SCALING HYBRID CONSTRAINED ZONOTOPES WITH OPTIMISATION

THOMAS WINNINGER, CATERINA URBAN, AND GUANNAN WEI

Abstract. Zonotopes are promising abstract domains for machine learning oriented tasks due to their computational efficiency, but they lack the expressiveness required to precisely handle complex transformations, such as the softmax function, prevalant in the transformer architecture. Hybrid Constrained Zonotopes (HCZ) address this limitation by incorporating linear constraints and binary generators, enabling the representation of non-convex sets. However, existing HCZ rely on Mixed-Integer Linear Programming (MILP) solvers for concretisation, resulting in exponential time complexity that renders them impractical for large-scale models. This work leverages the Lagrangian duality to develop polynomial-time convex relaxations for HCZ operations. Nonetheless, as the HCZ's operations are not designed to handle convex relaxations, the overapproximation is potentially exessive. Thus, future work is needed to make HCZ usable for Large Language Model's verification.

1. Introduction

A Zonotope represents a set as the Minkowski sum of line segments, enabling instant concretisation. As they lack precision, recent research proposed various new enhanced Zonotope abstract domains. Constrained Zonotopes add linear constraints to improve precision. Polynomial Zonotopes extend the representation by incorporating polynomial generators. Finally, Hybrid Constrained Zonotopes (HCZ) combine binary generators and linear equality constraints, enabling the use of non-convex sets, unions and intersections, which makes operations like the ReLU exact.

Nonetheless, despite the advantage of constrained Zonotopes, their practical use is limited by their computational complexity as the concretisation requires solving the linear equalities, which is NP-hard. Working with Large Language Models (LLM), they become unusable.

This paper addresses the scalability issue of HCZ through the following contributions: a polynomial-time convex relaxation for the concretisation, a polynomial-time dot product, lighter reductions methods that do not require solving MILP, and memory-efficient operations using sparse tensors.

It does not, however, addresses the new problem of over-approximation induced by the relaxations. Future work will have to adapt the operations to this concretisation method.

We will start by quickly presenting classical Zonotopes, and the previous state of HCZ. We will then describe the new methods, along with proofs. A later paragraph will be dedicated on the over-approximation issue, and potential direction for future work.

2. Classical Zonotope

A classical Zonotope [1] in $\mathbb{R}^N$ abstracts a set of N variables through affine expressions over shared noise symbols:

$$z = c + G\mathcal{E} \tag{1}$$

Where $c \in \mathbb{R}^N$ is the center, or the bias, $G \in \mathbb{R}^{N\times I}$ is the generator matrix, and $\mathcal{E} \in [-1,1]^I$ represents I noise symbols.

The concretisation $\gamma(z) = \left\{c + G\mathcal{E} \mid \mathcal{E} \in [-1,1]^I\right\}$ can be efficiently computed as interval bounds, the lower bound $l = c - \|G\|_1$, and the upper bound $u = c + \|G\|_1$. This $O(N \times I)$ complexity makes Zonotope highly scalable, but greatly limits their expressiveness.

3. Hybrid Constrained Zonotope

Hybrid Constrained Zonotopes extend classical zonotopes by incorporating linear constraints and binary generators, they are defined as:

$$z = c + G\mathcal{E} + G'\mathcal{E}', \qquad \text{s.t.} \begin{cases} A\mathcal{E} + A'\mathcal{E}' = b \\ \mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'} \end{cases} \tag{2}$$

Where $c \in \mathbb{R}^N$ is the center, $G \in \mathbb{R}^N \times \mathbb{R}^I$ and $G' \in \mathbb{R}^N \times \mathbb{R}^{I'}$ are continuous and binary generator matrices, $\mathcal{E} \in [-1,1]^I$ and $\mathcal{E}' \in \{-1,1\}^J$ represent the I continuous and J binary noise symbols. $A \in \mathbb{R}^J \times \mathbb{R}^I$, $A' \in \mathbb{R}^J \times \mathbb{R}^{I'}$, and $b \in \mathbb{R}^J$ define the I' linear constraints.

This definition can be extended to multi-dimensional variables with $c \in \mathbb{R}^{\cdots}, G \in \mathbb{R}^{\cdots\times I}$, and $G' \in \mathbb{R}^{\cdots\times I'}$.

3.1. **Dual-based concretisation.**

The primary computational bottleneck in HCZ is concretisation. The exact lower bound requires solving:

$$l = \min_{\substack{A\mathcal{E}+A'\mathcal{E}'=b \\ \mathcal{E}\in[-1,1]^I \\ \mathcal{E}'\in\{-1,1\}^{I'}}} c + G\mathcal{E} + G'\mathcal{E}' \tag{3}$$

This is a MILP due to the binary constraints, leading to exponential complexity. We address this with the dual Lagrangian optimisation problem.

Proposition 1: ***(Dual concretisation bounds)*** *If the HCZ is not empty – ie, there exists $\mathcal{E} \in [-1,1]^I, \mathcal{E}' \in \{-1,1\}^{I'}$ such that $A\mathcal{E} + A'\mathcal{E}' = b$ – sound lower and upper bounds can be computed as:*

$$l \geq \max_{\Lambda\in\mathbb{R}^N\times\mathbb{R}^J} c + \Lambda b - \|G - \Lambda A\|_1 - \|G' - \Lambda A'\|_1 \tag{4.1}$$

$$u \leq -\max_{\Lambda\in\mathbb{R}^N\times\mathbb{R}^J} -c + \Lambda b - \|G + \Lambda A\|_1 - \|G' + \Lambda A'\|_1 \tag{4.2}$$

Proof. The previous minimisation problem can be rewritten:

$$l = \min_{\substack{\forall j\in[\![1,J]\!] \\ A_j\mathcal{E}+A'_j\mathcal{E}'-b_j=0}} \min_{\substack{\mathcal{E}\in[-1,1]^I \\ \mathcal{E}'\in\{-1,1\}^{I'}}} c + G\mathcal{E} + G'\mathcal{E}' \tag{5.1}$$

$$= \min_{\substack{\forall j \in \llbracket 1, J \rrbracket \\ h_j(\mathcal{E}, \mathcal{E}')=0}} f(\mathcal{E}, \mathcal{E}') \tag{5.2}$$

Transforming the objective into its dual, it becomes:

$$L(\mathcal{E}, \mathcal{E}', \lambda) = \max_{\lambda \in \mathbb{R}^N} f(\mathcal{E}, \mathcal{E}') - \sum_{j}^{J} \lambda_j h_j(\mathcal{E}, \mathcal{E}') \tag{6.1}$$

$$= \max_{\lambda \in \mathbb{R}^N} \min_{\substack{\mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'}}} c + G\mathcal{E} + G'\mathcal{E}' - \sum_{j}^{J} \lambda_j \big(A_j \mathcal{E} + A'_j \mathcal{E}' - b_j \big) \tag{6.2}$$

Using matrices instead:

$$L(\mathcal{E}, \mathcal{E}', \Lambda) = \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} \min_{\substack{\mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'}}} c + \Lambda b + (G - \Lambda A)\mathcal{E} + (G' - \Lambda A')\mathcal{E}' \tag{7}$$

Since $[-1, 1]^I \times \{-1, 1\}^{I'}$ is compact, we can reverse the min-max order:

$$L(\mathcal{E}, \mathcal{E}', \Lambda) = \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} \min_{\substack{\mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'}}} c + \Lambda b + (G - \Lambda A)\mathcal{E} + (G' - \Lambda A')\mathcal{E}' \tag{8.1}$$

$$= \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} c + \Lambda b - \|G - \Lambda A\|_1 - \|G' - \Lambda A'\|_1 \tag{8.2}$$

$$= \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} d(\Lambda) \tag{8.3}$$

By weak duality, this provides a sound lower bound: $l \geq L(\mathcal{E}, \mathcal{E}', \Lambda)$ for every $\Lambda \in \mathbb{R}^N \times \mathbb{R}^J, \mathcal{E} \in [-1, 1], \mathcal{E}' \in \{-1, 1\}$.

Furthermore, $(G, G') \in [-1, 1]^I \times \{-1, 1\}^{I'}$ ensures the set defined by the HCZ is finite and admits a maximum. Thus the maximum can be computed as the minimum of $-f$, and the upper bound u can be computed as follows:

$$-u \geq \min_{\substack{\forall j \in \llbracket 1, J \rrbracket \\ A_j \mathcal{E} + A'_j \mathcal{E}' - b_j = 0}} \min_{\substack{\mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'}}} -c - G\mathcal{E} - G'\mathcal{E}' \tag{9}$$

The same procedure leads to:

$$-u \geq \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} -c + \Lambda b - \|-G - \Lambda A\|_1 - \|-G' - \Lambda A'\|_1 \tag{10}$$

□

Proposition 2: ***(Concretisation complexity)*** *The concretisation has a time complexity linear in the number of variables and precision:* $O(N \times J \times \max(I, I') \times \text{n\_steps})$.

Proof. The forward pass has a complexity of $O(N \times J) + O(N \times J \times I) + O(N \times J \times I')$ for the multiplications, and $O(N \times I) + O(N \times I')$ for the norm, thus an overall complexity of $O(N \times J \times \max(I, I'))$. The backward pass has the same complexity with automatic differentiation, which is the case in frameworks like PyTorch.

If the optimiser used is Adam, it costs $O(N \times J)$, with n_steps iterations, which gives the total complexity of $O(N \times J \times \max(I, I') \times \text{n\_steps})$. □

Proposition 3: ***(Emptiness)*** *There exists a finite number of optimisation steps N, such that for z concretized with* n_steps $> N$*:*

$$z \neq \emptyset \Leftrightarrow l \leq u \tag{11}$$

Proof. Using the Equation 4 to compute l gives for each optimisation step k:

$$l_k \geq \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} \min_{\substack{\mathcal{E} \in [-1,1]^I \\ \mathcal{E}' \in \{-1,1\}^{I'}}} c + G\mathcal{E} + G'\mathcal{E}' - \Lambda(A\mathcal{E} + A'\mathcal{E}' - b) \tag{12.1}$$

$$\geq \max_{\Lambda \in \mathbb{R}^N \times \mathbb{R}^J} \alpha - \Lambda\beta \tag{12.2}$$

With $\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^J$. If the HCZ is empty, there is no $\mathcal{E} \in [-1,1]^I, \mathcal{E}' \in \{-1,1\}^{I'}$ such that $A\mathcal{E} + A'\mathcal{E}' - b$, thus $|\beta| > 0$, and $l_k \overset{\infty}{\to} l = +\infty$. Similarly, $u_k \overset{\infty}{\to} u = -\infty$. As the optimisation objective is concave – $d(\Lambda)$ is concave – there exists M such that $\forall n \geq N, l_n > u_n$. □

4. Abstract Transformers

Set operations – Minkowski sum, cartesian product, intersection, union – are already defined by [2], and the general abstract transformer construction for the classical Zonotope on convex functions is already defined by [3]. The following sections will only extend the tranformer construction to HCZ, and propose the new operation needed for LLMs, the dot product.

4.1. General abstract transformer construction for classical Zonotopes.

Proposition 4: ***(General abstract transformer construction for classical Zonotopes)*** *Given an input Zonotope x, with bounds $[l,u]$, the sound abstract transformer of a convex C^1 continuous function $f : \mathbb{R} \to \mathbb{R}$ is defined as:*

$$y = \lambda x + \mu + \beta\varepsilon_{\text{new}} \tag{13}$$

With:

$$\lambda = f'(t) \tag{14.1}$$

$$\mu = \frac{1}{2}\left(f(t) - \lambda t + \begin{cases} f(l) - \lambda l, & \text{if } t \geq t_{\text{crit}} \\ f(u) - \lambda u, & \text{if } t < t_{\text{crit}} \end{cases}\right) \tag{14.2}$$

$$\beta = \frac{1}{2}\left(\lambda t - f(t) + \begin{cases} f(l) - \lambda l, & \text{if } t \geq t_{\text{crit}} \\ f(u) - \lambda u, & \text{if } t < t_{\text{crit}} \end{cases}\right) \tag{14.3}$$

$$\nabla_x f(x)|_{x=t_{\text{crit}}} = \frac{f(u) - f(l)}{u - l} \tag{14.4}$$

The minimal area abstract transformer is computed using $t = t_{\text{crit}}$.

Proof. [3] □

It is possible to take into account additional constraints on the output, by applying them to Equation 13, which will yield t_{crit_2}.

Proposition 5: ***(Positive exponential abstract transformer)*** *The positive exponential can be computed using $t_{\text{opt}} = \min\left(t_{\text{crit}}, t_{\text{crit}_2}\right)$, with $t_{\text{crit}_2} = l + 1$.*

Proof. For the output of the exponential to be positive, y has to verify $0 \leq y$. The critical point t_{crit_2} can then be chosen by solving $\min y = 0$. As the exponential is monotonically increasing, the minimum of y will be found at the lower bound, and the t_{crit_2} will have to be chosen inferior to t_{crit}. This leads to $\min y = \lambda l + \mu - |\beta|$. As the exponential is convex, $f'(t) \leq \frac{f(t)-f(u)}{t-u} \Leftrightarrow f(t) - \lambda t \leq f(u) - \lambda u \Leftrightarrow \lambda t - f(t) + f(u) - \lambda u \geq 0 \Leftrightarrow \beta \geq 0$, which leads to $\min y = \lambda(l-t) + f(t)$. With values, $0 = e^{t_{\text{crit}_2}}\left(l - t_{\text{crit}_2} + 1\right) \Rightarrow t_{\text{crit}_2} = l + 1$. □

Proposition 6: ***(Positive reciprocal abstract transformer)*** *The positive reciprocal can be computed using $t_{\text{opt}} = \max\left(t_{\text{crit}}, t_{\text{crit}_2}\right)$, with $t_{\text{crit}_2} = \frac{u}{2}$.*

Proof. The same method applied to the reciprocal function, which is monotonically decreasing leads to $\min y = \lambda u + \mu - |\beta| \Rightarrow 0 = \lambda(u-t) + f(t)$, and $0 = -\frac{1}{t^2_{\text{crit}_2}}\left(u - t_{\text{crit}_2}\right) + \frac{1}{t_{\text{crit}_2}} \Rightarrow t_{\text{crit}_2} = \frac{u}{2}$. □

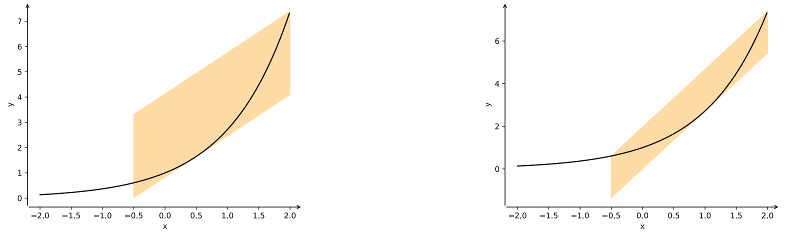

FIGURE 1. Exponential abstract transformer for Zonotope using the positive Zonotope **(left)**, and the minimal area Zonotope **(right)**.

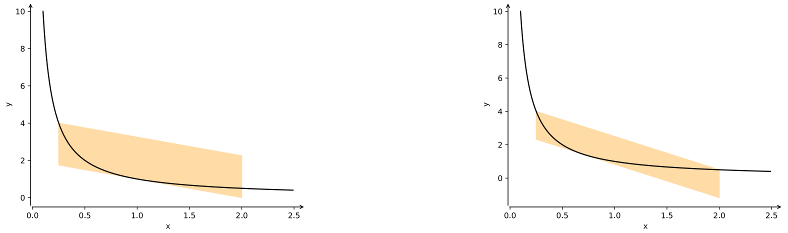

FIGURE 2. Reciprocal abstract transformer for Zonotope using the positive Zonotope **(left)**, and the minimal area Zonotope **(right)**.

4.2. **General abstract transformer construction for HCZ.**

Proposition 7: ***(Double Zonotope HCZ transformer)*** *As the HCZ handles unions, it is possible to construct the HCZ using a union of two zonotopes, which reduces the over-approximation area, while being non-convex. An approximately minimal area HCZ can be defined using a mid point m as $f'(m) = \frac{f(u)-f(l)}{u-l}$, creating two abstract transformer Zonotopes z_1, z_2 on $[l, m]$ and $[m, u]$, and performing a union to create the HCZ abstract transformer.*

Proof. If we only consider the part of the zonotope above the function as a proxy to find a mid point, the area is:

$$A(m) = \underbrace{\frac{f(m)-f(l)}{2}(m-l) - \int_l^m f}_{z_1} + \underbrace{\frac{f(u)-f(m)}{2}(u-m) - \int_m^u f}_{z_1} \tag{15}$$

As $\int_l^m f + \int_m^u f = \int_l^u f$ is constant, the mid point m^* minimizing A can be found directly with:

$$\frac{\mathrm{d}A}{\mathrm{d}m} = 0 \Rightarrow f'(m) = \frac{f(u)-f(l)}{u-l} \tag{16}$$

□

Using this method, the mid point for the exponential is $m^* = \log\left(\frac{e^u - e^l}{u-l}\right)$, and for the reciprocal: $m^* = \sqrt{ul}$.

This method does not work when using additional constraints, and t_{crit_2} instead of t_{crit}.

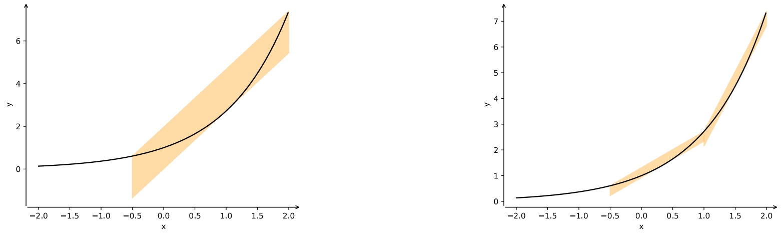

FIGURE 3. Exponential abstract transformer for the minimal area Zonotope vs the approximately minimal area HCZ

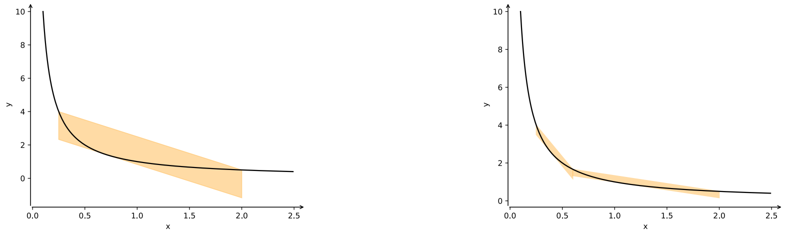

FIGURE 4. Reciprocal abstract transformer for the minimal area Zonotope vs the approximately minimal area HCZ

4.3. **Dot product.**

Proposition 8: ***(Dot product)*** *Let* $Z_1 = \langle c_1, G_1, G_1', A_1, A_1', b_1\rangle, Z_2 = \langle c_2, G_2, G_2', A_2, A_2', b_2\rangle \in \mathbb{R}^N$*, then, the dot product can be computed as* $Z_1 \cdot Z_2 = \langle c, G, G', A, A', b\rangle$ *with:*

$$c = c_1^\top c_2, \quad G = \begin{bmatrix} c_2^\top G_1 & c_1^\top G_2 & \hat{G} \end{bmatrix}, \quad G' = \begin{bmatrix} c_2^\top G_1' & c_1^\top G_2' \end{bmatrix} \tag{17.1}$$

$$A = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{A} \end{bmatrix}, \quad A' = \begin{bmatrix} A'_1 & \mathbf{0} \\ \mathbf{0} & A'_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \hat{b} \end{bmatrix} \tag{17.2}$$

$$\hat{G} = \begin{bmatrix} \hat{l} & 0 & \hat{u} & 0 \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{17.3}$$

$$\hat{l} = (l_1 - c_1)^\top (l_2 - c_2), \quad \hat{u} = (u_1 - c_1)^\top (u_2 - c_2) \tag{17.4}$$

The matrices $\hat{G}, \hat{A}, \hat{b}$ handle the quadratic cross-terms by introducing auxiliary constraints and generators.

Proof.

$$\begin{aligned} Z_1 \cdot Z_2 &= (c_1 + G_1\mathcal{E}_1 + G'_1\mathcal{E}'_1)^\top (c_2 + G_2\mathcal{E}_2 + G'_2\mathcal{E}'_2) && (18.1)\\ &= c_1^\top c_2 && (18.2)\\ &\quad + c_1^\top G_2\mathcal{E}_2 + c_2^\top G_1\mathcal{E}_1 + c_1^\top G'_2\mathcal{E}'_2 + c_2^\top G'_1\mathcal{E}'_1 && (18.3)\\ &\quad + \mathcal{E}_1^\top G_1^\top G_2\mathcal{E}_2 + \mathcal{E}_1^\top G_1^\top G'_2\mathcal{E}'_2 + \mathcal{E}_1'^\top G_1'^\top G_2\mathcal{E}_2 + \mathcal{E}_1'^\top G_1'^\top G'_2\mathcal{E}'_2 && (18.4) \end{aligned}$$

Under the conditions:

$$(C_{1,2})\begin{cases} A_1\mathcal{E}_1 + A'_1\mathcal{E}'_1 = b_1 \\ A_2\mathcal{E}_2 + A'_2\mathcal{E}'_2 = b_2 \end{cases}, \ (C_{\infty,1})\begin{cases} \mathcal{E}_1 \in [-1,1]^{I_1} \\ \mathcal{E}'_1 \in \{-1,1\}^{I'_1} \end{cases}, \ (C_{\infty,2})\begin{cases} \mathcal{E}_2 \in [-1,1]^{I_2} \\ \mathcal{E}'_2 \in \{-1,1\}^{I'_2} \end{cases} \tag{19}$$

By defining $\mathcal{E}$ as $\begin{bmatrix} \mathcal{E}_1 \\ \mathcal{E}_2 \end{bmatrix}$, and $\mathcal{E}'$ as $\begin{bmatrix} \mathcal{E}'_1 \\ \mathcal{E}'_2 \end{bmatrix}$ in Equation 18.3, the new generators can be defined as: $\begin{bmatrix} c_2^\top G_1 & c_1^\top G_2 \end{bmatrix}, \begin{bmatrix} c_2^\top G'_1 & c_1^\top G'_2 \end{bmatrix}$, and the new constraints as: $\begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix}, \begin{bmatrix} A'_1 & \mathbf{0} \\ \mathbf{0} & A'_2 \end{bmatrix}$, which will form a valid HCZ with the same constraints.

To take into account the last term (Equation 18.4), it is possible to bound it into $[l, u]$, and then create new continuous noise terms for $Z_1 \cdot Z_2$.

$$\begin{aligned} \hat{l} &= \min_{C_{1,2}} \min_{C_{\infty,1,2}} \mathcal{E}_1^\top G_1^\top G_2\mathcal{E}_2 + \mathcal{E}_1^\top G_1^\top G'_2\mathcal{E}'_2 + \mathcal{E}_1'^\top G_1'^\top G_2\mathcal{E}_2 + \mathcal{E}_1'^\top G_1'^\top G'_2\mathcal{E}'_2 && (20.1)\\ &= \min_{C_{1,2}} \min_{C_{\infty,1,2}} (\mathcal{E}_1^\top G_1^\top G_2 + \mathcal{E}_1'^\top G_1'^\top G_2)\mathcal{E}_2 + (\mathcal{E}_1^\top G_1^\top G'_2 + \mathcal{E}_1'^\top G_1'^\top G'_2)\mathcal{E}'_2 && (20.2)\\ &= \min_{C_{1,2}} \min_{C_{\infty,1,2}} (Z_1 - c_1)^\top G_2\mathcal{E}_2 + (Z_1 - c_1)^\top G'_2\mathcal{E}'_2 && (20.3)\\ &\geq \min_{C_2} \min_{C_{\infty,2}} (l_1 - c_1)^\top (G_2\mathcal{E}_2 + G'_2\mathcal{E}'_2) && (20.4)\\ &\geq \min_{C_2} \min_{C_{\infty,2}} (l_1 - c_1)^\top (Z_2 - c_2) && (20.5)\\ &\geq (l_1 - c_1)^\top (l_2 - c_2) && (20.6) \end{aligned}$$

Similarly: $\hat{u} \leq (u_1 - c_1)^\top (u_2 - c_2)$. To add the bounded error $\left[\hat{l}, \hat{u}\right]$ to the resulting HCZ, we can add two error terms $\hat{l}\varepsilon_l$ and $\hat{u}\varepsilon_u$, with the constraints: $-1 \leq \varepsilon_l \leq 0$ and $0 \leq \varepsilon_u \leq 1$. These constraints can be expressed as equalities by introducing two new error terms $\tilde{\varepsilon}_l, \tilde{\varepsilon}_u$: $-1 \leq \varepsilon_l \leq 0 \wedge 0 \leq \varepsilon_u \leq 1 \equiv \varepsilon_l + \tilde{\varepsilon}_l = -1 \wedge \varepsilon_u + \tilde{\varepsilon}_u = 1 \wedge \varepsilon_l, \tilde{\varepsilon}_l, \varepsilon_u, \tilde{\varepsilon}_u \in [-1, 1]$, as $\varepsilon_l + \tilde{\varepsilon}_l = -1 \wedge \varepsilon_l, \tilde{\varepsilon}_l \in [-1, 1] \equiv \varepsilon_l \in [-2, 0] \wedge \varepsilon_l \in [-1, 1] \equiv \varepsilon_l \in [-1, 0]$.

The corresponding generators and constraints matrices are then:

$$\hat{G} = \begin{bmatrix} \hat{l} & 0 & \hat{u} & 0 \end{bmatrix}, \hat{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \hat{b} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{21}$$

□

5. Reduction

5.1. **Redundant continuous generators.**

Proposition 9: *Let $i \in \mathbb{N}^I$ and:*

$$\mathcal{E}_{L,i} = \min\left\{\mathcal{E}_i \mid A\mathcal{E} + A'\mathcal{E}' = b, \left\|\mathcal{E}_{j\neq i}\right\|_\infty \leq 1, \mathcal{E}' \in \{-1,1\}^{I'}\right\} \tag{22.1}$$

$$\mathcal{E}_{U,i} = \max\left\{\mathcal{E}_i \mid A\mathcal{E} + A'\mathcal{E}' = b, \left\|\mathcal{E}_{j\neq i}\right\|_\infty \leq 1, \mathcal{E}' \in \{-1,1\}^{I'}\right\} \tag{22.2}$$

Then, if $\left[\mathcal{E}_{L,i}, \mathcal{E}_{U,i}\right] \subseteq [-1,1]$, the i-th noise term can be removed to form the following reduced HCZ [2]:

$$\tilde{Z} = \langle c + \Gamma_G b, G - \Gamma_G A, G' - \Gamma_G A', A - \Gamma_A A, A' - \Gamma_A A', b - \Gamma_A b\rangle \tag{23}$$

where $\Gamma_G = GE_{ik}(A_{ki})^{-1} \in \mathbb{R}^{N\times J}, \Gamma_A = AE_{ik}(A_{ki})^{-1} \in \mathbb{R}^{J\times J}, E_{ik} \in \mathbb{R}^{I\times J}$ is a matrix with zero entries except for a one in the (i,k) position, and $k \in [\![1,J]\!]$ such that $A_{ki} \neq 0$.

Proof. [2] □

While solving Equation 22 would require a MILP, it is possible to overapproximate and find a portion of the candidates generators.

Proposition 10: *For $j \in [\![1,I]\!]$, and the interval $\mathcal{J}_j$ defined by:*

$$\mathcal{J}_j = \bigcap_k \begin{cases} \frac{1}{a_{kj}}\left[b_k - \sum_{i\neq j}|a_{ki}| - \sum_i |a'_{ki}|, b_k + \sum_{i\neq j}|a_{ki}| + \sum_i |a'_{ki}|\right], \text{if } a_{kj} \neq 0 \\ [-\infty,\infty], \text{else} \end{cases} \tag{24}$$

we have $\mathcal{E}_j \subseteq \mathcal{J}_j$. Thus $\mathcal{J}_j \subseteq [-1,1] \Rightarrow \mathcal{E}_j \in [-1,1]$, and $\mathcal{E}_j$ can be a candidate to reduction.

Proof. Let $j \in [\![1,I]\!]$, and $k \in [\![1,J]\!]$, $A_k\mathcal{E} + A'_k\mathcal{E}' = b_k$ can be rewritten:

$$a_{kj}\varepsilon_j = b_k - \sum_{i\neq j} a_{ki}\varepsilon_i - \sum_i a'_{ki}\varepsilon'_i \tag{25}$$

If ε_j does not appear in the equation, $a_{kj} = 0$ and the equation does not add any information on ε_j, $\varepsilon_j \in [-\infty,\infty]$. If $a_{kj} \neq 0$, the equation gives a lower and upper bound on ε_j:

$$\frac{1}{a_{kj}}b_k - \sum_{i\neq j}|a_{ki}| - \sum_i |a'_{ki}| \leq \varepsilon_j \leq \frac{1}{a_{kj}}b_k + \sum_{i\neq j}|a_{ki}| + \sum_i |a'_{ki}| \tag{26}$$

Let's name this interval $\mathcal{J}_{jk}$, then every line gives a new constraint, either $[-\infty,\infty]$, or $\mathcal{J}_{jk}$, and ε_j can be bounded by $\mathcal{J}_j = \bigcap_k \mathcal{J}_{jk} \vee [-\infty,\infty]$. □

Even if the hypothesis Equation 22 is not valid, it is still possible to apply the reduction. The resulting Zonotope will then be an over-approximation of the initial one [2].

5.2. Additional trivial checks.

For every constraint j, if $b_j = \left\|A_j\right\|_1 = \left\|A'_j\right\|_1 = 0$, then the constraint can be removed. If $\left|b_j\right| > \left\|A_j\right\|_1 + \left\|A'_j\right\|_1$, then the zonotope is empty.

For every generator i, if $\|G_i\|_1 = \|A_i\|_1 = 0$, the continuous generator i can be removed. If $\|G'_i\|_1 = \|A'_i\|_1 = 0$, the binary generator i can be removed.

6. Implementation

6.1. Tuning hyperparameters.

If $\|\Lambda\|$ is big, the norms in the concretisation ($\|G - \Lambda A\|_1$) will add a huge over-approximation, even if G_i is zero for a given variable. This occurs when the number of noise generators is large (~1000). To reduce this over-approximation, it is important to choose well the hyperparameters, especially the learning rate. Similar to LLM training, it can be chosen very small (~$1e - 5$).

6.2. Sparse tensors.

The union makes the size of the tensors grow exponentially, which quickly leads to OOM errors. For instance, $N = 1000$ with $I = 1000$ requires approximately 10 Go of VRAM. However, as the tensors are mostly empty, it is possible to use sparse matrices, which considerably reduces the memory footprint. For instance, the same parameters ($N = 1000, I = 1000$) takes only 300Mo of VRAM with sparse tensors. Which makes it usable in practice.

Appendix

A.1 Other dot product.

Proposition 11: ***(Dot product 2nd version, reduces the number of noise, but less precise)*** *Let* $Z_1 = \langle c_1, G_1, G'_1, A_1, A'_1, b_1\rangle, Z_2 = \langle c_2, G_2, G'_2, A_2, A'_2, b_2\rangle \in \mathbb{R}^N$*, then, the dot product can be computed as* $Z_1 \cdot Z_2 = \langle c, G, G', A, A', b\rangle$ *with:*

$$c = c_1^\top c_2 + m_1^\top m_2 - m_1^\top c_2, \quad G = \left[c_2^\top G_1 \;\; c_2^\top \delta_2 \;\; m_1^\top \delta_2 + \left|\delta_1^\top \delta_2\right|\right] \tag{27.1}$$

$$G' = \left[c_2^\top G'_1\right], \quad A = [A_1 \;\; 0 \;\; 0], \quad A' = A'_1, \quad b = b_1 \tag{27.2}$$

$$m_1 = \frac{u_1 + l_1}{2}, \quad m_2 = \frac{u_2 + l_2}{2}, \quad \delta_1 = \frac{u_1 - l_1}{2}, \quad \delta_2 = \frac{u_2 - l_2}{2}, \tag{27.3}$$

Proof.

$$Z_1 \cdot Z_2 = Z_1^\top (c_2 + G_2 \mathcal{E}_2 + G'_2 \mathcal{E}'_2) \tag{28.1}$$

$$= c_1^\top c_2 + c_2^\top G_1 \mathcal{E}_1 + c_2^\top G'_1 \mathcal{E}'_1 + Z_1^\top G_2 \mathcal{E}_2 + Z_1^\top G'_2 \mathcal{E}'_2 \tag{28.2}$$

$Z_1 \subseteq [l_1, u_1] = \frac{u_1+l_1}{2} + \frac{u_1-l_1}{2}[-1, 1] = m_1 + \delta_1 \xi_1, \xi_1 \in [-1, 1]$, and $G_2 \mathcal{E}_2 + G'_2 \mathcal{E}'_2 = Z_2 - c_2 \subseteq \frac{u_2+l_2}{2} + \frac{u_2-l_2}{2}[-1, 1] - c_2 = m_2 - c_2 + \delta_2 \xi_2, \xi_2 \in [-1, 1]$. Then,

$$Z_1^\top (G_2 \mathcal{E}_2 + G'_2 \mathcal{E}'_2) \subseteq m_1^\top m_2 - m_1^\top c_2 + m_1^\top \delta_2 \xi_2 + \xi_1^\top \delta_1^\top \delta_2 \xi_2 - \xi_1^\top \delta_1^\top c_2 \tag{29}$$

The quadratic term was not removed but changed into $\xi_1^\top \delta_1^\top \delta_2 \xi_2$. The difference is that the constraints also moved, δ_1 and δ_2 were computed taking into account the constraints of Z_1 and Z_2, while the new error terms ξ_1, ξ_2 don't have constraints. Thus, the new quadratic term can be bounded simply with:

$$\xi_1^\top \delta_1^\top \delta_2 \xi_2 \subseteq \left|\delta_1^\top \delta_2\right| \xi_2 \tag{30}$$

Hence:

$$Z_1^\top (G_2 \mathcal{E}_2 + G_2' \mathcal{E}_2') \subseteq m_1^\top m_2 - m_1^\top c_2 + (m_1^\top \delta_2 + \left|\delta_1^\top \delta_2\right|)\xi_2 + c_2^\top \delta_1 \xi_1 \tag{31}$$

□

References

1. Albarghouthi, A.: Introduction to Neural Network Verification. (2021)
2. Bird, T.J.: Hybrid Zonotopes: A Mixed-Integer Set Representation for the Analysis of Hybrid Systems. (2022)
3. Mark Niklas Müller, M.V., Mislav Balunović: Certify or Predict: Boosting Certified Robustness with Compositional Architectures. (2021)

Sécurité des systèmes et des réseaux, Télécom SudParis, Évry, France
Email address: thomas.winninger@télécom-sudparis.eu
URL: https://le-magicien-quantique.github.io

ANTIQUE, INRIA, Paris, France
Email address: caterina.urban@inria.fr
URL: https://caterinaurban.github.io

ANTIQUE, INRIA, Paris, France
Email address: guannan.wei@inria.fr
URL: https://continuation.passing.style