF $\overline{F}, \underline{F}$, the p-box $[\underline{F}, \overline{F}]$ represents the set of distribution functions F such that $\underline{F}(x) \leq F(x) \leq \overline{F}(x)$ for all $x \in \mathbb{R}$.*

P-boxes can be combined in mathematical calculations, but analytical solutions are usually not available. Interval Dempster-Shafer structures [23] provide a simple way to soundly over-approximate the set of cdfs using a discrete representation, for which the arithmetic operations can be converted into a series of elementary interval calculations.

Definition 2 (Interval Dempster-Shafer structure). *An interval Dempster-Shafer structure (DSI in short) is a finite set of intervals, named focal elements, associated with a probability, written $d = \left\{\langle \boldsymbol{x_1}, w_1\rangle, \langle \boldsymbol{x_2}, w_2\rangle, \ldots, \langle \boldsymbol{x_n}, w_n\rangle\right\}$, where $\boldsymbol{x_i}$ is an interval and $w_i \in (0, 1]$ is its probability, with $\sum_{k=1}^{n} w_k = 1$.*

Proposition 1 (CDF of an Interval Dempster-Shafer structure). *A DSI $d = \left\{\langle \boldsymbol{x_1}, w_1\rangle, \langle \boldsymbol{x_2}, w_2\rangle, \ldots, \langle \boldsymbol{x_n}, w_n\rangle\right\}$ defines the discrete pbox $\left[\underline{F_d}, \overline{F_d}\right]$ representing the sets of distributions such that $\underline{F_d}(u) \leq \mathbb{P}(X \leq u) \leq \overline{F_d}(u)$ with $\underline{F_d}(u) = \sum_{\overline{\boldsymbol{x_i}} < u} w_i$ and $\overline{F_d}(u) = \sum_{\underline{\boldsymbol{x_i}} \leq u} w_i$.*

Conversely, discrete upper and lower approximations of distribution functions can be constructed, for instance using the inverse CDF as in [31]. Given a discretization size N, they define a DSI with N focal elements where all weights are equal to $1/N$. The focal elements $\boldsymbol{x_i}$ are defined evaluating the quantiles or inverse cdfs for uniformly spaced probability levels $p_i = \frac{i-1}{N}$ for $i = 1, \ldots, N+1$, by $\boldsymbol{x_i} = [\overline{F}^{-1}(p_i), \underline{F}^{-1}(p_{i+1})]$ where $F^{-1}(p) = \inf\{x \mid F(x) \geq p\}$.

The arithmetic operators on DSI structures [8, 31] compute guaranteed enclosures of all possible distributions of an output variable if the input p-boxes enclose the input distributions. Let two random variables X and Y represented by DSI structures $d_X = \{\langle \boldsymbol{x_i}, w_i\rangle,\ i \in [1, n]\}$ and $d_Y = \{\langle \boldsymbol{y_j}, w'_j\rangle,\ j \in [1, m]\}$, and Z be the random variable such that $Z = X + Y$ (the algorithms for other arithmetic operations are similar). In particular, they define algorithms for the extreme cases of unknown dependence and independence between X and Y.

Definition 3 (Probabilistic dependence and dependence graph). *Two random variables X_1 and X_2 are independent if and only if their CDF can be decomposed as $F(x_1, x_2) = F_1(x_1)F_2(x_2)$. Otherwise, the random variables are called correlated. The probabilistic dependence graph G over a set of n variables $X_1, \ldots, X_n$ is an undirected graph where the X_i are the vertices and there exists an edge (X_i, X_j) in the graph iff variables X_i and X_j are correlated.*

The addition of DSI independent variables is obtained as a discrete convolution of the two input distributions:

Definition 4 (Addition of independent DSIs). *If X and Y are independent random variables, then the DSI for $Z = X \oplus Y$ is $d_Z = \{\langle \boldsymbol{z_{i,j}}, r_{i,j}\rangle,\ i \in [1, n],\ j \in [1, m]\}$ such that: $\forall i \in [1, n],\ j \in [1, m],\ \boldsymbol{z_{i,j}} = \boldsymbol{x_i} + \boldsymbol{y_j}$ and $r_{i,j} = w_i \times w'_j$.*

The number of focal elements grows exponentially with the number of such operations. In order to keep the computation tractable, the number of focal elements is usually bounded, at the cost of some over-approximations.

Different algorithms have been proposed for the addition of DSIs with unknown dependence, relying on the Fréchet–Hoeffding copula bounds or on linear programming, most of them produce the same result [8, 31, 21].

DSI analysis of neural networks We now define a sound probability bounds analysis of ReLU neural networks.

Modelling the network inputs Consider an h_0-dimensional uncertain input vector $x^0 = (x_1^0, \ldots, x_{h_0}^0)$, which can be represented as a vector $d^0 = (d_1^0, \ldots, d_{h_0}^0)$ of h_0 DSI, each with the same number n of focal elements for simplicity of presentation: $d_i^0 = \{\left\langle \boldsymbol{x_{i,1}^0}, w_{i,1}^0 \right\rangle, \left\langle \boldsymbol{x_{i,2}^0}, w_{i,2}^0 \right\rangle, \ldots, \left\langle \boldsymbol{x_{i,n}^0}, w_{i,n}^0 \right\rangle\}$ for $i \in 1, \ldots, h_0$, where $\boldsymbol{x_{i,j}^0} \in \mathbb{IR}$ is an interval and $w_{i,j}^0 \in]0, 1]$ is the associated probability, with $\sum_{j=1}^{n} w_{i,j}^0 = 1$, for all $i \in 1, \ldots, h_0$. A dependence graph is assumed to be known between the components of the input vector.

Affine transform of a vector of DSI structures Given a vector of random variables $X = (X_1, \ldots, X_k)$ represented as a vector $d = (d_1, \ldots, d_k)$ of DSI structures, and a dependence graph G, we define a DSI $d^y = \sum_{j=1}^{k} a_j d_j + b$ which includes the result of $Y = \sum_{j=1}^{k} a_j X_j + b$ on the DSI d by:

- we note $a_j d_j$ where $a_j \in \mathbb{R}$ and d_j is a DSI $\{\langle \boldsymbol{x_{j,i}}, w_{j,i} \rangle \mid i = 1, \ldots, n\}$, the result of the multiplication of a constant by a DSI: $\{\langle a_j \boldsymbol{x_{j,i}}, w_{j,i} \rangle \mid i = 1, \ldots, n\}$,
- we arbitrary choose to compute the sum $\sum_{j=1}^{k} a_j d_j$ as $((a_1 d_1 + a_2 d_2) + a_3 d_3) + \ldots a_k d_k)$, applying for the j-th sum the right operators depending of the dependence between X_{j+1} and X_1 to X_j,
- we note $d + b$ where d is a DSI $\{\langle \boldsymbol{x_i}, w_i \rangle \mid i = 1, \ldots, n\}$ and $b \in \mathbb{R}$ the result of the addition of a constant to a DSI: $\{\langle b + \boldsymbol{x_i}, w_i \rangle \mid i = 1, \ldots, n\}$,
- the dependence graph is updated by adding an edge between Y and all X_j such that a_j is non zero

Interpreting the action of the ReLU function $Y = \max(0, X)$ means enforcing the constraints $Y \geq X$ and $Y \geq 0$. This means that Y is obtained by intersecting the focal elements of the representation of X with $[0, \infty)$:

Definition 5 (ReLU of a DSI). *Given a random variable X represented by the DSI $d = \{\langle \boldsymbol{x_i}, w_i \rangle, i \in [1, n]\}$, 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 $y_i = [\max(0, \underline{x_i}), \max(0, \overline{x_i})]$.*

This leads to Algorithm 1 for the analysis of an L-layer ReLU network with the notations of Section 2.

Algorithm 1 ReLU feedforward neural network analysis by DSI arithmetic

Input: d^0 a h_0-dimensional vector of DSI
for $k = 0$ to $L-1$ **do**
  for $l = 1$ to h_{k+1} **do**
    $d_l^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} a_{lj}^k d_j^k + b_l^k)$ ▷ Affine transform and Definition 5
  end for
end for
return (d^L,cdf(Hd^L, w))

Output The output after propagation in the network consists in:

- the vector of DSI d^L characterizing the network output (solving Problem 1)
- interval bounds noted $cdf(Hd^L, w)$ on the probability $\mathbb{P}(Hx^L) \leq w$ (solving Problem 2). Let $[\underline{P_m}, \overline{P_m}]$, with m ranging over the lines of H and w, be the interval for the probability $\mathbb{P}(\sum_{i=1}^{h_L} h_{mi} x_i^L \leq w_m)$. It is obtained applying Proposition 1 to compute the CDF at w on each component of the vector Hd^L. We define $cdf(Hd^L, w) = [\min_m \underline{P_m}, \min_m \overline{P_m}]$.

The DSI computation encodes the marginal distribution of each component of a vector x^i as a DSI. The probability of a conjunction Hx^L is thus computed considering each inequality independently and expressing that the probability of the conjunction is lower or equal than the probability of each term.

Analysis of the toy example Consider Example 1. A classical interval analysis of the network from the input set $x^0 = (x_1^0, x_2^0) \in [-2, 2] \times [-1, 1]$ yields the output ranges $x_1^2 \in [-3, 3]$ and $x_2^2 \in [0, 6]$. As these have non empty intersection with the property $x_1^2 \leq -2 \wedge x_2^2 \geq 2$, this analysis does not allow to conclude.

Uniform distribution on inputs abstracted by DSI with 2 focal elements Let us now suppose that we additionally know that the 2 components of the input follow a uniform distribution. We first choose a discretization of the inputs by DSI with 2 focal elements, $d_1^0 = \{\langle[-2, 0], 0.5\rangle, \langle[0, 2], 0.5\rangle\}$ and $d_2^0 = \{\langle[-1, 0], 0.5\rangle, \langle[0, 1], 0.5\rangle\}$. Let us suppose the inputs independent, the first output after the first affine layer, $d_{y_1} = d_1^0 - d_2^0$, computed following Definition 4, is $\{\langle[-2, 1], 0.25\rangle, \langle[-3, 0], 0.25\rangle, \langle[0, 3], 0.25\rangle, \langle[-1, 2], 0.25\rangle\}$. In order to limit the complexity of computation, the result of each operation on DSI can be reduced by a sound overapproximation with a fixed number of focal elements. This can be done by joining some focal elements and adding the corresponding weights. For instance here, when reducing to 2 focal elements by joining the first 2 and the last 2 focal elements, this results in $d_{y_1} = \{\langle[-3, 1], 0.5\rangle; \langle[-1, 3], 0.5\rangle\}$. Then, applying to d_{y_1} the ReLU function using Definition 5 produces $d_1^1 = \{\langle[0, 1], 0.5\rangle, \langle[0, 3], 0.5\rangle\}$. The other output x_2^1 of the first layer has the same DSI representation. After the output layer, the first output is $d_1^2 = d_1^1 - d_2^1 = \{\langle[-3, 1], 0.5\rangle, \langle[-1, 3], 0.5\rangle\}$. Here x_1^1 and x_2^1 can no longer be considered as independent as they both are correlated to x_1^0 and x_2^0, the subtraction of their DSI

representation is computed accordingly. The second output is $d_2^2 = d_1^1 + d_2^1 = \{\langle[0,4],0.5\rangle, \langle[0,6],0.5\rangle\}$.

Take now the property $x_1^2 \leq -2 \wedge x_2^2 \geq 2$. Using Proposition 1, we deduce from d_1^2 and d_2^2 that $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.5]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0.0, 1.0]$, from which $\mathbb{P}(x_1^2 \leq -2 \wedge x_2^2 \geq 2) \in [0, 0.5]$. Consider for instance $\mathbb{P}(x_1^2 \leq -2)$ evaluated using $d_1^2 = \{\langle[-3,1],0.5\rangle, \langle[-1,3],0.5\rangle\}$. Its lower bound is obtained using Proposition 1 by $\underline{P}(-2) = \sum_{\overline{\boldsymbol{x_i}} < -2} w_i = 0$, as the upper bounds of the 2 focal elements $[-3,1]$ and $[-1,3]$ are both greater than -2. The upper bound is $\overline{P}(-2) = \sum_{\underline{\boldsymbol{x_i}} \leq u} w_i = 0.5$, as the lower bound of $[-3,1]$ is lower than -2, which is not the case for $[-1,3]$.

Increasing the number of focal elements refines the over-approximation of the input distributions and the sets of CDF obtained for the outputs. For instance for 100 focal elements, in the case inputs can be considered as independent, we obtain $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.07]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0.05, 0.52]$. In the case of inputs with unknown correlation, $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.26]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 1]$. However, the supports of the sets of distribution remain unchanged and are equal to the ranges obtained through interval analysis. Indeed, the affine layers introduce some conservatism due to the wrapping effect of the intervals used as focal elements. Additionally, joint distribution are not naturally represented in the DSI framework, making it difficult to accuractely verify general properties.

4 Analysis with probabilistic zonotopes

Probabilistic affine forms [4, 5] are affine forms where the symbolic variables or noise symbols are constrained by DSI structures instead of being simply bounded in [-1,1]. This can be seen as a simple way to encode affine correlations between uncertain variables abstracted by p-boxes, or a quantitative version of affine forms. We first briefly introduce the probabilistic affine forms, presented under the form of probabilistic zonotopes, which represent vectors of probabilistic affine forms. Then we propose an analysis of neural networks relying on these probabilistic zonotopes.

Affine forms, zonotopes and probabilistic zonotopes An affine form is a linear expression $\alpha_0 + \sum_{j=1}^{p} \alpha_j \varepsilon_j$ with real coefficients α_j and symbolic variables ε_j called noise symbols which values range in [-1,1]. A zonotope is the geometric concretization of a vector of affine forms:

Definition 6 (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 $\mathcal{Z} = \langle c, \Gamma \rangle = \{c + \Gamma\varepsilon \mid \|\varepsilon\|_\infty \leq 1\}$.*

We note $\gamma_i(\mathcal{Z}) = c_i + \sum_{j=1}^{p} g_{ij}[-1,1]$ the range of its i-th component.

Zonotopes are closed under affine transformations:

Proposition 2 (Affine transforms of a zonotope). *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.*

Definition 7 (Probabilistic Zonotope). *For ε a vector of random variables of $\mathbb{R}^p$, a zonotope $\mathcal{Z} = \langle c, \Gamma \rangle$ with $c \in \mathbb{R}^n$ and $\Gamma \in \mathbb{R}^{n,p}$ can be interpreted as a probabilistic zonotope noted $p\mathcal{Z}(\varepsilon) = \langle c, \Gamma, \varepsilon \rangle$ representing the $n-$dimensional random variable $Z = c + \Gamma\varepsilon$. Let d_ε be a $p-$dimensional vector of DSI structures with support in $[-1,1]^p$ and G a dependence graph on the components $\varepsilon_1, \ldots, .\varepsilon_p$. The marginal of each component of $p\mathcal{Z}(d_\varepsilon)$ is the affine transform on DSI structures: $c^i + \sum_{j=1}^p g_{ij} d_{\varepsilon_j}$, $i = 1, \ldots, n$ computed as in Section 3.*

Zonotopes represent affine relations that hold between uncertain quantities. In the case of probabilistic zonotopes, *imprecise* affine relations hold:

Example 2. Let $x_1 = 1 + \varepsilon_1 - \varepsilon_2$, $x_2 = -\frac{1}{2}\varepsilon_1 + \frac{1}{4}\varepsilon_2$, $d_{\varepsilon_1} = \{\langle [-1, 0], \frac{1}{2}\rangle, \langle [0, 1], \frac{1}{2}\rangle\}$, $d_{\varepsilon_2} = \{\langle [-\frac{1}{10}, 0], \frac{1}{2}\rangle, \langle [0, \frac{1}{10}], \frac{1}{2}\rangle\}$, Then $x_1 + 2x_2 = 1 - \frac{1}{2}\varepsilon_2$, with $d = d_{x_1+2x_2} = \{\langle [\frac{19}{20}, 1], \frac{1}{2}\rangle, \langle [1, \frac{21}{20}], \frac{1}{2}\rangle\}$. Thus the lower probability that $x_1 + 2x_2 \leq \frac{21}{20}$ is 1; and the upper probability that $x_1 + 2x_2 < \frac{19}{20}$ is 0. But for instance, $x_2 + 2x_2 \leq 1$ has upper probability $\frac{1}{2}$ and lower probability 0 and is thus an imprecise relation.

Probabilistic zonotopes for the analysis of neural networks Algorithm 2 defines a neural network analysis using probabilistic zonotopes, which we detail in this section.

Algorithm 2 Neural network analysis by Probabilistic Zonotopes

Input: d^0 a h_0-dimensional vector of DSI
1: $p\mathcal{Z}^0(\varepsilon) = \langle c^0, \Gamma^0, d_\varepsilon \rangle \leftarrow$ dsi-to-pzono(d^0)
2: **for** $k = 0$ to $L - 1$ **do**
3: $\quad \mathcal{Z}^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} A^k \mathcal{Z}^k + b^k)$ ▷ Proposition 2 and Proposition 3
4: **end for**
5: $d^L \leftarrow$ pzono-to-dsi($\mathcal{Z}^L, d_\varepsilon$) ▷ Definition 7
6: **return** (d^L,cdf(pzono-to-dsi($H\mathcal{Z}^L, d_\varepsilon$), w))

Input and initialization The input of the algorithm is the same as in Section 3, the uncertain input is modelled as a vector $d^0 = (d_1^0, \ldots, d_{h_0}^0)$ of h_0 DSI. We can then define $\boldsymbol{x}^0 \in \mathbb{IR}^{h_0}$ the h_0-dimensional box obtained as the support of d^0, computed for each DSI as the union of its focal elements with non-zero weight. Finally, we define $p\mathcal{Z}^0(\varepsilon) = \langle c^0, \Gamma^0, d_\varepsilon \rangle$ in Line 1 of Algorithm 2 by:
- $\mathcal{Z}^0 = \langle c^0, \Gamma^0 \rangle$, is built from the box $\boldsymbol{x}^0$,
- d_ε is the vector of DSI obtained by rescaling d^0 between -1 and 1.

Propagation in the layers The propagation in the affine layers can be expressed directly as affine transform on the zonotope by Proposition 2, and later interpreted as a probabilistic zonotope. Proposition 3 introduces the ReLU transformer proposed in [24], encoded in zonotope matrix form. The ReLU transform is applied componentwise (on each line) and a new noise symbol (and thus a

new column in the generator matrix) is added whenever an over-approximation is needed, that is when the input is not either always positive or negative.

Proposition 3 (ReLU transform of a zonotope). *Let $\mathcal{Z} = \langle c, \Gamma \rangle$ with $\Gamma = (g_{ij})_{i,j} \in \mathbb{R}^{n,p}$ be a zonotope, we note $[l_i, u_i] = \gamma_i(\mathcal{Z})$ the range of its i-th component. The result of applying componentwise the ReLU activation function is a zonotope $\mathcal{Z}' = \langle c', \Gamma' \rangle$ where $c' \in \mathbb{R}^n$ and $\Gamma' \in \mathbb{R}^{n,p+n}$, with $c'_i = \lambda_i c_i + \mu_i$ and*

$$\Gamma' = \begin{bmatrix} \lambda_1 g_{11} & \dots & \lambda_1 g_{1p} & \mu_1 & 0 & \dots & 0 \\ \lambda_2 g_{21} & \dots & \lambda_2 g_{2p} & 0 & \mu_2 & \dots & 0 \\ \dots & & & & & & \\ \lambda_n g_{n1} & \dots & \lambda_n g_{np} & 0 & 0 & \dots & \mu_n \end{bmatrix}, (\lambda_i, \mu_i) = \begin{cases} (1,0) & \text{if } l_i \geq 0, \\ (0,0) & \text{if } u_i \leq 0, \\ (\frac{u_i}{u_i - l_i}, -\frac{u_i l_i}{2(u_i - l_i)}) & \text{otherwise.} \end{cases}$$

Output The output zonotope after the L layers is $\mathcal{Z}^L = \langle c^L, \Gamma^L \rangle$ with $c^L \in \mathbb{R}^{h_L}$ and $\Gamma^L \in \mathbb{R}^{h_L, \sum_{k=0}^{L} h_k}$. At line 5 of Algorithm 2, the probabilistic zonotope $p\mathcal{Z}^L(d_\varepsilon)$ is converted into a vector of DSI, following Definition 7. In this interpretation as a probabilistic zonotope, we must define the DSI structures corresponding to the $\sum_{k=1}^{L} h_k$ new noise symbols introduced by the ReLU transformers. A sound although conservative interpretation is to take the interval $[-1, 1]$ as DSI for them. This corresponds to considering that there is no available information about the distribution of the variable represented by these new noise symbols.

At line 6, the transform $H\mathcal{Z}^L$, interpreted as a probabilistic zonotope, is converted in a vector of DSI and used to bound the probability $\mathbb{P}(Hy \leq w)$.

Analysis of the toy example We consider again Example 1.

Deterministic zonotopes analysis From the input sets $x^0 \in [-2, 2] \times [-1, 1]$, the zonotopic interpretation is initialized with the affine forms $x_1^0 = 2\varepsilon_1$, $x_2^0 = \varepsilon_2$ with $\varepsilon_1, \varepsilon_2 \in [-1, 1]$, encoded: $\mathcal{Z}^0 = \langle c^0, \Gamma^0 \rangle$ with $c^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Gamma^0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$. Using the affine transforms on zonotopes and Proposition 3 for the ReLU layer with $(\lambda, \mu) = (0.5, 0.75)$ for both neurons, we obtain after the second affine layer:

$$\mathcal{Z}^2 = A_2 \mathcal{Z}^1 + b_2 = \langle \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0.75 & -0.75 \\ 2 & 0 & 0.75 & 0.75 \end{bmatrix} \rangle \subseteq \begin{bmatrix} [-2.5, 2.5] \\ [-2, 5] \end{bmatrix}$$

The first output x_1^2 is bounded in a tighter interval than with interval propagation ([-3,3]), the second ouput x_2^2 is incomparable to the interval computation ([0,6]).

Probabilistic zonotopes analysis Let us now suppose that the inputs x_1^0 and x_2^0 follow a uniform law, which can be abstracted as in Section 3 with DSI structures d_1^0 and d_2^0. Algorithm 2 produces the same input zonotope and propagation through the network as above. Let us discretize the inputs with 2 focal elements. The rescaling of the DSI d_1^0 and d_2^0 between -1 and 1 yields $d_{\varepsilon_1} = \{\langle [-1, 0], 0.5 \rangle, \langle [0, 1], 0.5 \rangle\}$ and $d_{\varepsilon_2} = \{\langle [-1, 0], 0.5 \rangle, \langle [0, 1], 0.5 \rangle\}$.

The concretization of the final probabilistic zonotope $p\mathcal{Z}^2(d_\varepsilon)$ to a vector of DSI writes: $d_1^2 = -d_{\varepsilon_2} + 0.75d_{\varepsilon_3} - 0.75d_{\varepsilon_4}$ and $d_2^2 = 1.5 + 2d_{\varepsilon_1} + 0.75d_{\varepsilon_3} + 0.75d_{\varepsilon_4}$, where d_{ε_3} and d_{ε_4} are the DSI corresponding to the noise symbols introduced in the analysis by the ReLU function, with unknown distribution in $[-1, 1]$. We get $d_1^2 = \{\langle[-2.5, 1.5], 0.5\rangle, \langle[-1.5, 2.5], 0.5\rangle\}$ and $d_2^2 = \{\langle[-2., 3.], 0.5\rangle, \langle[0., 5.], 0.5\rangle\}$ and deduce $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.5]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 1]$.

The supports of the DSI are equal to the range obtained by the classical zonotopic analysis, thus incomparable to the support of the DSI obtained by Algorithm 1. The results are more generally not strictly comparable to those of DSI computation. For instance here with 100 focal elements, we have $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.26]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 0.76]$ both in the case of independent inputs x_1^0 and x_2^0 and unknown correlation, which is better than DSI in the case of unknown correlation, while DSI are better for independent inputs. The reason why the results do not depend here on the correlation between inputs is that d_{ε_1} does not appear in the expression of d_1^2 and d_{ε_2} in the expression of d_2^2, so that the information of correlation between inputs is not used.

5 Analysis with Zonotopic Dempster-Shafer structures

In Section 4, a unique initial zonotope is built and propagated in the network. This propagation is exact through affine layers, but can be highly conservative for nonlinear operations such as the activation functions. In Algorithm 3, we suppose that the inputs are independent and perform the zonotopic propagation at a finer grain, on each tuple of focal elements of the inputs. This can be seen as using zonotopic focal elements to represent the input vector of a layer, instead of interval focal elements to represent each component of the input vector.

Algorithm 3 Neural network analysis by Dempster-Shafer zonotopic layers

Input: d^0 a h_0-dimensional vector of DSI
1: $d_{\mathcal{Z}}^0 = \left\{\langle \mathcal{Z}_{i_1 \dots i_{h_0}}^0, w_{1,i_1}^0 \dots w_{h_0,i_{h_0}}^0 \rangle,\ (i_1, \dots, i_{h_0}) \in [1, n]^{h_0}\right\} \leftarrow$ 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: $\mathcal{Z}_{i_1 \dots i_{h_0}}^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} A^k \mathcal{Z}_{i_1 \dots i_{h_0}}^k + b^k)$ ▷ Proposition 2 and Proposition 3
5: **end for**
6: **end for**
7: $d_{\mathcal{Z}}^L = \left\{\langle \mathcal{Z}_{i_1 \dots i_{h_0}}^L, w_{1,i_1}^0 \dots w_{h_0,i_{h_0}}^0 \rangle,\ (i_1, \dots, i_{h_0}) \in [1, n]^{h_0}\right\}$
8: $d^L \leftarrow$ dsz-to-dsi($d_{\mathcal{Z}}^L$)
9: **return** (d^L,cdf($(Hd_{\mathcal{Z}}^L, w)$))

Input and initialization The input is the same as in Sections 3 and 4: the uncertain input is modelled as a vector $d^0 = (d_1^0, \dots, d_{h_0}^0)$ of h_0 DSI. Assuming the

input components as independent, we perform the convolution of the distributions of the input components to build a DSZ abstraction of the input vector: we construct one zonotope per possible h_0-tuple of focal elements representing the input vector of DSI d^0, with weight the product of the weights of each interval focal elements: we define for each $(i_1, i_2, \ldots, i_{h_0}) \in [1, n]^{h_0}$ the zonotope $\mathcal{Z}^0_{i_1 \ldots i_{h_0}} = \langle c^0_{i_1 \ldots i_{h_0}}, \Gamma^0_{i_1 \ldots i_{h_0}} \rangle$, built from the box $\boldsymbol{x}^0_{1i_1} \times \boldsymbol{x}^0_{2i_2} \times \ldots \times \boldsymbol{x}^0_{h_0 i_{h_0}}$ and define the input $d^0_{\mathcal{Z}}$ as a Dempster Shafer structure with zonotopic focal elements (DSZ in short): $d^0_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}^0_{i_1 \ldots i_{h_0}}, w^0_{1,i_1} w^0_{2,i_2} \ldots w^0_{h_0,i_{h_0}} \rangle, (i_1, i_2, \ldots, i_{h_0}) \in [1, n]^{h_0} \right\}$.

The number of focal elements does not have to be identical for each component of the input vector d^0, this choice was made here for simplicity of notation. It is also natural to proceed to reductions by heuristically joining some of the focal elements as in the interval case, although we did not implement this yet.

The propagation in the layers then consists in propagating each zonotope focal elements. Note that the number of focal elements remains constant through the propagation in the layers because all convolutions were computed at initialization, only the zonotopes size evolves with the layer dimensions.

Output The DSZ $d^L_{\mathcal{Z}}$ is projected on the output vector, defined for each $i \in [1, h_L]$ by the DSI $d^L_i = \left\{ \langle \gamma_i(\mathcal{Z}^0_{i_1 \ldots i_{h_0}}), w^0_{1,i_1} w^0_{2,i_2} \ldots w^0_{h_0,i_{h_0}}, (i_1, i_2, \ldots, i_{h_0}) \in [1, n]^{h_0} \rangle \right\}$.

The property can be assessed by evaluating the set of joint cumulative distributions represented by the DSZ $Hd^L_{\mathcal{Z}}$, by generalizing the definition of Proposition 1 from interval to zonotopic focal elements:

Proposition 4 (CDF of a Zonotopic Dempster-Shafer structure). *Let X be a random variable in $\mathbb{R}^n$ and $d_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}_1, w_1 \rangle, \langle \mathcal{Z}_2, w_2 \rangle, \ldots, \langle \mathcal{Z}_u, w_u \rangle \right\}$ be a DSZ with $\mathcal{Z}_k = \langle c_k, \Gamma_k \rangle$ with $c_k \in \mathbb{R}^n$ and $\Gamma_k \in R^{pn}$ and $w_k \in]0, 1]$ and $\sum_{k=1}^u w_u = 1$. The DSZ $d_{\mathcal{Z}}$ defines a discrete pbox representing the sets of joint cumulative distribution functions such that for $v \in \mathbb{R}^n$,*

$$\sum_{k \in [1,u],\, \overline{\mathcal{Z}_k} < v} w_k = \underline{P}_v \leq \mathbb{P}(X \leq v) \leq \overline{P}_v = \sum_{k \in [1,u],\, \underline{\mathcal{Z}_k} \leq v} w_k$$

Practically, we can use the ranges or projections of each component of $\mathcal{Z}_k$ to get a conservative over-approximation of the pbox:

$$\overline{P}_v \leq \sum_{k \in [1,u], \bigwedge_{i \in [1,n]} \gamma_i(\mathcal{Z}_k) \leq v_i} w_k \ \wedge \ \underline{P}_v \geq \sum_{k \in [1,u], \bigwedge_{i \in [1,n]} \gamma_i(\mathcal{Z}_k) v_i} w_k$$

This proposition can be derived from the notion of cdf of a random set of [22]. The bounds obtained by Proposition 4 are always at least as good than by first converting the DSZ as a vector and then applying Proposition 1.

DSZ analysis of the toy example We consider again Example 1. with 2 focal elements for each input, we have $d^0_1 = \{\langle [-2, 0], 0.5 \rangle, \langle [0, 2], 0.5 \rangle\}$ and $d^0_2 = \{\langle [-1, 0], 0.5 \rangle, \langle [0, 1], 0.5 \rangle\}$. At Line 1 of Algorithm 3, $d^0_{\mathcal{Z}}$ is a DSZ structure with 4

zonotopic focal elements, each with weight 0.25: $\mathcal{Z}^0_{11} = \langle \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$, $\mathcal{Z}^0_{12} = \langle \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$, $\mathcal{Z}^0_{21} = \langle \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$, $\mathcal{Z}^0_{22} = \langle \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$. After the output layer, the 4 zonotopic elements, each with weight 0.25, are: $\mathcal{Z}^2_{11} = \langle \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle$, $\mathcal{Z}^2_{12} = \langle \begin{bmatrix} -\frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle$, $\mathcal{Z}^2_{21} = \langle \begin{bmatrix} \frac{5}{6} \\ \frac{13}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{5}{6} & -\frac{1}{3} \\ \frac{10}{6} & -\frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle$, $\mathcal{Z}^2_{22} = \langle \begin{bmatrix} -\frac{5}{6} \\ \frac{13}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{5}{6} & \frac{1}{3} \\ \frac{10}{6} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle$. From these and their projected ranges for x^2_1 and x^2_2, we deduce (see AppendixA for details) using Proposition 4 that $\mathbb{P}(x^2_1 \leq -2) \in [0, 0.25]$ and $\mathbb{P}(x^2_2 \geq 2) \in [0, 0.5]$ and the conjunction $\mathbb{P}(Hy \leq w) \in [0.0, 0.25]$.

6 Evaluation

Implementation We implemented our approach[1] using the Julia library ProbabilityBoundsAnalysis.jl[2] [9], for Interval Dempster Shafer abstraction and arithmetic. In this library, the focal elements of a DSI structure all have same weight. The result is reduced after each arithmetic operation to keep a constant number of focal elements. We rely on this DSI implementation, but our DSZ implementation does not present the same restrictions. The focal elements are bounded, but a flag allows the user to specify that a distribution may have unbounded support, and this knowledge is used to produce a sound CDF estimation for unbounded distributions. In our current implementation, we do not use this possibility, but we believe that the work presented here can be extended to unbounded support.

All timings for the results of our analysis are on a MacBook Pro 2,3 GHz Intel Core i9 with 8 cores (the implementation is not parallelized for now, although the technique is obviously easily parallelizable).

Comparing DSI, probabilistic zonotopes and DSZ on the toy example We compare in Table 1 our 3 abstractions in the case of independent inputs, varying the number of focal elements and the input distributions: $U(n)$ denotes a uniform distribution represented with n focal elements, and $N(n)$ a truncated normal law in the same range with n focal elements. On this example, the DSZ analysis is by far more precise, followed by the DSI and finally the probabilistic zonotopes. Refining the input discretization with more focal elements tightens the output of all analyzes, but only the DSZ converges to actually tight bounds. In particular, for DSI and probabilistic zonotopes, the support of the output distribution is unchanged when the input is refined. The computation times are, on this example, of the same order of magnitude for all three analyzes, slightly higher for DSZ, and lower for probabilistic zonotopes. The reason for the probabilistic zonotopes to have lower cost is that affine transforms are computed on the zonotopes, and the costly operations between DSI are delayed until the final

[1] prototype version available at `https://github.com/sputot/DSZAnalysis`

[2] `https://github.com/AnderGray/ProbabilityBoundsAnalysis.jl`

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}	$[0, 0.03]$	$[0.2, 0.3]$	$< e^{-3}$
$U(100)$	$[0, 0.07]$	$[0.05, 0.52]$	0.022	$[0, 0.26]$	$[0, 0.76]$	0.013	$[0, 0.0014]$	$[0.25, 0.26]$	0.026
$U(1000)$	$[0, 0.063]$	$[0.062, 0.502]$	2.4	$[0, 0.251]$	$[0, 0.751]$	1.2	$[0, 3.e^{-6}]$	$[0.25, 0.251]$	3
$N(10)$	$[0, 0.1]$	$[0, 0.4]$	e^{-3}	$[0, 0.1]$	$[0, 1]$	e^{-3}	$[0, 0.01]$	$[0, 0.1]$	$< e^{-3}$
$N(100)$	$[0, 0.01]$	$[0, 0.2]$	0.022	$[0, 0.07]$	$[0, 0.94]$	0.013	$[0, 4.e^{-4}]$	$[0.06, 0.07]$	0.026
$N(1000)$	$[0, 0.004]$	$[3e^{-3}, 0.182]$	2.4	$[0, 0.067]$	$[0, 0.934]$	1.2	$[6e^{-5}, 1.1e^{-4}]$	$[0.066, 0.067]$	3

representation as a DSI. It is not surprising that the DSZ have slightly higher cost, because of the exponential number of zonotopic focal elements. However, we demonstrate in the remaining of this section that it is still able to solve challenging problems and compares favorably to the state of the art. Moreover, the computation is obviously parallelizable.

We can also note the strong impact of the input hypotheses on the results, advocating the need of such an approach which can account in a same framework and computation for large classes of inputs. For instance, changing the input distribution from a uniform to a Gaussian truncated to same support produces very different probability bounds. In Table 1, we supposed the inputs independent. For instance, the DSI analysis for 100 focal elements and a uniform law, produces for independent inputs $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.07]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0.05, 0.52]$, while for inputs with unknown dependence, $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.26]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 1]$.

For independent inputs, the DSZ is the best choice among our approaches. In the case of correlated inputs, it is hard to conclude from such a simple example between DSI and probabilistic zonotopes. In the context of discrete dynamical systems where probabilistic zonotopes were proposed [4, 5], they were much better than DSI both in terms of efficiency and accuracy. The context of neural networks is less favorable, but it is probable that probabilistic zonotopes can be more interesting than DSI for larger networks. However, our focus is to explore in the future the encoding of multivariate probabilistic distributions as input distributions, and lift this current restriction on the DSZ analysis.

Comparing DSZ to Probstar [27] We now compare our approach to the results of the closely related approach [27] on their two benchmark examples. On these examples, the inputs are considered independent in [27], we consider the same hypotheses and use the DSZ analysis.

ACAS Xu We consider the ACAS Xu networks benchmark, where the networks have 5 inputs and 5 outputs, with the same input configurations and properties ($P_2 : y_1 > y_2 \wedge y_1 > y_3 \wedge y_1 > y_4 \wedge y_1 > y_5$, $P_3/P_4 : y_1 < y_2 \wedge y_1 < y_3 \wedge y_1 < y_4 \wedge y_1 < y_5$) as in [27]. The lower and upper bounds on the inputs, lb and ub, depend on the property, and are used in [27] to define probabilistic input sets by Gaussian distributions with mean $m = (ub + lb)/2$ and standard deviation

$(ub - m)/a$, where $a = 3$, truncated between lb and ub. In our work, after creation of the input DSI from the above Gaussian distribution, we truncate all focal elements so that the support of the DSI is restricted to the input range $[lb, ub]$. In [27], an argument is used to deduce bounds for the probability for non truncated distributions. We could use a similar argument here but we focus on the results for the truncated distributions, and compare our results to the interval $[US - Prob - LB, US - Prob - UB]$ with the notations of [27].

We choose for the DSZ approach an initial over-approximation of the input distributions using a different number of focal elements for each component of the vector input, roughly based on the relative widths of the input intervals. We represent these as vectors of number of focal elements, taking [5, 80, 50, 6, 5] for Property 2, [5, 20, 1, 6, 5] for Properties 3 and 4. In Table 2, we compare these with the Probstar approach with two parametrizations: $p_f = 0$ corresponds to an exact set-based propagation, while $p_f = e^{-5}$ corresponds to the level of over-approximation in propagation most widely used in [27]. Let us first comment

Table 2: Probability bounds for the ACAS Xu example.

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

on the running times: the timings for Probstars in Table 2 are those of [27], hence not computed on the same computer as our's. We reproduced Property 2 on Net 1-6 with Probstars on our MacBook: for $p = 0$, it takes 3614s on 8 cores, 5045s on 4 cores, 12542 on 1 core, to be compared to the 1424s in Table 2; for $p = e^{-5}$, it takes 425s on 8 cores, 489 on 4 cores, 1489 on 1 core, to be compared to the 206s in Table 2) and to the 46s with DSZ.

The tightness of the enclosures of the DSZ is comparable to Probstars with $p = e^{-5}$, for an analysis being generally an order of magnitude faster. The results look consistent between the 2 analyzes for Property 2. Properties 3 and 4 (originally from [13]) are true on the whole input range, which can be proven by classical set-based analysis, and our approach accordingly produces a probability equal to 1. The approach of [27] produces more precise, "exact" results, when $p = 0$, than our approach. However, only the set-based propagation is exact,

there is also a part of probabilistic estimation. For instance, when reproducing Property 2 on Net 1-6 with Probstars, we obtained for $p = 0$, the probabilities 1.56119e-05 with 8 cores, 6.76052e-06 with 4 cores, 7.22045e-06 with 1 core, to be compared to the "exact" 1.87224e-05 in the table. In contrast, our approach produces fully guaranteed bounds while allowing a much richer classes of inputs.

In Table 2, we manually chose the number of focal element per input component. Although we refrained from optimizing too much, choosing for instance the same discretization for different networks, this impairs the practicality of the approach. As a first answer, we implemented a simple loop to automatically refine the discretization starting from a very rough one, by some basic sensitivity analysis. For instance, for Property 2 and net-1-6, the total refinement process with as stopping criterion the width of the probability interval lower than 0.05 takes 112 seconds and leads to the number of focal elements $[5, 81, 38, 5, 5]$ and a probability in [0, 0.0276], with bounds twice tighter than Probstars with $p = e^{-5}$.

Rocket lander Let us now consider the rocket lander example of [27], with the same inputs and properties. The networks have here 9 inputs. Taking the vector of focal elements [7, 12, 10, 17, 9, 7, 1, 1, 2, 1, 1] produces the results of Table 3 Again, the timings for Probstars are those of [27]; we executed for instance on

Table 3: Comparing probability bounds for the rocket lander example.

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

our MacBook the analysis of Property 1 on network 0 with 4 cores, the running times were 1351.2s for $p_f = 1e{-}5$ and 12127s for $p_f = 0$.

7 Conclusion

A central notion for dealing with multivariate probabilistic distributions is that of a copula [16], and in particular Sklar's theorem which links multivariate cdf with the cdf of its marginals. Multiple authors have considered generalizing Sklar's theorem to imprecise probabilities, [17, 15], with e.g. applications in [10] to the analysis of non-linear dynamical systems. In this work, we developed the case of multidimensional imprecise probabilites described by the independence copula. Future work includes the tractable treatment of other copulas in our framework, and more generally a better representation of inputs by DSZ structures. Finally, we focused here on ReLU-based networks, but the approach is by no means restricted to this activation function.

References

1. Stanley Bak and Parasara Sridhar Duggirala. Simulation-equivalent reachability of large linear systems with inputs. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification*, pages 401–420, Cham, 2017. Springer International Publishing.
2. S. Ferson Beer, M. and V. Kreinovich. Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing*, 37(1):4–29, 2013.
3. Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *AAAI*, Jan 2019.
4. Olivier Bouissou, Eric Goubault, Jean Goubault-Larrecq, and Sylvie Putot. A generalization of p-boxes to affine arithmetic. *Computing*, 94(2–4):189–201, 2012.
5. Olivier Bouissou, Eric Goubault, Sylvie Putot, Aleksandar Chakarov, and Sriram Sankaranarayanan. Uncertainty propagation using probabilistic affine forms and concentration of measure inequalities. In Marsha Chechik and Jean-François Raskin, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 225–243. Springer, 2016.
6. Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019.
7. Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Probabilistic verification and reachability analysis of neural networks via semidefinite programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2726–2731, 2019.
8. S. Ferson, V. Kreinovich, L.R. Ginzburg, and D.S. Myers. Constructing probability boxes and dempster-shafer structures. Technical report, Sandia National Laboratories, SAND2002-4015, Albuquerque, New Mexico, 2003.
9. Ander Gray, Scott Ferson, and Edoardo Patelli. `ProbabilityBoundsAnalysis.jl`: Arithmetic with sets of distributions. In *Proceedings of JuliaCon*, 2021.
10. Ander Gray, Marcelo Forets, Christian Schilling, Scott Ferson, and Luis Benet. Verified propagation of imprecise probabilities in non-linear ODEs. *International Journal of Approximate Reasoning*, 164:109044, 2024.
11. Patrick Henriksen and Alessio R. Lomuscio. Efficient neural network verification via adaptive refinement and adversarial search. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2513–2520. IOS Press, 2020.
12. Chengqiang Huang, Zheng Hu, Xiaowei Huang, and Ke Pei. Statistical certification of acceptable robustness for neural networks. In Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 79–90, Cham, 2021. Springer International Publishing.

13. Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification*, pages 97–117, Cham, 2017. Springer International Publishing.
14. Zhaoyang Lyu, Ching-Yun Ko, Zhifeng Kong, Ngai Wong, Dahua Lin, and Luca Daniel. Fastened crown: Tightened neural network robustness certificates. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5037–5044, 2020.
15. Ignacio Montes, Enrique Miranda, Renato Pelessoni, and Paolo Vicig. Sklar's theorem in an imprecise setting. *Fuzzy Sets and Systems*, 278:48–66, 2015. Special Issue on uncertainty and imprecision modelling in decision making (EUROFUSE 2013).
16. Roger B Nelsen. *An Introduction to Copulas*. Springer, New York, NY, USA, second edition, 2006.
17. Matjaž Omladič and Nik Stopar. A full scale sklar's theorem in the imprecise setting. *Fuzzy Sets and Systems*, 393:113–125, 2020. Copulas and Related Topics.
18. Mikhail Pautov, Nurislam Tursynbek, Marina Munkhoeva, Nikita Muravev, Aleksandr Petiushko, and Ivan Oseledets. Cc-cert: A probabilistic approach to certify general robustness of neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:7975–7983, 06 2022.
19. Joshua Pilipovsky, Vignesh Sivaramakrishnan, Meeko Oishi, and Panagiotis Tsiotras. Probabilistic verification of relu neural networks via characteristic functions. In Nikolai Matni, Manfred Morari, and George J. Pappas, editors, *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 966–979. PMLR, 15–16 Jun 2023.
20. Corina Păsăreanu, Hayes Converse, Antonio Filieri, and Divya Gopinath. On the probabilistic analysis of neural networks. In *2020 IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 5–8, 2020.
21. Helen Regan, Scott Ferson, and Daniel Berleant. Equivalence of methods for uncertainty propagation of real-valued random variables. *International Journal of Approximate Reasoning*, 36:1–30, 04 2004.
22. Bernhard Schmelzer. Random sets, copulas and related sets of probability measures. *International Journal of Approximate Reasoning*, 160:108952, 2023.
23. Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
24. Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
25. Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin T. Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 10825–10836, 2018.
26. Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, (POPL), 2019.
27. Hoang-Dung Tran, Sungwoo Choi, Hideki Okamoto, Bardh Hoxha, Georgios Fainekos, and Danil Prokhorov. Quantitative verification for neural networks using probstars. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '23, New York, NY, USA, 2023. Association for Computing Machinery.

28. Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall, 1991.
29. Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. A statistical approach to assessing neural network robustness. *ICLR and arXiv:1811.07209*, 2019.
30. Lily Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. PROVEN: Verifying robustness of neural networks with a probabilistic approach. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6727–6736. PMLR, 09–15 Jun 2019.
31. Robert C. Williamson and Tom Downs. Probabilistic arithmetic: Numerical methods for calculating convolutions and dependency bounds. *Journ. Approx. Reas.*, 1990.
32. Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: a functional optimization based framework. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.
33. Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 4939–4948. Curran Associates, Inc., 2018.
34. Tianle Zhang, Wenjie Ruan, and Jonathan E. Fieldsend. Proa: A probabilistic robustness assessment against functional perturbations. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III*, page 154–170, Berlin, Heidelberg, 2023. Springer-Verlag.

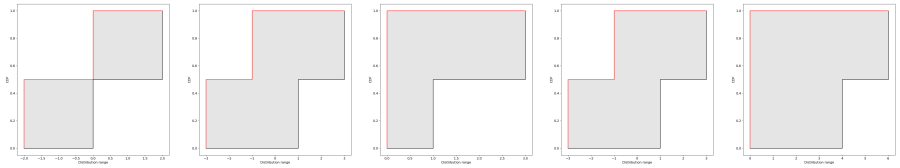
A Analyzes of Example 1 (toy example)

Deterministic (interval) inputs and analysis An interval propagation in the network from the input set $x^0 = (x_1^0, x_2^0) \in [-2, 2] \times [-1, 1]$ yields for the first affine layer followed by the ReLU activation: $x_1^1 = \sigma([-2, 2] - [-1, 1]) = \sigma([-3, 3]) = [0, 3]$ and $x_2^1 = \sigma([-2, 2] + [-1, 1]) = [0, 3]$. After the 2nd layer, $x_1^2 = [0, 3] - [0, 3] = [-3, 3]$ and $x_2^2 = [0, 3] + [0, 3] = [0, 6]$. As the output ranges for x_1^2 and x_2^2 have non empty intersection with the property $x_1^2 \leq -2 \wedge x_2^2 \geq 2$, this analysis does not allow to conclude.

Uniform distribution on inputs Let us now suppose that we additionally know that the 2 components of the input follow a uniform distribution over the previous range.

DSI with 2 focal elements Let us first choose for demonstration purpose a discretization by a DSI with 2 focal elements. The DSI for x_1^0 is $d_1^0 = \{\langle[-2, 0], 0.5\rangle; \langle[0, 2], 0.5\rangle\}$, represented Figure 1a. This produces a rough staircase over-approximation of the CDF of the uniform distribution, which would be a diagonal line here. Similarly, a DSI discretizing x_2^0 following a uniform distribution betwwen -1 and 1 with 2 focal elements is $d_2^0 = \{\langle[-1, 0], 0.5\rangle; \langle[0, 1], 0.5\rangle\}$. Let us suppose the inputs are known to be independent, the first output after the first affine layer, $d_{y_1} = d_1^0 - d_2^0$, computed following Definition 4, is $\{\langle[-2, 1], 0.25\rangle; \langle[-3, 0], 0.25\rangle; \langle[0, 3], 0.25\rangle; \langle[-1, 2], 0.25\rangle\}$. In order to limit the complexity of computation, the result of each operation on DSI can be reduced by a sound overapproximation with a fixed number of focal elements. This can be done by joining some focal elements and adding the corresponding weights. For instance here, when reducing to 2 focal elements by joining the first 2 and the last 2 focal elements, this results in $d_{y_1} = \{\langle[-3, 1], 0.5\rangle, \langle[-1, 3], 0.5\rangle\}$, represented Figure 1b. Then, applying to d_{y_1} the ReLU function using Definition 5 produces $d_1^1 = \{\langle[0, 1], 0.5\rangle; \langle[0, 3], 0.5\rangle\}$ represented Figure 1b. The other output x_2^1 of the first layer has the same DSI representation. After the output layer, the first output is $d_1^2 = d_1^1 - d_2^1 = \{\langle[-3, 1], 0.5\rangle; \langle[-1, 3], 0.5\rangle\}$, represented Figure 1d. Here x_1^1 and x_2^1 can no longer be considered as independent as they both are correlated to x_1^0 and x_2^0 and the subtraction of their DSI representation is computed accordingly. The second output $d_2^2 = d_1^1 + d_2^1 = \{\langle[0, 4], 0.5\rangle; \langle[0, 6], 0.5\rangle\}$ is represented Figure 1e.

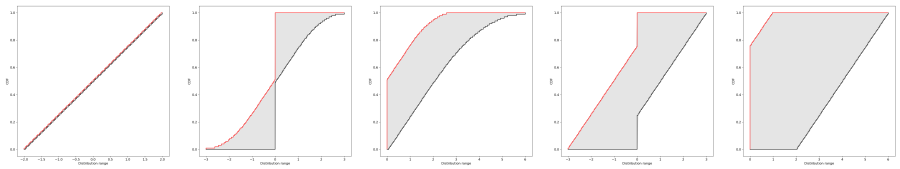
Let us now consider the property $x_1^2 \leq -2 \wedge x_2^2 \geq 2$. Using Proposition 1, we can deduce from the DSI d_1^2 and d_2^2 that $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.5]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0.0, 1.0]$, from which $\mathbb{P}(x_1^2 \leq -2 \wedge x_2^2 \geq 2) \in [0, 0.5]$. Consider for instance $\mathbb{P}(x_1^2 \leq -2)$ evaluated using $d_1^2 = \{\langle[-3, 1], 0.5\rangle; \langle[-1, 3], 0.5\rangle\}$. We get the lower bound using Proposition 1 by $\underline{P}(-2) = \sum_{\overline{\boldsymbol{x_i}} < -2} w_i = 0$, as the upper bounds of the 2 focal elements $[-3, 1]$ and $[-1, 3]$ are both greater than -2. We get the upper bound by $\overline{P}(-2) = \sum_{\underline{\boldsymbol{x_i}} \leq u} w_i = 0.5$, as the lower bound of $[-3, 1]$ is lower than -2, which is not the case for $[-1, 3]$.



(a) d_1^0 (b) d_{y_1} (c) $d_1^1 = \sigma(d_{y_1})$ (d) d_1^2 (e) d_2^2

Fig. 1: Toy example, Uniform law on inputs (DSI with 2 focal elements)

DSI with 100 focal elements The DSI computation with 100 focal elements produces the results of Figure 2. Figure 2a represents the staircase over-approximation of the CDF of the uniform law. Without surprise, with this more accurate representation of the inputs, the DSI outputs d_1^2 and d_2^2 of the network, represented Figure 2b and Figure 2c, correspond to a smaller set of CDF than the same outputs computed with 2 focal elements of Figure 1d and Figure 1e and thus refine these results. We also represent in Figure 2d and Figure 2e, the outputs of the same network when the inputs are no longer supposed independent, but with unknown correlation. As can be expected, the sets of CDF for the outputs are larger than when making the assumption of independence.



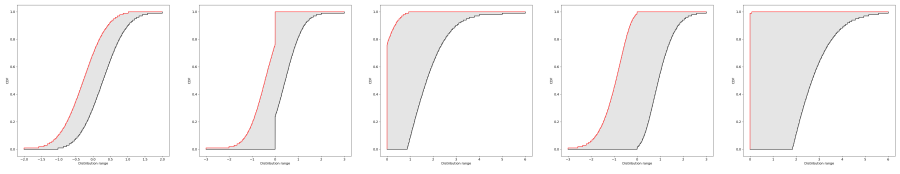
(a) d_1^0 (b) d_1^2 (c) d_2^2 (d) d_1^2 unknown correlation (e) d_2^2 unknown correlation

Fig. 2: Toy example, Uniform law on inputs (DSI with 100 focal elements)

Increasing the number of focal elements refines the sets of CDF obtained for the outputs. However, it should be noted that the supports of the sets of distribution are unchanged, and equal to the ranges obtained by classical interval analysis ([-3,3] for x_1^2 and [0,6] for x_2^2). Indeed, some conservatism is introduced in the interpretation of affine layers due to the wrapping effect when computing on the interval focal elements.

Let us consider the property $x_1^2 \leq -2 \wedge x_2^2 \geq 2$. In the case inputs can be considered as independent, the DSI for x_1^2 and x_2^2 allow us to conclude that $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.07]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0.05, 0.52]$. In the case of inputs with unknown correlation, $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.26]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 1]$. These bounds directly result from Proposition 1.

Truncated Gaussian distribution on inputs We now consider that the inputs follow a Gaussian law with uncertain mean, truncated to the same support as before, and discretized with 100 focal elements. The input DSI d_1^0 and the outputs DSIs are represented Figure 3. The support of the output distributions are the same as obtained for inputs with uniform law, but the output distributions are quite different. In this case, for independent inputs, $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.02]$



(a) d_1^0 (b) d_1^2 (c) d_2^2 (d) d_1^2 unknown correlation (e) d_2^2 unknown correlation

Fig. 3: Toy example, uncertain Gaussian law (DSI with 100 focal elements)

and $\mathbb{P}(x_2^2 \geq 2) \in [0, 0.37]$. In the case of inputs with unknown dependence, $\mathbb{P}(x_1^2 \leq -2) \in [0, 0.06]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0, 0.89]$.

Deterministic (zonotopic) analysis From the input sets $x^0 \in [-2, 2] \times [-1, 1]$, the zonotopic interpretation is initialized with the affine forms $x_1^0 = 2\varepsilon_1$, $x_2^0 = \varepsilon_2$ with $\varepsilon_1, \varepsilon_2 \in [-1, 1]$, encoded:

$$\mathcal{Z}^0 = \langle c^0, \Gamma^0 \rangle \text{ with } c^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Gamma^0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

The first affine layer yields

$$A_1 \mathcal{Z}^0 + b_1 = \langle \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} \rangle \subseteq \begin{bmatrix} [-3, 3] \\ [-3, 3] \end{bmatrix}$$

Using Proposition 3 for the ReLU layer with $(\lambda, \mu) = (0.5, 0.75)$ for both neurons produces:

$$\mathcal{Z}^1 = \sigma(A_1 \mathcal{Z}^0 + b_1) = \langle \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}, \begin{bmatrix} 1 & -0.5 & 0.75 & 0 \\ 1 & 0.5 & 0 & 0.75 \end{bmatrix} \rangle$$

Finally, after the second affine layer:

$$\mathcal{Z}^2 = A_2 \mathcal{Z}^1 + b_2 = \langle \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0.75 & -0.75 \\ 2 & 0 & 0.75 & 0.75 \end{bmatrix} \rangle \subseteq \begin{bmatrix} [-2.5, 2.5] \\ [-2, 5] \end{bmatrix}$$

The interval ranges for the outputs of the first layer are larger than the ones obtained with direct interval computation in Section 3. Indeed, the interpretation of the ReLU activation by a zonotope is conservative. However, the affine

forms express correlations, so that after the second layer the first component x_1^2 ranges in a tighter interval than obtained with the direct interval propagation ([-3,3]), while the second component x_2^2 is incomparable to the direct interval computation ([0,6]).

Analysis with probabilistic zonotopes Let us now suppose that the inputs x_1^0 and x_2^0 follow a uniform law over their range, which can be abstracted as in Section 3 with DSI structures d_1^0 and d_2^0. Algorithm 2 produces the same input zonotope and propagation through the network as above. Let us discretize the inputs with 2 focal elements. The rescaling of the DSI d_1^0 and d_2^0 between -1 and 1 yields $d_{\varepsilon_1} = \{\langle[-1,0],0.5\rangle;\langle[0,1],0.5\rangle\}$ and $d_{\varepsilon_2} = \{\langle[-1,0],0.5\rangle;\langle[0,1],0.5\rangle\}$.

The concretization of the final probabilistic zonotope $p\mathcal{Z}^2(d_\varepsilon)$ to a vector of DSI writes: $d_1^2 = -d_{\varepsilon_2} + 0.75d_{\varepsilon_3} - 0.75d_{\varepsilon_4}$ and $d_2^2 = 1.5 + 2d_{\varepsilon_1} + 0.75d_{\varepsilon_3} + 0.75d_{\varepsilon_4}$, where d_{ε_3} and d_{ε_4} are the DSI corresponding to the noise symbols introduced in the analysis by the ReLU function, with unknown distribution in $[-1,1]$.

The supports of the DSI are equal to the range obtained by the classical zonotopic analysis, thus incomparable to the support of the DSI obtained by Algorithm 1. With 2 focal elements, from the concretization of $p\mathcal{Z}^2(d_\varepsilon)$ we obtain $d_1^2 = \{\langle[-2.5,1.5],0.5\rangle;\langle[-1.5,2.5],0.5\rangle\}$ and $d_2^2 = \{\langle[-2.,3.],0.5\rangle;\langle[0.,5.],0.5\rangle\}$. We deduce $\mathbb{P}(x_1^2 \leq -2) \in [0,0.5]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0,1]$.

The results are not strictly comparable to the case of direct DSI computation. For instance here with 100 focal elements, the probabilities of property violation are $\mathbb{P}(x_1^2 \leq -2) \in [0,0.26]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0,0.76]$ both in the case of independent inputs x_1^0 and x_2^0 and unknown correlation. These results are better than for direct DSI computation with 100 focal elements in the case of unknown correlation, but the direct DSI are better in the case of independent inputs. The reason why the results do not depend on the correlation between inputs in the case of probabilistic zonotopes is that this is a very particular case where one of the 2 inputs (d_{ε_1} or d_{ε_2}) cancels out in both expressions of the output DSI $d_1^2 = -d_{\varepsilon_2} + 0.75d_{\varepsilon_3} - 0.75d_{\varepsilon_4}$ and $d_2^2 = 1.5 + 2d_{\varepsilon_1} + 0.75d_{\varepsilon_3} + 0.75d_{\varepsilon_4}$, so that the information of correlation between inputs is not used.

Note finally that refining the input discretization by using more focal elements tightens the output DSI and probability bounds, but not considerably so. For instance, with 10 focal elements, we obtain $\mathbb{P}(x_1^2 \leq -2) \in [0,0.3]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0,0.8]$, while for 500 elements, we obtain $\mathbb{P}(x_1^2 \leq -2) \in [0,0.252]$ and $\mathbb{P}(x_2^2 \geq 2) \in [0,0.752]$.

DSZ analysis for 2 focal elements Let us take 2 focal elements for each of the 2 inputs, we initially have $d_1^0 = \{\langle[-2,0],0.5\rangle;\langle[0,2],0.5\rangle\}$ and $d_2^0 = \{\langle[-1,0],0.5\rangle;\langle[0,1],0.5\rangle\}$. At Line 1 of Algorithm 3, $d_{\mathcal{Z}}^0$ is a DSZ structure with 4 zonotopic focal elements, each with weight 0.25:

$$\mathcal{Z}_{11}^0 = \langle \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle,\ \mathcal{Z}_{12}^0 = \langle \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle,$$

$$\mathcal{Z}^0_{21} = \langle \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle, \mathcal{Z}^0_{22} = \langle \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$$

The first affine layer transforms each of these 4 zonotopes relying on Proposition 2 and produces:

$$\mathcal{Z}'^0_{11} = \langle \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle, \mathcal{Z}'^0_{12} = \langle \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle,$$

$$\mathcal{Z}'^0_{21} = \langle \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle, \mathcal{Z}'^0_{22} = \langle \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \rangle$$

Applying the ReLU activation on each zonotope using Proposition 3 produces:

$$\mathcal{Z}^1_{11} = \langle \begin{bmatrix} \frac{1}{6} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix} \rangle, \mathcal{Z}^1_{12} = \langle \begin{bmatrix} 0 \\ \frac{1}{6} \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle,$$

$$\mathcal{Z}^1_{21} = \langle \begin{bmatrix} 1.5 \\ \frac{2}{3} \end{bmatrix}, \begin{bmatrix} 1 & -0.5 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \rangle, \mathcal{Z}^1_{22} = \langle \begin{bmatrix} \frac{2}{3} \\ 1.5 \end{bmatrix}, \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\ 1 & 0.5 & 0 \end{bmatrix} \rangle$$

Finally, after the output affine layer (without ReLU):

$$\mathcal{Z}^2_{11} = \langle \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle, \mathcal{Z}^2_{12} = \langle \begin{bmatrix} -\frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle,$$

$$\mathcal{Z}^2_{21} = \langle \begin{bmatrix} \frac{5}{6} \\ \frac{13}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{5}{6} & -\frac{1}{3} \\ \frac{10}{6} & -\frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle, \mathcal{Z}^2_{22} = \langle \begin{bmatrix} -\frac{5}{6} \\ \frac{13}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{5}{6} & \frac{1}{3} \\ \frac{10}{6} & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \rangle$$

We can now analyze the probability of the network output y satisfying the linear safety property $Hy \leq w$. In order to evaluate on the output $d^2_{\mathcal{Z}} = \{\langle \mathcal{Z}^2_{i_1 i_2}, (i_1, i_2) \in [1,2]^2, 0.25 \rangle\}$, we first compute the affine transform $Hd^2_{\mathcal{Z}}$ and then use the resulting DSZ to bound the probability $\mathbb{P}(y = Hx^2 \leq w)$. The affine transform $Hd^2_{\mathcal{Z}}$ produces the DSZ with the four following zonotopes with equal weight of 0.25:

$$H\mathcal{Z}^2_{11} = \langle \begin{bmatrix} \frac{1}{6} \\ -\frac{1}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{6} & -\frac{1}{3} \end{bmatrix} \rangle, H\mathcal{Z}^2_{12} = \langle \begin{bmatrix} -\frac{1}{6} \\ -\frac{1}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{3} \end{bmatrix} \rangle,$$

$$H\mathcal{Z}^2_{21} = \langle \begin{bmatrix} \frac{5}{6} \\ -\frac{13}{6} \end{bmatrix}, \begin{bmatrix} \frac{1}{3} & -\frac{5}{6} & -\frac{1}{3} \\ -\frac{10}{6} & \frac{1}{6} & -\frac{1}{3} \end{bmatrix} \rangle, H\mathcal{Z}^2_{22} = \langle \begin{bmatrix} -\frac{5}{6} \\ -\frac{13}{6} \end{bmatrix}, \begin{bmatrix} -\frac{1}{3} & -\frac{5}{6} & \frac{1}{3} \\ -\frac{10}{6} & -\frac{1}{6} & -\frac{1}{3} \end{bmatrix} \rangle$$

The projected ranges of the 4 zonotopes, which have probability 0.25, are:

$$\gamma(y_1) \in [-0.67, 1.0] \wedge \gamma(y_2) \in [-1.0, 0.67];$$
$$\gamma(y_1) \in [-1.0, 0.67] \wedge \gamma(y_2) \in [-1.0, 0.67];$$
$$\gamma(y_1) \in [-0.67, 2.34] \wedge \gamma(y_2) \in [-4.34, 0.0];$$
$$\gamma(y_1) \in [-2.34, 0.67] \wedge \gamma(y_2) \in [-4.34, 0.0]]$$

From these, we deduce using Proposition 4 that $\mathbb{P}(x^2_1 \leq -2) = \mathbb{P}(y_1 \leq -2) \in [0, 0.25]$ and $\mathbb{P}(x^2_2 \geq 2) = \mathbb{P}(y_2 \leq -2) \in [0, 0.5]$. When considering the conjunction, $\mathbb{P}(x^2_1 \leq -2 \wedge x^2_2 \geq 2) \in [0.0, 0.25]$.