PARAMETERIZED HARDNESS OF ZONOTOPE CONTAINMENT AND NEURAL NETWORK VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks with ReLU activations are a widely used model in machine learning. It is thus important to have a profound understanding of the properties of the functions computed by such networks. Recently, there has been increasing interest in the (parameterized) computational complexity of determining these properties. In this work, we close several gaps and resolve an open problem posted by Froese et al. [COLT '25] regarding the parameterized complexity of various problems related to network verification. In particular, we prove that deciding positivity (and thus surjectivity) of a function $f\colon \mathbb{R}^d \to \mathbb{R}$ computed by a 2-layer ReLU network is W[1]-hard when parameterized by d. This result also implies that zonotope (non-)containment is W[1]-hard with respect to d, a problem that is of independent interest in computational geometry, control theory, and robotics. Moreover, we show that (a) approximating the maximum within any multiplicative factor in 2-layer ReLU networks, (b) computing the L_p-Lipschitz constant for $p \in (0, \infty]$ in 2-layer networks, and (c) approximating the L_p-Lipschitz constant in 3-layer networks are all NP-hard and W[1]-hard with respect to d. Notably, our hardness results are the strongest known so far and imply that the naive enumeration-based methods for solving these fundamental problems are all essentially optimal under the Exponential Time Hypothesis.

1 INTRODUCTION

Neural networks with rectified linear unit (ReLU) activations are a common model in deep learning. In practice, such networks are trained on finite datasets and are expected to generalize reliably to unseen inputs. However, even minor perturbations of the input may lead to unexpected or erroneous outputs (Szegedy et al., 2014). This highlights the importance of certification of trained models, which in turn requires a detailed understanding of the functions computed by ReLU networks.

A central problem in this context is *network verification*: Given a subset of inputs $\mathcal{X}$, the question is whether the network's outputs are guaranteed to lie within a prescribed set $\mathcal{Y}$. Commonly, the sets $\mathcal{X}$ and $\mathcal{Y}$ take the form of balls or are specified by linear constraints. This question has received increasing attention in recent years, particularly due to the deployment of neural networks in safety-critical applications (Bojarski et al., 2016; Weng et al., 2018a; Kouvaros & Lomuscio, 2021; Rössig & Petkovic, 2021; Katz et al., 2022). Recently, Froese et al. (2025b) established a connection between the basic verification task to decide whether a 2-layer ReLU network attains a positive output (which is equivalent to surjectivity) and the classical geometry problem of *zonotope containment*. The latter asks whether one zonotope is contained within another, a question that has been extensively studied due to its applications in areas such as robotics and control (Sadraddini & Tedrake, 2019; Gruber & Althoff, 2020; 2021; Kulmburg & Althoff, 2021; Yang et al., 2022; Kulmburg et al., 2025).

Beyond verification, robustness is often a crucial requirement since trained networks are typically expected to be insensitive to small input perturbations. This property is commonly quantified in terms of the network's *Lipschitz constant*, which should ideally be small (Virmaux & Scaman, 2018; Weng et al., 2018b; Fazlyab et al., 2019; Jordan & Dimakis, 2020).

Network verification (Katz et al., 2022; Sälzer & Lange, 2022; Froese et al., 2025b), estimating the Lipschitz constant (Virmaux & Scaman, 2018; Jordan & Dimakis, 2020) and zonotope containment (Kulmburg & Althoff, 2021) are all known to be (co)NP-hard. This intractability is closely linked to the curse of dimensionality: As the input dimension d grows, the search space becomes prohibitively large. A natural follow-up question is whether these problems become tractable for low-dimensional input spaces. This perspective is particularly relevant since, in practice, high-dimensional data is often assumed to lie near a low-dimensional submanifold of the input space. Motivated by this, recent work has studied the *parameterized complexity* of neural network problems such as training (Arora et al., 2018; Froese et al., 2022; Brand et al., 2023; Froese & Hertrich, 2023) and verification (Froese et al., 2025b). Notably, while checking injectivity of a 2-layer ReLU network with n hidden neurons can be done in $(d+1)^d \cdot n^{O(1)}$ time (that is, *fixed-parameter tractability* with respect to d) (Froese et al., 2025b), the parameterized complexity status of network verification (in particular positivity) and the Lipschitz constant have been posed as open problems at COLT '25 (Froese et al., 2025a).

1.1 Our Contributions

We answer the aforementioned questions by proving W[1]-hardness for the parameter input dimension (thus excluding fixed-parameter tractability under standard complexity assumptions). Moreover, we show that solving these problems via simple "brute-force" enumeration of the linear regions of the network's function is essentially optimal under the Exponential Time Hypothesis (ETH).

In Section 3, we give a reduction from the well-known Multicolored Clique problem to network verification in which the network's input dimension depends linearly on the clique size. This reduction forms the basis for our hardness results and yields strong lower bounds based on the ETH. The key difficulty here is that the input dimension must scale linearly with the clique size (in contrast, standard NP-hardness reductions allow the input dimension to grow without restriction).

Network Verification. We study the (co)NP-hard problems of deciding positivity, surjectivity, and approximating the maximum of a 2-layer ReLU network $f: \mathbb{R}^d \to \mathbb{R}$ (with n hidden neurons), and also the problem of deciding whether a 3-layer ReLU network computes the constant zero function. All these problems are special cases of (complements of) verification. For example, positivity corresponds to checking whether there exists $x \in \mathbb{R}^d$ with $f(x) > 0$, that is, $f(\mathbb{R}^d) \nsubseteq (-\infty, 0]$. All these problems can be solved in $n^{\mathcal{O}(d)} \cdot \mathrm{poly}(N)$ time with simple "brute-force" enumeration algorithms (see Section 2). In Section 4, we prove W[1]-hardness with respect to d for all problems, thereby resolving the open question by Froese et al. (2025a). Our reductions imply a running time lower bound of $n^{\Omega(d)} \cdot \mathrm{poly}(N)$ based on the ETH which shows that the simple enumeration algorithms are essentially optimal. In particular, this implies an $n^{\Omega(d)} \cdot \mathrm{poly}(N)$-time lower bound for the general network verification problem.

Zonotope Containment. In Section 5, we study the coNP-hard problem of deciding whether a zonotope $Z \subset \mathbb{R}^d$ (given by its generators) is contained in another zonotope $Z' \subset \mathbb{R}^d$. Based on a duality of 2-layer ReLU networks and zonotopes, we obtain W[1]-hardness with respect to d and an analogous running time lower bound of $n^{\Omega(d)} \cdot \mathrm{poly}(N)$ assuming the ETH which shows that the simple vertex enumeration algorithm is essentially optimal.

Lipschitz Constant. Virmaux & Scaman (2018) proved that computing the L_2-Lipschitz constant of a 2-layer ReLU network is NP-hard. In Section 6, we extend this to NP-hardness for every $p \in (0, \infty]$ and even W[1]-hardness with respect to d. Approximating the L_p-Lipschitz constant within any multiplicative constant for 3-layer ReLU networks is known to be NP-hard (Jordan & Dimakis, 2020; Froese et al., 2025b). We also extend this result to W[1]-hardness with respect to d. Again, our reductions imply running time lower bounds matching the running times of simple enumeration algorithms. On the positive side, we show that for the restricted class of *input convex* networks, computing the L_1-Lipschitz constant is polynomial-time solvable and the L_∞-Lipschitz constant is *fixed-parameter tractable* (FPT) with respect to d. In Section 7, we discuss the equivalence between Lipschitz constant computation and norm maximization on zonotopes and present a randomized FPT-approximation algorithm, using results from subspace embeddings.

Limitations. Our paper is clearly of purely theoretical nature. We aim for a thorough understanding of the problems from a computational complexity perspective. Hence, our results are naturally worst-case results. Although the algorithms we give are essentially optimal in terms of running time (assuming the ETH), it might be possible to achieve a better running time by reducing the constant hidden in the exponent. Moreover, additional assumptions on the network structure might render the problems tractable (as in the case of input convex networks for the L_1-Lipschitz constant). A full literature review (e.g., for the broad field of network verification) is beyond the scope of this paper.

1.2 Further Related Work

Various heuristic methods for network verification have been proposed, including interval bound propagation (Gowal et al., 2018), DeepZ (Wong et al., 2018), DeepPoly (Singh et al., 2019), multi-neuron verification Ferrari et al. (2022), ZonoDual (Jordan et al., 2022), and cutting planes (Zhang et al., 2022). Baader et al. (2024) and Mao et al. (2024) study the expressivity of convex relaxations that are often used in practical network verification algorithms. L_p-norm maximization on zonotopes is also known as the *Longest Vector Sum* problem and has a wide range of applications in pattern recognition, clustering, signal processing, and analysis of large-scale data (Baburin & Pyatkin, 2007; Shenmaier, 2018; 2020). Special cases were studied before (Bodlaender et al., 1990; Ferrez et al., 2005).

2 Preliminaries

Notation. For $n \in \mathbb{N}$, we define $[n] \coloneqq \{1, \dots, n\}$. For $k, n \in \mathbb{N}, k \leq n$, we define $\binom{[n]}{k} \coloneqq \{A \subseteq [n] : |A| = k\}$. A function $f : \mathbb{R}^d \to \mathbb{R}^m$ is *positively homogeneous* if $f(\lambda x) = \lambda f(x)$ holds for all $x \in \mathbb{R}^d$ and $\lambda \geq 0$. Given a *generator* matrix $A = (a_1, \dots, a_n) \in \mathbb{R}^{d \times n}$, the corresponding *zonotope* is $Z(A) \coloneqq \sum_{i=1}^n \operatorname{conv}(\{0, a_i\})$, where the sum is the Minkowski sum of the generators.

L_p-Lipschitz Constant. For $p \in (0, \infty)$ and a vector $x \in \mathbb{R}^d$, we define $\|x\|_p \coloneqq \left(\sum_{i=1}^d |x_i|^p\right)^{\frac{1}{p}}$, and for $p = \infty$ we set $\|x\|_\infty \coloneqq \max_{i \in [d]} |x_i|$. For $p \in [1, \infty]$, the function $\|\cdot\|_p$ is the L_p-*norm*, and for $p \in (0, 1)$, it is the L_p-*quasinorm*. The L_0-function is defined by $\|x\|_0 \coloneqq |\{i \in [d] : x_i \neq 0\}|$. The L_p-Lipschitz constant of a function f is $L_p(f) \coloneqq \sup_{x \neq y} \frac{\|f(x) - f(y)\|_p}{\|x - y\|_p}$.

ReLU Networks. A *ReLU layer* with d inputs, m outputs, weights $W \in \mathbb{R}^{m \times d}$, and biases $b \in \mathbb{R}^m$ computes the map $\phi_{W,b} : \mathbb{R}^d \to \mathbb{R}^m,\ x \mapsto \max(0, Wx + b)$, where the maximum is applied in each component. A *ReLU network* with $\ell \geq 1$ layers and one-dimensional output is defined by ℓ weight matrices $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and biases $b_i \in \mathbb{R}^{n_i}$ for $i \in [\ell]$, where $n_0 \coloneqq d, \dots, n_\ell \coloneqq 1 \in \mathbb{N}^+$, and computes the *continuous piecewise linear* (CPWL) function $f : \mathbb{R}^d \to \mathbb{R}$ with

$$f(x) \coloneqq W_\ell \cdot (\phi_{W_{\ell-1}, b_{\ell-1}} \circ \cdots \circ \phi_{W_1, b_1})(x) + b_\ell.$$

Observe that no activation function is applied in the output layer. The $\ell - 1$ ReLU layers are also called *hidden layers*. The *width* and *size* of the network are $\max\{n_1, \dots, n_{\ell-1}\}$ and $\sum_{i=1}^{\ell-1} n_i$, respectively. Additional details can be found in Appendix A.

Polytopes and Duality. There is a duality between positively homogeneous convex CPWL functions from $\mathbb{R}^d$ to $\mathbb{R}$ (the set of which is denoted $\mathcal{F}_d$) and polytopes in $\mathbb{R}^d$ (denoted $\mathcal{P}_d$), which we will briefly sketch. Any function $f \in \mathcal{F}_d$ can be written as $f(x) = \max\{a_1^\top x, \dots, a_k^\top x\}$ for some $a_i \in \mathbb{R}^d$, and its *Newton polytope* is $\operatorname{Newt}(f) \coloneqq \operatorname{conv}(\{a_1, \dots, a_k\})$. Equivalently, f is the *support function* of $\operatorname{Newt}(f)$, that is, $f(x) = \max_{y \in \operatorname{Newt}(f)} y^\top x$. The map $\varphi : \mathcal{F}_d \to \mathcal{P}_d$, defined by $f \mapsto \operatorname{Newt}(f)$, is a bijection satisfying $\varphi(f + g) = \varphi(f) + \varphi(g)$ and $\varphi(\max\{f, g\}) = \operatorname{conv}(\varphi(f) \cup \varphi(g))$, where $+$ denotes pointwise addition or Minkowski sum, respectively.

Parameterized Complexity. We assume basic knowledge on computational complexity theory. Parameterized complexity is a multivariate approach to study the time complexity of computational problems (Cygan et al., 2015; Downey & Fellows, 2013). A *parameterized problem* $L \subseteq \Sigma^* \times \mathbb{N}$ consists of instances (x, k) where x encodes a classical problem instance and k is a *parameter*. A parameterized problem L is *fixed-parameter tractable* (contained in the class FPT) if it can be solved in $f(k) \cdot |x|^{\mathcal{O}(1)}$ time, where f is an arbitrary function that only depends on k. The class XP contains all parameterized problems which can be solved in polynomial time for constant parameter values, that is, in $f(k) \cdot |x|^{g(k)}$ time, where g is an arbitrary function that only depends on k. It is known that FPT $\subsetneq$ XP. The class W[1] can be defined as the set of all parameterized problems which can be reduced to CLIQUE (with parameter solution size) via a *parameterized reduction*. It is known that FPT $\subseteq$ W[1] $\subseteq$ XP and it is widely believed that W[1] contains problems which are not in FPT (namely the W[1]-hard problems such as CLIQUE). A parameterized reduction from L to L' is an algorithm that maps an instance (x, k) in $f(k) \cdot |x|^{\mathcal{O}(1)}$ time to an instance (x', k') such that $k' \leq g(k)$ for an arbitrary function g and $(x, k) \in L$ if and only if $(x', k') \in L'$.

The *Exponential Time Hypothesis* (Impagliazzo & Paturi, 2001) states that 3-SAT on n variables cannot be solved in $2^{o(n)}$ time. The ETH implies FPT $\neq$ W[1] (which implies P $\neq$ NP), as well as running time lower bounds: For example, CLIQUE cannot be solved in $\rho(k) \cdot n^{o(k)}$ time, where k is the size of the requested clique and n is the number of nodes in the graph (Cygan et al., 2015).

2.1 PROBLEM DEFINITIONS AND WARM-UP

For given generator matrices $A \in \mathbb{R}^{d\times n}$ and $B \in \mathbb{R}^{d\times m}$, and a scalar $L \in \mathbb{R}$, we consider the following problems:

- ZONOTOPE CONTAINMENT: Is $Z(A) \subseteq Z(B)$?
- L_p-MAX ON ZONOTOPES: Is $\max_{x\in Z(A)} \|x\|_p \geq L$?

For an ℓ-layer ReLU network defined by weight matrices $W_i \in \mathbb{R}^{n_i\times n_{i-1}}$ and biases $b_i \in \mathbb{R}^{n_i}$ for $i \in [\ell]$, where $d \coloneqq n_0, \dots, n_\ell \coloneqq 1 \in \mathbb{N}^+$ that computes the function $f \colon \mathbb{R}^d \to \mathbb{R}$, $f(x) \coloneqq W_\ell \cdot (\phi_{W_{\ell-1},b_{\ell-1}} \circ \cdots \circ \phi_{W_1,b_1})(x) + b_\ell$, we consider the following problems:

- ℓ-LAYER RELU POSITIVITY: Is there an $x \in \mathbb{R}^d$ such that $f(x) > 0$?
- ℓ-LAYER RELU SURJECTIVITY: Is f surjective (that is, $\forall y \in \mathbb{R}\ \exists x \in \mathbb{R}^d : f(x) = y$)?
- ℓ-LAYER RELU L_p-LIPSCHITZ CONSTANT: Is $L_p(f) \geq L$?

In fact, all these problems are known to be in XP for the parameter d (simply enumerate vertices of zonotopes and linear regions of ReLU networks; see Appendix B for more details).

Theorem 2.1. ZONOTOPE CONTAINMENT *and* L_p-MAX ON ZONOTOPES *can be solved in* $\mathcal{O}(n^{d-1} \cdot \mathrm{poly}(N))$ *time (where n is the number of generators and N is the input bit-length).*

Theorem 2.2. ℓ-LAYER RELU POSITIVITY, ℓ-LAYER RELU SURJECTIVITY, ℓ-LAYER RELU L_p-LIPSCHITZ CONSTANT, *computing the maximum of an ℓ-layer ReLU network over a polyhedron and deciding whether an ℓ-layer ReLU network computes the zero function can be solved in* $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ *time (where n is the network width and N is the input bit-length).*

In particular, we prove in Appendix B that network verification for ℓ-layer ReLU networks $f : \mathbb{R}^d \to \mathbb{R}^m$ is solvable in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time, assuming that $\mathcal{X}$ and $\mathcal{Y}$ are polyhedra in halfspace representation. Later, we will prove that, assuming the ETH, the 2-layer or the 3-layer versions of all of these problems cannot be solved in $\rho(d) \cdot N^{o(d)}$ time for any function ρ, which means that the $\mathcal{O}(n^d \cdot \mathrm{poly}(N))$- and $\mathcal{O}(n^{2d} \cdot \mathrm{poly}(N))$-time algorithms (for 2- and 3-layer networks) are essentially optimal with respect to the runtime dependency on d. Note that hardness results for 2- or 3-layer networks also imply hardness for deeper networks with $\ell \geq 3$ layers: simply concatenate the 2- or 3-layer network with trivial additional layers that compute the identity map.

3 REDUCTION FROM MULTICOLORED CLIQUE

In this section, we present a parameterized reduction which forms the basis for the hardness results for all our considered problems. (All proofs that are omitted from the main text as well as some auxiliary statements can be found in Appendix B.) We reduce from the following problem.

MULTICOLORED CLIQUE
Input: A graph $G = (V = V_1 \dot\cup \cdots \dot\cup V_k, E)$, where each node in V_i has color i.
Question: Does G have a k-colored clique (a clique with exactly one node of each color)?

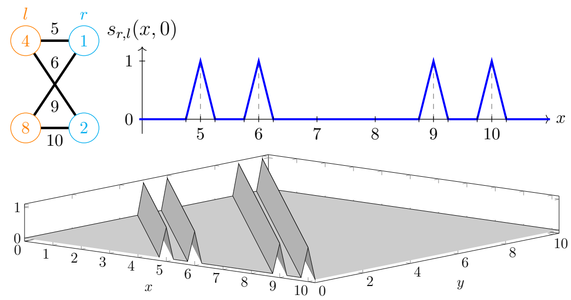

Figure 1: Spike function $s_{r,l}$ for a colored graph (top left). Node labels: $\omega_{r,1} = 1, \omega_{r,2} = 2, \omega_{l,1} = 4, \omega_{l,2} = 8$. Edge labels: $\omega_{r,1,l,1} = 5, \omega_{r,2,l,1} = 6, \omega_{r,1,l,2} = 9, \omega_{r,2,l,2} = 10$.

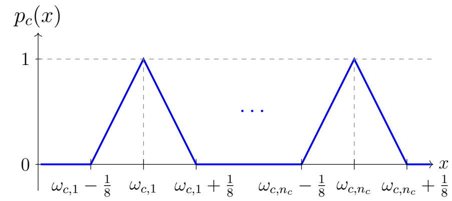

Figure 2: Penalty function p_c.

MULTICOLORED CLIQUE is NP-hard, W[1]-hard with respect to k and not solvable in $\rho(k) \cdot |V|^{o(k)}$ time for any computable function ρ assuming the ETH (Cygan et al., 2015).

Proposition 3.1. *For every instance* $(G = (V = V_1 \dot\cup \cdots \dot\cup V_k, E), k)$ *of* MULTICOLORED CLIQUE, *it is possible to construct in polynomial time a 2-layer ReLU network computing a function* $f \colon \mathbb{R}^k \to \mathbb{R}$ *such that* $\max_{x \in \mathbb{R}^k} f(x) = k + \binom{k}{2}$ *if and only if* G *contains a* k*-colored clique and* $\max_{x \in \mathbb{R}^k} f(x) \leq k + \binom{k}{2} - 1$ *otherwise.*

Proof Sketch. Let $(G = (V = V_1 \dot\cup \cdots \dot\cup V_k, E), k)$ be an instance of MULTICOLORED CLIQUE, where $V_c = \{v_{c,1}, \ldots, v_{c,n_c}\}$ and $E = \bigcup_{(r,l) \in \binom{[k]}{2}} E_{r,l}$, where $E_{r,l}$ denotes the set of edges whose nodes have color r and l. We assign each node $v_{c,i}$ a unique label $\omega_{c,i} \in \mathbb{N}$ such that every edge $\{v_{r,i}, v_{l,j}\}$ gets a unique label $\omega_{r,i,l,j} := \omega_{r,i} + \omega_{l,i}$ (using Sidon sets, see Appendix B for details).

For every color pair $(r, l) \in \binom{[k]}{2}$, we introduce a *spike function* $s_{r,l} \colon \mathbb{R}^2 \to [0, 1]$ (see Figure 1) that is zero everywhere except for a set of $|E_{r,l}|$ parallel stripes in which $s_{r,l}$ forms a spike, that is, goes up from 0 to 1 and goes down from 1 to 0 again. The spike function attains value 1 if and only if the sum of its inputs is equal to $\omega_{r,i,l,j}$ for some edge $\{v_{r,i}, v_{l,j}\} \in E_{r,l}$. The spike function can be implemented with $3|E_{r,l}|$ neurons. For every color $c \in [k]$, we create a *penalty function* $p_c \colon \mathbb{R} \to [0, 1]$ (see Figure 2) that has a narrow spike around the value $\omega_{c,i}$ for each node $v_{c,i}$ and is zero everywhere else. The penalty function p_c can be implemented with $3n_c$ neurons.

By computing all spike and penalty functions in parallel and summing them up, we obtain a 2-layer ReLU network with $3(|V| + |E|)$ ReLU neurons that computes $f \colon \mathbb{R}^k \to [0, k + \binom{k}{2}]$ with

$$f(x_1, \ldots, x_k) = \sum_{(r,l) \in \binom{[k]}{2}} s_{r,l}(x_r, x_l) + \sum_{c \in [k]} p_c(x_c).$$

Next, we show that if there exists a k-colored clique $\{v_{1,a_1}, \ldots, v_{k,a_k}\}$ in G, then $f((\omega_{1,a_1}, \ldots, \omega_{k,a_k})) = k + \binom{k}{2}$. On the other hand, we show that if there is a point $x^* \in \mathbb{R}^k$ with $f(x^*) > k + \binom{k}{2} - 1$, then G has a k-colored clique. The idea is that in this case, all spike and penalty functions must have positive output. For the penalty functions, this means that every input value x_c^* must be close to a value ω_{c,a_c} which corresponds to the node v_{c,a_c}. Since the spike functions only give a positive output if the two node inputs correspond to adjacent nodes, the nodes $v_{1,a_1}, \ldots, v_{k,a_k}$ then form a k-colored clique in G. □

In the following, we will use modifications of this construction to prove our hardness results. All our (parameterized) reductions are in fact *polynomial-time reductions* and thus also prove NP-hardness. We will only state this explicitly if the NP-hardness of the problem was not previously known.

4 HARDNESS OF NETWORK VERIFICATION PROBLEMS

We first prove W[1]-hardness (w.r.t. d) of 2-LAYER RELU POSITIVITY. The NP-hardness of 2-LAYER RELU POSITIVITY was established by Froese et al. (2025b). We prove W[1]-hardness via

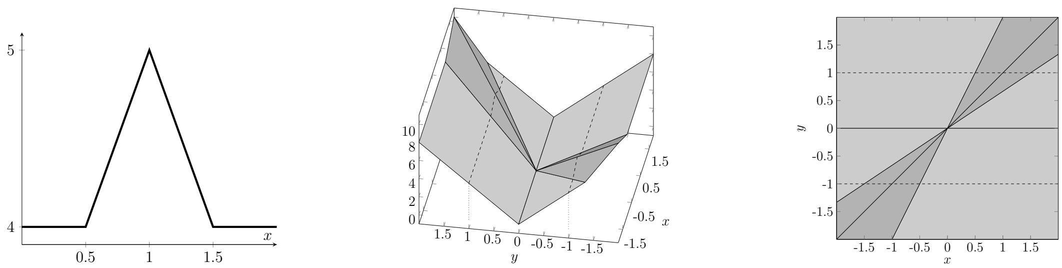

Figure 3: Homogenization: the function $\max(0, 2x - 1) - \max(0, 4x - 4) + \max(0, 2x - 3) + 4$ (left) is turned into $\max(0, 2x - y) - \max(0, 4x - 4y) + \max(0, 2x - 3y) + 4|y|$ (right).

the reduction from Proposition 3.1, which relies on the use of biases. To extend the hardness result to other problems, we need to show a stronger statement: that 2-LAYER RELU POSITIVITY remains W[1]-hard even when all biases are equal to zero. For this, we use *homogenized* ReLU networks.

Definition 4.1. Given a 2-layer ReLU network with a single output neuron, its *homogenization* is the ReLU network (with all biases equal to zero) that is obtained by adding an extra input variable y to the network, replacing all biases b of neurons in the first hidden layer by $y \cdot b$ and replacing the bias b of the output neuron by $|y| \cdot b$ using two extra neurons in the hidden layer.

Figure 3 illustrates the effect of homogenization on the function of a 2-layer ReLU network.

Theorem 4.2. 2-LAYER RELU POSITIVITY *is W[1]-hard with respect to d and not solvable in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH, even if all biases are zero.*

Proof Sketch. Setting the output node bias of the ReLU network constructed in the proof of Proposition 3.1 to $1 - k - \binom{k}{2}$ yields a network that has a positive output if and only if the graph G from the MULTICOLORED CLIQUE instance contains a k-colored clique. We then show that homogenizing this network preserves this equivalence, which yields a parameterized reduction from MULTICOLORED CLIQUE to 2-LAYER RELU POSITIVITY without biases (and thus proves W[1]-hardness). Note that the input dimension d of the constructed network is $k + 1$. Hence, any algorithm solving 2-LAYER RELU POSITIVITY in $\rho(d) \cdot N^{o(d)}$ time would imply an algorithm for MULTICOLORED CLIQUE running in $\rho(k) \cdot |V|^{o(k)}$ time (since $N \leq |V|^{\mathcal{O}(1)}$) contradicting the ETH. $\square$

Theorem 4.2 also implies W[1]-hardness w.r.t. the input dimension d for approximating the maximum of a 2-layer ReLU network over a polyhedron within any multiplicative factor. Froese et al. (2025b, Corollary 13) showed that approximating this value is NP-hard.

Corollary 4.3. *Approximating the maximum of a 2-layer ReLU network over a polyhedron within any multiplicative factor is W[1]-hard with respect to its input dimension d and cannot be done in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

By adding another hidden layer with a single ReLU neuron to the network constructed in the proof of Theorem 4.2, we obtain a 3-layer ReLU network that has a non-zero output if and only if the original 2-layer network has a positive output. This yields the following corollary.

Corollary 4.4. *The problem of deciding whether a 3-layer ReLU network computes a non-zero function is W[1]-hard with respect to its input dimension d and not solvable in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

The NP-hardness of the above problem was established by Froese et al. (2025b). For 2-layer networks, it is solvable in polynomial time (Froese et al., 2025b), which holds also in the presence of biases (Stargalla et al., 2025). Thus, Corollary 4.4 draws an even clearer boundary between the computational complexity of this problem in the 2-layer and 3-layer cases.

Froese et al. (2025b) proved NP-hardness of 2-LAYER RELU SURJECTIVITY and asked whether the problem is fixed-parameter tractable with respect to d. We give a negative answer to this question.

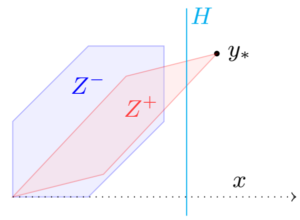

Figure 4: An illustration of the equivalence between 2-LAYER RELU POSITIVITY and ZONOTOPE NON-CONTAINMENT. Let $H = \{y \in \mathbb{R}^d \mid y^\top x = b\}$ be a hyperplane that separates y_* from Z^-. Then $g(x) = \max_{y \in Z^+} y^\top x - \max_{y \in Z^-} y^\top x > y_*^\top x - b > 0$.

Theorem 4.5. 2-LAYER RELU SURJECTIVITY *is $W[1]$-hard with respect to d and not solvable in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

Proof. Recall that a positively homogeneous function $g\colon \mathbb{R}^d \to \mathbb{R}$ is surjective if and only if there exist two points $v^+, v^- \in \mathbb{R}^d$ such that $g(v^+) > 0$ and $g(v^-) < 0$. The positively homogeneous function $f\colon \mathbb{R}^{k+1} \to \mathbb{R}$ of the 2-layer ReLU network from the proof of Theorem 4.2 is in fact surjective if and only if it has a positive point, as $f(\mathbf{0}, 1) < 0$. □

5 HARDNESS OF ZONOTOPE NON-CONTAINMENT

In this section, we prove W[1]-hardness for ZONOTOPE NON-CONTAINMENT, the complement of ZONOTOPE CONTAINMENT. ZONOTOPE CONTAINMENT is coNP-complete, and can be solved in $O(n^{d-1} \cdot \mathrm{poly}(N))$ time (where N is the input bit-length) by enumerating the vertices of one zonotope, but fixed-parameter tractability with respect to the dimension d remained open so far (Froese et al., 2025a). Moreover, Kulmburg & Althoff (2021) showed that containment is equivalent to maximizing a certain zonotope norm, making it a special case of norm maximization on zonotopes.

Froese et al. (2025b) showed that 2-LAYER RELU POSITIVITY is equivalent to ZONOTOPE NON-CONTAINMENT following from the duality of positively homogeneous convex CPWL functions from $\mathbb{R}^d$ to $\mathbb{R}$ and polytopes in $\mathbb{R}^d$. We will briefly sketch this equivalence. Let a 2-layer ReLU network without biases be given by $g(x) = \sum_{i=1}^m \lambda_i \max\{0, w_i^\top x\}$. We can assume without loss of generality that $\lambda_i \in \{-1, 1\}$ due to the positive homogeneity of $\max\{0, w_i^\top x\}$. Hence $g(x) = \sum_{i \in I^+} \max\{0, w_i^\top x\} - \sum_{i \in I^-} \max\{0, w_i^\top x\}$ for vectors $w_i \in \mathbb{R}^d, i \in I^+ \cup I^-$. Defining the zonotopes

$$Z^+ \coloneqq \varphi\Big(\sum_{i \in I^+} \max\{0, w_i^\top x\}\Big) = \sum_{i \in I^+} \mathrm{conv}(\{0, w_i\}),$$

$$Z^- \coloneqq \varphi\Big(\sum_{i \in I^-} \max\{0, w_i^\top x\}\Big) = \sum_{i \in I^-} \mathrm{conv}(\{0, w_i\}),$$

it holds that $g = \varphi^{-1}(Z^+) - \varphi^{-1}(Z^-)$. By definition of the support function, $Z^+ \subseteq Z^-$ implies $\varphi^{-1}(Z^+) \leq \varphi^{-1}(Z^-)$. Conversely, if $y_* \in Z^+ \setminus Z^-$, then there is a separating hyperplane $H = \{y \in \mathbb{R}^d : y^\top x = b\}$ such that $y_*^\top x > b$ and $y^\top x < b$ for all $y \in Z^-$. Hence, $g(x) > 0$ (see Figure 4 for an illustration). Since also any pair of zonotopes is of this form, 2-LAYER RELU POSITIVITY is equivalent to ZONOTOPE NON-CONTAINMENT. Thus, Theorem 4.2 implies the following theorem.

Theorem 5.1. ZONOTOPE NON-CONTAINMENT *is $W[1]$-hard with respect to d and not solvable in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

6 HARDNESS OF COMPUTING THE LIPSCHITZ CONSTANT

Jordan & Dimakis (2020) established the NP-hardness for approximating the L_p-Lipschitz constant for $p = 1$ and $p = \infty$ for 3-layer ReLU networks within a multiplicative factor of $\Omega(N^{1-\varepsilon})$ for every constant $\varepsilon > 0$, where N is the encoding size of the ReLU network. The NP-hardness result of Froese et al. (2025b) for 2-LAYER RELU POSITIVITY implies NP-hardness for approximating the L_p-Lipschitz constant for $p \in [0, \infty]$ within any multiplicative factor for 3-layer ReLU networks. We extend this by showing W[1]-hardness of the problem.

Corollary 6.1. *For all $p \in [0, \infty]$, approximating the L_p-Lipschitz constant of a 3-layer ReLU network by any multiplicative factor is W[1]-hard with respect to its input dimension d and cannot be done in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

Proof. Adding a hidden layer with a single ReLU neuron to the construction in the proof of Theorem 4.2 yields a 3-layer network which computes a function with a non-zero L_p-Lipschitz constant if and only if the original 2-layer network has a positive output. Hence, any multiplicative approximation could be used to decide 2-LAYER RELU POSITIVITY. □

Virmaux & Scaman (2018) established the NP-hardness of 2-LAYER RELU L_2-LIPSCHITZ CONSTANT. We extend the NP-hardness to $p \in (0, \infty]$ and show W[1]-hardness w.r.t. d.

Theorem 6.2. *For all $p \in (0, \infty]$, 2-LAYER RELU L_p-LIPSCHITZ CONSTANT is NP-hard, $W[1]$-hard with respect to d and not solvable in $\rho(d) \cdot N^{o(d)}$ time (where N is the input bit-length) for any function ρ assuming the ETH.*

Proof Sketch. First, we show that for any positively homogeneous CPWL function $f : \mathbb{R}^d \to \mathbb{R}$, we have $L_p(f) = \max_{\|x\|_p \le 1} |f(x)|$. The idea is now to scale all y coefficients of the function $g : \mathbb{R}^{k+1} \to \mathbb{R}$ computed by the homogenized network constructed in the proof of Proposition 3.1 by a sufficiently small amount ε to obtain the positively homogeneous CPWL function $h : \mathbb{R}^{k+1} \to \mathbb{R}$. Then, every $x^* \in \arg\max_{x \in \mathbb{R}^k} h(x, 1)$ has (sufficiently) small entries, as scaling the y coefficients is equivalent to scaling the spike and penalty functions. We then show that $L_p(h)$ is almost equal to $\mathcal{L} := \max_{x \in \mathbb{R}^k} h(x, 1)$, as we can scale down a maximizer $x^* \in \arg\max_{x \in \mathbb{R}^k} h(x, 1)$ with a y^* that is only slightly smaller than 1 to obtain a feasible point $y^*(x, 1)$ for $\max_{\|(x,y)\|_p \le 1} |h(x, y)|$ with value $|h(y^* \cdot x, y^*)| = y^* |h(x^*, 1)|$, which proves $\mathcal{L} \ge L_p(h) \ge \mathcal{L} \cdot y^*$ (so $L_p(h) \approx \mathcal{L}$). We conclude the proof by showing that the hardness of computing $\mathcal{L}$ transfers to computing $L_p(h)$. □

On the positive side, we show that for a special subclass of ReLU networks, computing the L_1- and L_∞-Lipschitz constant is tractable.

Input Convex Neural Networks. A ReLU network is *input-convex* (ICNN) if the weight matrices of all but the first layer have only nonnegative entries, resulting in a convex function $f(x) = \max\{a_1^\top x + b_1, \dots, a_k^\top x + b_k\}$. The L_p-Lipschitz constant of f is given by the maximum, taken over all linear regions C of f, of the L_p-Lipschitz constant of f restricted to C, where $f(x) = a_C^\top x + b_C$ for all $x \in C$. Using the well-known equality $L_p(g) = \max_{x \in \mathbb{R}^d} \|\nabla g(x)\|_q$ for smooth functions $g : \mathbb{R}^d \to \mathbb{R}$ (Jordan & Dimakis, 2020), we derive that the L_p-Lipschitz constant of f restricted to the region C is equal to $\|a_C\|_q$ and thus $L_p(f) = \max_{C \text{ linear region of } f} \|a_C\|_q = \max_i \|a_i\|_q$, where $\|\cdot\|_q$ is the *dual* norm of the L_p-norm. Note that the function f of the ICNN has the same L_p-Lipschitz constant as the function $g(x) = \max\{a_1^\top x, \dots, a_k^\top x\}$ computed by the same network where all biases are set to 0, which implies that we might assume without loss of generality that the network does not have biases and hence computes a function f that is convex and positively homogeneous.

Hertrich & Loho (2024) showed that there is a small *extended formulation* of $\operatorname{Newt}(f)$ for a function f computed by an ICNN without biases. More precisely, their proofs reveal that for a function $f : \mathbb{R}^d \to \mathbb{R}$ computed by an ICNN, there is a polytope $Q \subseteq \mathbb{R}^{d+m}$ and a projection $\pi : \mathbb{R}^{d+m} \to \mathbb{R}^d$ such that $\pi(Q) = \operatorname{Newt}(f)$ and the encoding size of (Q, π) is polynomial in the encoding size of f, where Q is given in half-space representation. Using this, we prove the following proposition.

Proposition 6.3. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an ICNN with encoding size N. Then $L_1(f)$ can be computed in $\operatorname{poly}(N)$ time and $L_\infty(f)$ can be computed in $O(2^d \operatorname{poly}(N))$ time.*

Proof. By the discussion above, we can assume without loss of generality that there are no biases and f is positively homogeneous. In this case, the definition of the support function implies that $L_p(f) = \max_{y \in \operatorname{Newt}(f)} \|y\|_q$. By Hertrich & Loho (2024), there exists a polytope Q and a projection π with $\operatorname{poly}(N)$ encoding size such that $L_p(f) = \max_{y \in Q} \|\pi(y)\|_q$. For $p = \infty$ and $p = 1$, this maximization can be reduced to finitely many LPs: Indeed,

$\max_{y \in Q} \|\pi(y)\|_\infty = \max_{c \in \{\pm e_1, \ldots, \pm e_d\}} \max_{y \in Q} c^\top \pi(y)$, which requires solving only $2d$ LPs, while $\max_{y \in Q} \|\pi(y)\|_1 = \max_{c \in \{\pm 1\}^d} \max_{y \in Q} c^\top \pi(y)$, which requires solving 2^d LPs. Since LPs can be solved in polynomial time, the statements follow. $\square$

7 Norm Maximization on Zonotopes

We close with a short section describing a connection between Lipschitz constants of neural networks and norm maximization on zonotopes. For 2-layer ICNNs $f\colon \mathbb{R}^d \to \mathbb{R}$, we can restrict ourselves without loss of generality to the case where all output weights are equal to 1. In this case, as shown by Froese et al. (2025a), the Newton polytope $\mathrm{Newt}(f)$ is a zonotope and computing $L_p(f)$ is equivalent to maximizing the dual norm of the L_p-norm over this zonotope.

Baburin & Pyatkin (2007) showed that maximizing the L_∞-norm on zonotopes is solvable in polynomial time and maximizing the L_1-norm on zonotopes is fixed-parameter tractable for d (our Proposition 6.3 generalizes these results). Note that Theorem 5.1 implies that maximizing a zonotope-norm over a zonotope is W[1]-hard with respect to the dimension d (since zonotope containment is equivalent to this problem (Kulmburg & Althoff, 2021)). For $p \in (1, \infty)$, however, it is an open question whether L_p-maximization on zonotopes is fixed-parameter tractable for d (Froese et al., 2025a). Shenmaier (2018) proved NP-hardness and inapproximability for $p \in [1, \infty)$ and showed a randomized (sampling based) $(1-\varepsilon)$-approximation with probability $1 - 1/\varepsilon$ in time $(1 + 2/\varepsilon)^d \operatorname{poly}(d, n)$ for every $\varepsilon \in (0,1)$ and an arbitrary norm. We show that known results from *subspace embedding* theory can also be used to obtain randomized approximations, which is an interesting application of these results. The worst-case running time, however, is worse, but in practice the actual running time might still be faster. Bozzai et al. (2023) observed that results for ℓ_1 subspace embeddings (Cohen & Peng, 2015) yield *zonotope order reductions*, that is, approximations of zonotopes with few generators. More precisely, the following can be derived.

Theorem 7.1. *There is a polynomial-time algorithm which, given a matrix $A \in \mathbb{R}^{d \times n}$ and $\varepsilon > 0$, outputs a matrix $A' \in \mathbb{R}^{d \times r}$ with $r \in \mathcal{O}(d \log d \varepsilon^{-2})$ such that with high probability*

$$(1+\varepsilon)^{-1} Z(A') \subseteq Z(A) \subseteq (1+\varepsilon) Z(A').$$

This order reduction yields a simple randomized approximation algorithm.

Theorem 7.2. *Let $\|\cdot\|$ be any norm on $\mathbb{R}^d$ (computable in time T). There is a randomized algorithm which, given a matrix $A \in \mathbb{R}^{d \times n}$ and $\varepsilon > 0$, outputs a value $\alpha \in \mathbb{R}$ in $\mathcal{O}((cd \log d/\varepsilon^2)^{d-1} \cdot T + \operatorname{poly}(n))$ time (for some constant $c > 0$) such that with high probability*

$$(1+\varepsilon)^{-1} \alpha \leq \max_{x \in Z(A)} \|x\| \leq (1+\varepsilon)\alpha.$$

Proof. Note that every norm is convex and convex functions attain their maximum on a polytope at a vertex. On input (A, ε), we run the algorithm from Theorem 7.1 to obtain a matrix A' with $r \in \mathcal{O}(d \log d \varepsilon^{-2})$ columns in polynomial time. The zonotope $Z(A')$ has at most $\mathcal{O}(r^{d-1})$ vertices (Zaslavsky, 1975), which can be enumerated in $\mathcal{O}(r^{d-1})$ time (Ferrez et al., 2005). We simply return the maximum $\|\cdot\|$-value α of these vertices. Then, with high probability, it holds $(1+\varepsilon)^{-1} Z(A') \subseteq Z(A) \subseteq (1+\varepsilon) Z(A')$, which implies

$$\max_{x \in (1+\varepsilon)^{-1} Z(A')} \|x\| = (1+\varepsilon)^{-1} \alpha \leq \max_{x \in Z(A)} \|x\| \leq (1+\varepsilon)\alpha = \max_{x \in (1+\varepsilon) Z(A')} \|x\|,$$

due to absolute homogeneity of norms. $\square$

8 Conclusion

We proved the strongest hardness results for various computational problems related to ReLU network verification known so far. Note that nearly all considered problems can be phrased in terms of maximizing a certain norm over a zonotope; a problem with numerous applications in other areas. Most importantly, our results imply that simple "brute-force" enumeration algorithms are basically best possible with respect to the dependency of the running time on the input dimension. Thus, we settled the parameterized complexity of a wide range of problems almost completely. Moreover, our results show that it does not help to assume that the network weights are sparse and small, since our constructions use only a constant number of (polynomially bounded) non-zero weights for each ReLU neuron. It is thus not easy to formulate a general guidance to circumvent this hardness in practice. One would have to make very specific assumptions on the network structure to ensure that the number of linear regions is small and easy to enumerate. It is not clear which assumptions would be natural here and whether networks trained on real-world data satisfy them. Alternatively, one might use techniques (possibly incorporated into the training process) that guarantee efficient verification or use special architectures (such as ICNNs). We also discussed some tractable cases for restricted subclasses of problems as well as a randomized FPT-approximation. Overall, our hardness results prove and justify that such techniques and the use of heuristics are indeed required in practice to achieve reasonable running times.

The most prominent open question is the fixed-parameter tractability of L_p-maximization on zonotopes for $p \in (1, \infty)$ when parameterized by d. Recall that this is equivalent to 2-LAYER RELU L_p-LIPSCHITZ CONSTANT with only positive output weights. As a first step, one might try to find a deterministic FPT-approximation for norm maximization on zonotopes (e.g., by derandomizing the subspace embedding approach). Also, the complexity of computing $L_0(f)$ for 2-layer ReLU networks is open. In general, it is an interesting question whether the constants in the exponents of the running times can be improved, e.g., is ZONOTOPE CONTAINMENT solvable in $\mathcal{O}(n^{cd})$ time for any $c < 1$?

REFERENCES

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

Maximilian Baader, Mark Niklas Mueller, Yuhao Mao, and Martin Vechev. Expressivity of reLU-networks under convex relaxations. In *International Conference on Learning Representations*, 2024.

A. E. Baburin and A. V. Pyatkin. Polynomial algorithms for solving the vector sum problem. *Journal of Applied and Industrial Mathematics*, 1(3):268–272, 2007.

Hans L. Bodlaender, Peter Gritzmann, Victor Klee, and Jan van Leeuwen. Computational complexity of norm-maximization. *Combinatorica*, 10(2):203–225, 1990.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

Rainie Bozzai, Victor Reis, and Thomas Rothvoss. The vector balancing constant for zonotopes. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1292–1300. IEEE, 2023.

Cornelius Brand, Robert Ganian, and Mathis Rocton. New complexity-theoretic frontiers of tractability for neural network training. *Advances in Neural Information Processing Systems*, 36, 2023.

Michael B. Cohen and Richard Peng. ℓ_p row sampling by lewis weights. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing (STOC '15)*, pp. 183–192. ACM, 2015.

Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15(2):341–363, 1986.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Claudio Ferrari, Mark Niklas Müller, Nikola Jovanović, and Martin Vechev. Complete verification via multi-neuron relaxation guided branch-and-bound. In *International Conference on Learning Representations*, 2022.

J.-A. Ferrez, Komei Fukuda, and Thomas M. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operations Research*, 166(1):35–50, 2005.

Vincent Froese and Christoph Hertrich. Training neural networks is np-hard in fixed dimension. *Advances in Neural Information Processing Systems*, 36, 2023.

Vincent Froese, Christoph Hertrich, and Rolf Niedermeier. The computational complexity of ReLU network training parameterized by data dimensionality. *Journal of Artificial Intelligence Research*, 74:1775–1790, 2022.

Vincent Froese, Moritz Grillo, Christoph Hertrich, and Martin Skutella. Open problem: Fixed-parameter tractability of zonotope problems. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 6210–6214. PMLR, 2025a.

Vincent Froese, Moritz Grillo, and Martin Skutella. Complexity of injectivity and verification of relu neural networks (extended abstract). In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 2188–2189. PMLR, 2025b.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018.

Felix Gruber and Matthias Althoff. Computing safe sets of linear sampled-data systems. *IEEE Control Systems Letters*, 5(2):385–390, 2020.

Felix Gruber and Matthias Althoff. Scalable robust output feedback mpc of linear sampled-data systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2563–2570. IEEE, 2021.

Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. *Advances in Neural Information Processing Systems*, 32, 2019.

Christoph Hertrich and Georg Loho. Neural networks and (virtual) extended formulations. *CoRR*, abs/2411.03006, 2024.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Matt Jordan, Jonathan Hayase, Alex Dimakis, and Sewoong Oh. Zonotope domains for lagrangian neural network verification. *Advances in Neural Information Processing Systems*, 35, 2022.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: a calculus for reasoning about deep neural networks. *Formal Methods in System Design*, 60(1): 87–116, 2022.

Panagiotis Kouvaros and Alessio Lomuscio. Towards scalable complete verification of ReLU neural networks via dependency-based branching. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI '21)*, pp. 2643–2650, 2021.

Adrian Kulmburg and Matthias Althoff. On the co-NP-completeness of the zonotope containment problem. *European Journal of Control*, 62:84–91, 2021.

Adrian Kulmburg, Lukas Schafer, and Matthias Althoff. Approximability of the containment problem for zonotopes and ellipsotopes. *IEEE Transactions on Automatic Control*, 2025.

Yuhao Mao, Yani Zhang, and Martin Vechev. On the expressiveness of multi-neuron convex relaxations. *arXiv preprint arXiv:2410.06816*, 2024.

Abdul Majid Mian and S Chowla. On the b2 sequences of sidon. *Proc. Nat. Acad. Sci. India. Sect. A*, 14:3–4, 1944.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

Kevin O'Bryant. A complete annotated bibliography of work related to sidon sequences. *The Electronic Journal of Combinatorics [electronic only]*, 11, 2004.

Ansgar Rössig and Milena Petkovic. Advances in verification of ReLU neural networks. *Journal of Global Optimization*, 81(1):109–152, 2021.

Sadra Sadraddini and Russ Tedrake. Linear encodings for polytope containment problems. In *2019 IEEE 58th conference on decision and control (CDC)*, pp. 4367–4372. IEEE, 2019.

Marco Sälzer and Martin Lange. Reachability in simple neural networks. *Fundamenta Informaticae*, 189(3-4):241–259, 2022.

A. Schrijver. *Theory of Linear and Integer programming*. Wiley-Interscience, 1986.

Vladimir Shenmaier. Complexity and approximation of finding the longest vector sum. *Computational Mathematics and Mathematical Physics*, 58(6):850–857, 2018.

Vladimir Shenmaier. Complexity and algorithms for finding a subset of vectors with the longest sum. *Theoretical Computer Science*, 818:60–73, 2020.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL), 2019.

Moritz Stargalla, Christoph Hertrich, and Daniel Reichman. The computational complexity of counting linear regions in ReLU neural networks. *arXiv preprint arXiv:2505.16716*, 2025. Accepted for publication at NeurIPS 2025.

Alfred Stöhr. Gelöste und ungelöste fragen über basen der natürlichen zahlenreihe. ii. *Journal für die reine und angewandte Mathematik*, 194:111–140, 1955.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, 2018a.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018b.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

Liren Yang, Hang Zhang, Jean-Baptiste Jeannin, and Necmiye Ozay. Efficient backward reachability using the minkowski difference of constrained zonotopes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):3969–3980, 2022.

Thomas Zaslavsky. Facing up to arrangements: face-count formulas for partitions of space by hyperplanes. *Memoirs of the American Mathematical Society*, 1(154), 1975.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. General cutting planes for bound-propagation-based neural network verification. *Advances in Neural Information Processing Systems*, 35, 2022.

A Additional Material

A.1 Additional Preliminaries

Geometry of ReLU Networks. We repeat basic definitions from polyhedral geometry, see Schrijver (1986) for more details. A *polyhedron* P is the intersection of finitely many closed halfspaces. A *polyhedral cone* $C \subseteq \mathbb{R}^d$ is a polyhedron such that $\lambda u + \mu v \in C$ for every $u, v \in C$ and $\lambda, \mu \in \mathbb{R}_{\geq 0}$. A cone is *pointed* if it does not contain a straight line. A *ray* ρ is a one-dimensional pointed cone; a vector r is a *ray generator* of ρ if $\rho = \{\lambda r : \lambda \geq 0\}$. A *polyhedral complex* $\mathcal{P}$ is a finite collection of polyhedra such that $\emptyset \in \mathcal{P}$, if $P \in \mathcal{P}$, then all faces of P are in $\mathcal{P}$, and if $P, P' \in \mathcal{P}$, then $P \cap P'$ is a face of P and P'. A *cell* is a full-dimensional element of a polyhedral complex. A *hyperplane arrangement* $\mathcal{H}$ is a finite collection of hyperplanes in $\mathbb{R}^d$.

A *ReLU network* computing the CWPL function $f\colon \mathbb{R}^d \to \mathbb{R}$ with

$$f(x) := W_\ell \cdot (\phi_{W_{\ell-1}, b_{\ell-1}} \circ \cdots \circ \phi_{W_1, b_1})(x) + b_\ell,$$

is affine linear on each cell of an associated polyhedral complex Σ_f (the one induced by the closure of the full-dimensional activation regions (Hanin & Rolnick, 2019) of the network). In particular, the encoding size of every polyhedron in this complex is polynomially bounded in the encoding size of the neural network (Froese et al., 2025b). It is well known that for 2-layer ReLU networks, Σ_f corresponds to a hyperplane arrangement (Montúfar et al., 2014).

Hyperplane Arrangements. Any hyperplane arrangement with n hyperplanes in d dimensions has at most $\mathcal{O}(n^d)$ many cells (Zaslavsky, 1975), and it is possible to enumerate them in $\mathcal{O}(n^d)$ time (Edelsbrunner et al., 1986). All zero- and one-dimensional faces of a hyperplane arrangement can be enumerated in $\mathcal{O}(n^d \cdot \mathrm{poly}(N))$ and $\mathcal{O}(n^{d-1} \cdot \mathrm{poly}(N))$ time, respectively, where N is the encoding size of the hyperplane arrangement, since they arise from an intersection of d and $d-1$ hyperplanes. An ℓ-layer ReLU network partitions $\mathbb{R}^d$ into at most $\mathcal{O}(n^{(\ell-1)d})$ cells, where n is the width of the network. We can enumerate these cells in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time, where N is the encoding size of the ReLU network (the first hidden layer gives a hyperplane arrangement; with every subsequent hidden layer, a cell is intersected with at most n hyperplanes, partitioning the cell into at most n^d cells).

A.2 Additional Figures

Figure 5 illustrates how the sum of the spike function $s_{r,l}$ and penalty functions p_r, p_l in the proof of Proposition 3.1 behaves for a fixed color pair.

B Proofs

B.1 Proof of Theorem 2.1

Proof. **Zonotope (Non-)Containment.** It is well-known that a zonotope $Z \subset \mathbb{R}^d$ with n generators has $\mathcal{O}(n^{d-1})$ vertices which can be enumerated in $\mathcal{O}(n^{d-1})$ time (Ferrez et al., 2005). Note that Z is contained in another zonotope Z' if and only if all vertices of Z are contained in Z'. Hence, by enumerating the vertices of Z and checking containment in Z' (e.g. by solving a linear program), we obtain an $\mathcal{O}(n^{d-1} \cdot \mathrm{poly}(N))$-time algorithm.

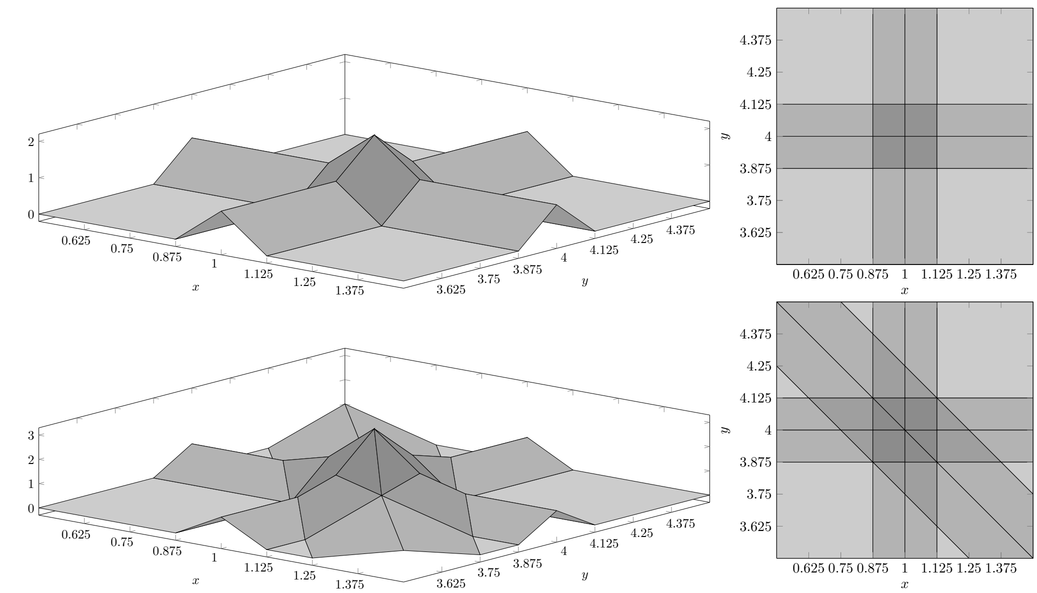

Figure 5: Illustration of the area around node values for a fixed color pair for non-adjacent and adjacent nodes. The values are based on the example given in Figure 1. For non-adjacent nodes, only the two penalty functions intersect, while for adjacent nodes, the penalty functions intersect also with the spike function.

L_p-MAX ON ZONOTOPES. We can enumerate in $\mathcal{O}(n^{d-1} \cdot \mathrm{poly}(N))$ time all vertices of the zonotope and thus obtain an $\mathcal{O}(n^{d-1} \cdot \mathrm{poly}(N) \cdot T)$ time algorithm, where T denotes the time to evaluate the L_p-norm of a vector in $\mathbb{R}^d$.

$\square$

B.2 PROOF OF THEOREM 2.2

Before going into the proof of Theorem 2.2, we first prove that network verification for ℓ-layer ReLU networks over polyhedra is solvable in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time.

Lemma B.1. *Let $f\colon \mathbb{R}^d \to \mathbb{R}^m$ be an ℓ-layer ReLU network and let $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^m$ be polyhedra given in halfspace representation. Then, we can decide whether $f(P) \subseteq Q$ holds in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time (where n is the network width and N the combined encoding size of the network and the polyhedra).*

Proof. Let $Q = \{x \in \mathbb{R}^m : v_i^\top x \leq u_i, i \in [k]\}$. Then, we have $f(P) \subseteq Q$ if and only if for every cell $C \in \Sigma_f$ and every constraint $i \in [k]$, we have

$$u_i \geq \max_{x \in C \cap P} v_i^\top f_C(x) = \max_{x \in C \cap P} v_i^\top (A_C^\top x + b_C) = v_i^\top b_C + \max_{x \in C \cap P} (A_C^\top v_i)^\top x,$$

where f_C is the affine linear function of f restricted to C, that is, $f(x) = f_C(x) \coloneqq A_C^\top x + b_C$ for all $x \in C$. Note that the above condition can be verified by solving a linear program whose encoding size is polynomially bounded in N. Since linear programs can be solved in polynomial time and cells can be enumerated in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time, it follows that we can check whether $f(P) \subseteq Q$ holds in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time. $\square$

Proof of Theorem 2.2. Let $f\colon \mathbb{R}^d \to \mathbb{R}$ with $f(x) = W_\ell \cdot (\phi_{W_{\ell-1}, b_{\ell-1}} \circ \cdots \circ \phi_{W_1, b_1})(x) + b_\ell$ be the function computed by an ℓ-layer ReLU network and let Σ_f be the corresponding polyhedral complex. Further, let n denote the width and N the encoding size of the network and let $f'\colon \mathbb{R}^d \to \mathbb{R}$ with $f'(x) \coloneqq W_\ell \cdot (\phi_{W_{\ell-1}, \mathbf{0}} \circ \cdots \circ \phi_{W_1, \mathbf{0}})(x)$ be the function of the ReLU network without biases.

Computing the Maximum over a Polyhedron P. Here, we assume that the maximum $\max_{x\in P} f(x)$ exists and N denotes the combined encoding size of the network and the polyhedron P. We can compute this maximum by enumerating cells of Σ_f and solving the linear program $\max\limits_{x\in C\cap P} f_C(x)$ for each cell $C \in \Sigma_f$, where f_C is the affine linear function of f restricted to C, that is, $f(x) = f_C(x) := a_C^\top x + b_C$ for all $x \in C$. Note that the encoding size of the linear program is polynomially bounded in N. Since linear programs can be solved in polynomial time and cells can be enumerated in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time, it follows that $\max_{x\in P} f(x) = \max\limits_{C \text{ cell of } \Sigma_f} \max\limits_{x\in C\cap P} f_C(x)$ can be computed in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time.

ℓ-LAYER ReLU POSITIVITY. Follows from applying Lemma B.1 to $P = \mathbb{R}^d$ and $Q = (-\infty, 0]$, since there being a point $x \in \mathbb{R}^d$ with $f(x) > 0$ is equivalent to $f(\mathbb{R}^d) \not\subseteq (-\infty, 0]$.

ℓ-LAYER ReLU SURJECTIVITY. Froese et al. (2025b, Lemma 14) show that for surjectivity, f is surjective if and only if f' is surjective, which is equivalent to there being two points $r^+, r^- \in \mathbb{R}^d$ with $f'(r^-) < 0 < f'(r^+)$. We can check this in $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N))$ time by applying the algorithm for ℓ-LAYER ReLU POSITIVITY to f' and $-f'$.

ℓ-LAYER ReLU L_p-LIPSCHITZ CONSTANT. Note that the L_p-Lipschitz constant of f is equal to the maximum L_q-norm value (where L_q is the dual norm of the L_p-norm, so $1/p + 1/q = 1$) of any gradient of the linear function that arises by restricting f to a cell of Σ_f. Thus, by enumerating all cells of Σ_f, we obtain an $\mathcal{O}(n^{(\ell-1)d} \cdot \mathrm{poly}(N) \cdot T)$ time algorithm, where T denotes the time to evaluate the L_q-norm of a vector in $\mathbb{R}^d$.

ℓ-LAYER ReLU ZERO FUNCTION CHECK. Follows from applying Lemma B.1 to $P = \mathbb{R}^d$ and $Q = \{0\}$. □

B.3 PROOF OF PROPOSITION 3.1

Before going into detail, we first introduce a useful definition and prove a preliminary result.

Definition B.2. A *Sidon set* is a set of positive integers $A = \{a_1, \dots, a_m\}$ where the sums $a_i + a_j$ with $i \le j$ are all different.

For a survey on Sidon sets, we refer to (O'Bryant, 2004). The greedy Sidon set, introduced by Mian & Chowla (1944), is recursively constructed as follows: take $a_1 = 1$, and for $n > 1$, let a_n be the smallest nonnegative integer such that $\{a_1, \dots, a_n\}$ is a Sidon set (see A005282). Stöhr (1955) noted that $a_n \in \mathcal{O}(n^3)$ holds. We note that the greedy Sidon set of size n can be computed in $n^{\mathcal{O}(1)}$ time. We use the following result.

Lemma B.3. *Let A be a Sidon set of size n, and let $W_1, \dots, W_k$ be a partition of A into disjoint subsets. Then, for every pair $i, j \in \binom{[k]}{2}$, the sums $a + b$ with $a \in W_i, b \in W_j$ are all different.*

Proof. Suppose that there are two pairs $(a, b) \neq (c, d) \in W_i \times W_j$ with $a + b = c + d$. Then, there exist elements $a_i, a_j, a_r, a_l \in A$ such that $a_i = a$, $a_j = b$, $a_r = c$, $a_l = d$ with $\{i, j\} \neq \{r, l\}$ and $a_i + a_j = a_r + a_l$, which contradicts the fact that A is a Sidon set. □

In other words, given an element $w \in W_i + W_j$, there is exactly one pair $(w_i, w_j) \in W_i \times W_j$ with $w = w_i + w_j$.

Proof of Proposition 3.1. Let $(G = (V = V_1 \dot\cup \dots \dot\cup V_k, E), k)$ be an instance of MULTICOLORED CLIQUE, where $V_c = \{v_{c,1}, \dots, v_{c,n_c}\}$ for $c \in [k]$ and $E = \bigcup_{(r,l)\in\binom{[k]}{2}} E_{r,l}$, where $E_{r,l}$ denotes the set of edges whose nodes have color r and l. Further, let A be the greedy Sidon set of size $|V|$ and let $W_1, \dots, W_k$ be a partition of A into k disjoint subsets such that $|W_i| = n_i$ holds. Note that this allows us to assign each node $v_{c,i}$ to a unique element $\omega_{c,i}$ of A, namely the i-th element of W_c. For every edge $\{v_{r,i}, v_{l,j}\}$, we define the constant $\omega_{r,i,l,j} := \omega_{r,i} + \omega_{l,i}$. Since A is a Sidon set, the value $\omega_{r,i,l,j}$ uniquely determines the edge $\{v_{r,i}, v_{l,j}\}$. We construct a ReLU network with k input variables $x_1, \dots, x_k$ and $3(|V| + |E|)$ hidden neurons as follows.

For every color $c \in [k]$, we introduce a *node selection* gadget which ensures that the input value x_c encodes a node in V_c. To this end, we create a "penalty function" $p_c \colon \mathbb{R} \to [0, 1]$ (see Figure 2) that has n_c narrow spikes around the value $\omega_{c,i}$ (that is, it goes up from 0 to 1 and down to 0 again) for each $v_{c,i} \in V_c$ and is zero everywhere else:

$$p_c(x) := \begin{cases} 8(x - \omega_{c,i} + \frac{1}{8}), & \text{if } x \in [\omega_{c,i} - \frac{1}{8}, \omega_{c,i}], i \in [n_c] \\ 1 - 8(x - \omega_{c,i}), & \text{if } x \in (\omega_{c,i}, \omega_{c,i} + \frac{1}{8}], i \in [n_c] \\ 0, & \text{if } x \notin \bigcup_{i \in V_c} [\omega_{c,i} - \frac{1}{8}, \omega_{c,i} + \frac{1}{8}] \end{cases}.$$

The penalty function p_c can be implemented with $3n_c$ hidden neurons:

$$p_c(x) = \sum_{i \in [n_c]} (\max(0, 8(x - \omega_{c,i} + \frac{1}{8})) - \max(0, 16(x - \omega_{c,i})) + \max(0, 8(x - \omega_{c,i} - \frac{1}{8})).$$

Next, we introduce an *edge verification* gadget which verifies that each pair of nodes selected by the node selection gadgets is connected by an edge. For every pair of colors $(r, l) \in \binom{[k]}{2}$, we define a "spike function" $s_{r,l} \colon \mathbb{R}^2 \to [0, 1]$ (see Figure 1) that is zero everywhere except for a set of $|E_{r,l}|$ parallel stripes in which $s_{r,l}$ forms a spike.

$$s_{r,l}(x, y) := \begin{cases} 4(x + y - \omega_{r,i,l,j} - \frac{1}{4}), & \text{if } x + y \in [\omega_{r,i,l,j} - \frac{1}{4}, \omega_{r,i,l,j}], \{v_{r,i}, v_{l,j}\} \in E_{r,l} \\ 1 - 4(x + y - \omega_{r,i,l,j}), & \text{if } x + y \in (\omega_{r,i,l,j}, \omega_{r,i,l,j} + \frac{1}{4}], \{v_{r,i}, v_{l,j}\} \in E_{r,l} \\ 0, & \text{if } x + y \notin \bigcup_{\{v_{r,i}, v_{l,j}\} \in E_{r,l}} [\omega_{r,i,l,j} - \frac{1}{4}, \omega_{r,i,l,j} + \frac{1}{4}] \end{cases}.$$

Note that Lemma B.3 implies that $s_{r,l}(x, y) \leq 1$ holds for any input $(x, y) \in \mathbb{R}^2$, as the sums $\omega_{r,i} + \omega_{l,i} = \omega_{r,i,l,j}$ for $\{v_{r,i}, v_{l,j}\} \in E_{r,l}$ are all integral and different. Thus, the spike functions attain value 1 if and only if its inputs correspond to two nodes that share an edge. The spike function can be implemented with $3|E_{r,l}|$ hidden neurons:

$$\begin{aligned} s_{r,l}(x, y) = \sum_{\{v_{r,i}, v_{l,j}\} \in E_{r,l}} (&\max(0, 4(x + y - \omega_{r,i,l,j} + \frac{1}{4})) - \max(0, 8(x + y - \omega_{r,i,l,j})) \\ &+ \max(0, 4(x + y - \omega_{r,i,l,j} - \frac{1}{4}))). \end{aligned}$$

By computing all penalty and spike functions in parallel and summing them up at the output neuron, we obtain a ReLU network that computes the function $f \colon \mathbb{R}^k \to \mathbb{R}$ with

$$f(x_1, \ldots, x_k) = \sum_{(r,l) \in \binom{[k]}{2}} s_{r,l}(x_r, x_l) + \sum_{c \in [k]} p_c(x_c).$$

Since every spike and penalty function is lower bounded by 0 and upper bounded by 1, it follows that f is lower bounded by 0 and upper bounded by $k + \binom{k}{2}$.

First, we show that the existence of a k-colored clique in G implies $\max_{x \in \mathbb{R}^k} f(x) = k + \binom{k}{2}$. Suppose that $\{v_{1,a_1}, \ldots, v_{k,a_k}\} \subset V$ forms a k-colored clique in G. Then, we claim that the point $x^* = (\omega_{1,a_1}, \ldots, \omega_{k,a_k})$ is a point with $f(x^*) = k + \binom{k}{2}$. First, note that $p_c(x^*_c) = 1$ holds for all $c \in [k]$. Further, for each pair of colors $(r, l) \in \binom{[k]}{2}$, $s_{r,l}(x^*_r, x^*_l) = 1$ holds, since $\{v_{r,a_r}, v_{l,a_l}\}$ is an edge in E_{rl}. Thus, we have $f(x^*) = k + \binom{k}{2}$.

Now, we show that if there is a point $x^* \in \mathbb{R}^k$ with $f(x^*) > k + \binom{k}{2} - 1$, then G has a k-colored clique. Suppose that $x^* \in \mathbb{R}^k$ is such a point. For this to be the case, the output of all spike and penalty functions must be strictly greater than zero, that is, we have $p_c(x^*_c) > 0$ for every $c \in [k]$ and $s_{r,l}(x^*_r, x^*_l) > 0$ for every pair $(r, l) \in \binom{[k]}{2}$, since otherwise $f(x^*) \leq k + \binom{k}{2} - 1$ holds. For every $c \in [k]$, $p_c(x^*_c) > 0$ implies by definition that there is exactly one element $a_c \in V_c$ with $x^*_c \in (\omega_{c,a_c} - \frac{1}{8}, \omega_{c,a_c} + \frac{1}{8})$. In other words, the input x^*_c corresponds to the node v_{c,a_c}. We now claim that $\{v_{1,a_1}, \ldots, v_{k,a_k}\}$ forms a k-colored clique in G. To see this, observe that for every pair $(r, l) \in \binom{[k]}{2}$, $s_{r,l}(x^*_r, x^*_l) > 0$ together with $x^*_r + x^*_l \in (\omega_{r,a_r,l,a_l} - \frac{1}{4}, \omega_{r,a_r,l,a_l} + \frac{1}{4})$ implies by definition of $s_{r,l}$ that $\{v_{r,a_r}, v_{l,a_l}\}$ is an edge, which proves that $\{v_{1,a_1}, \ldots, v_{k,a_k}\}$ is indeed a k-colored clique. $\square$

B.4 Proof of Theorem 4.2

Before proving Theorem 4.2, we first prove an auxiliary lemma.

Lemma B.4. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$, $\quad f(x) = \sum_{i=1}^n c_i \max\{0, a_i^\top x + b_i\} + B$ be the function that is computed by a 2-layer ReLU network, where a_i, b_i, c_i, B are the weights and biases of this network, and let $h\colon \mathbb{R}^{d+1} \to \mathbb{R}$ be the function computed by the homogenization of this network. Then, we have $h(x, 1) = h(-x, -1)$ if and only if $\sum_{i=1}^n c_i(a_i^\top x + b_i) = 0$.*

Proof. We have $h(x, y) = \sum_{i=1}^n c_i \max\{0, a_i^\top x + b_i y\} + B|y|$ and thus

$$h(x, 1) - h(-x, -1) = \sum_{i=1}^n c_i(\max\{0, a_i^\top x + b_i\} - \max\{0, -(a_i^\top x + b_i)\}) = \sum_{i=1}^n c_i(a_i^\top x + b_i).$$

□

Proof of Theorem 4.2. We give a parameterized reduction from Multicolored Clique. Let $(G = (V = V_1 \dot\cup \ldots \dot\cup V_k, E), k)$ be an instance of Multicolored Clique, and let $f\colon \mathbb{R}^k \to \mathbb{R}$ be the function of the network constructed in the proof of Proposition 3.1. Next, we modify the network by setting the bias of the output node to $1 - k - \binom{k}{2}$. Let $g\colon \mathbb{R}^k \to \mathbb{R}$ be the function of this modified network and let $h\colon \mathbb{R}^{k+1} \to \mathbb{R}$ be the function computed by the homogenization of this modified network. By construction, we have $h(x, 1) = g(x) = f(x) + 1 - k - \binom{k}{2}$ for every $x \in \mathbb{R}^k$. Note that $h(-x, -1) = h(x, 1)$ holds for every $x \in \mathbb{R}^k$, which follows directly from the definition of f and Lemma B.4. Since the underlying network has no biases, the function g computed by the network is positively homogeneous and thus $h(\lambda x, \lambda y) = \lambda h(x, y)$ holds for every $\lambda \geq 0$.

By Proposition 3.1, G has a k-colored clique if and only if g has a *positive point*, since $\max_{x \in \mathbb{R}^k} g(x) = 1$ holds if G has a k-colored clique and $\max_{x \in \mathbb{R}^k} g(x) \leq 0$ otherwise. To finish the proof, observe that h has a positive point if and only if g has a positive point. If g has a positive point x^*, then h also has a positive point $(x^*, 1)$. On the other hand, if h has a positive point (x^+, y^+), then by positive homogeneity $\operatorname{sgn}(y^+) \cdot \frac{x^+}{|y^+|}$ is a positive point for g, since $0 < \frac{1}{|y^+|} h(x^+, y^+) = h(\frac{x^+}{|y^+|}, \operatorname{sgn}(y^+)) = g(\operatorname{sgn}(y^+) \cdot \frac{x^+}{|y^+|})$. Note that $y^+ = 0$ is not possible, since $h(x, 0) = 0$ for every $x \in \mathbb{R}^k$ by construction, as deleting biases leads to the cancellation of all terms in the spike and penalty functions. □

B.5 Proof of Corollary 4.3

Proof. We give a parameterized reduction from Multicolored Clique to approximating the maximum of a 2-layer ReLU network. Let $(G = (V = V_1 \dot\cup \ldots \dot\cup V_k, E), k)$ be an instance of Multicolored Clique, let $f\colon \mathbb{R}^k \to \mathbb{R}$ be the function of the network constructed in the proof of Proposition 3.1 and let $g\colon \mathbb{R}^k \to \mathbb{R}$ be the function of the same network with an additional bias of $1 - k - \binom{k}{2}$ at the output node, that is, $g(x) = f(x) + 1 - k - \binom{k}{2}$ holds for every $x \in \mathbb{R}^k$. With Proposition 3.1, it follows that we have $\max_{x \in \mathbb{R}^k} g(x) = 1$ if and only if G has a k-colored clique and $\max_{x \in \mathbb{R}^k} g(x) \leq 0$ otherwise. Thus, approximating the maximum of this network within any multiplicative factor over the polytope $P = [0, n^3]^d$ would allow us to distinguish between the two cases, which implies the theorem. □

B.6 Proof of Theorem 5.1

Proof. Follows from Theorem 4.2 and the equivalence to 2-Layer ReLU Positivity without biases (Froese et al., 2025b, Proposition 18). □

B.7 Proof of Theorem 6.2

Proof. We give a parameterized reduction (which is also a polynomial reduction) from Multicolored Clique to 2-Layer ReLU L_p-Lipschitz Constant. We first discuss the case $p \in [1, \infty]$ and later discuss which modifications are necessary to extend the proof to $p \in (0, 1)$.

Let $(G = (V = V_1 \dot\cup \ldots \dot\cup V_k, E), k)$ be an instance of MULTICOLORED CLIQUE. Further, let $g\colon \mathbb{R}^{k+1} \to \mathbb{R}$ be the function computed by the homogenized network constructed in the proof of Proposition 3.1.

Note that for any positively homogeneous CPWL function $f : \mathbb{R}^d \to \mathbb{R}$, the L_p-Lipschitz constant can be rewritten to $L_p(f) = \max_{\|x\|_p \le 1} |f(x)|$, which follows from the fact that $L_p(f)$ is the maximum L_p-Lipschitz constant of f restricted to any of the full-dimensional cones $C \in \Sigma_f$, where $f(x) = a_C^\top x$ for all $x \in C$ (f is linear on C) and the L_p-Lipschitz constant of the linear function in this cell is equal to $\max_{\|x\|_p \le 1} |a_c^\top x|$ (Jordan & Dimakis, 2020).

We now scale all y coefficients of g by $\varepsilon := \frac{1}{2k \cdot a_n \cdot (k + \binom{k}{2})}$, where $a_n \in \mathcal{O}(n^3)$ is the maximum element of the greedy Sidon set of size n, and obtain the modified positively homogeneous CPWL function $h : \mathbb{R}^{k+1} \to \mathbb{R}$. Now, every maximizer x^* of $\max_{x \in \mathbb{R}^k} h(x, 1)$ satisfies $|x_i^*| \le a_n \cdot \varepsilon$, since every maximizer x' of $\max_{x \in \mathbb{R}^k} g(x, 1)$ previously satisfied $|x_i'| \le a_n$. This follows from the fact that scaling the y coefficients is equivalent to scaling the spike and penalty functions in the reduction. Now, we define

$$\mathcal{L} := \max_{x \in \mathbb{R}^k} h(x, 1)$$

and claim the following:

$$\mathcal{L} \ge L_p(h) \ge \mathcal{L}(1 - k \cdot a_n \cdot \varepsilon).$$

The inequality $\mathcal{L} \ge L_p(h)$ follows from the fact that if $(x^*, y^*) \in \arg\max_{\|(x,y)\|_p \le 1} h(x, y)$, then $|y^*| \le 1$ and

$$L_p(h) = h(x^*, y^*) = |y^*| \cdot h(\frac{x^*}{|y^*|}, \operatorname{sgn}(y^*)) \le h(\frac{x^*}{|y^*|}, \operatorname{sgn}(y^*)) \le \max_{x \in \mathbb{R}^k} h(x, 1) = \mathcal{L}$$

holds. The second inequality follows from the fact that if x^* is a maximizer of $\max_{x \in \mathbb{R}^k} h(x, 1)$, then $|x_i^*| \le a_n \cdot \varepsilon$ and thus

$$\|(1 - k \cdot a_n \cdot \varepsilon) \cdot (x^*, 1)\|_p \le \|(1 - k \cdot a_n \cdot \varepsilon) \cdot (x^*, 1)\|_1 \le (1 - k \cdot a_n \cdot \varepsilon) + \sum_{i=1}^{k} |x_i^*| \le 1$$

holds, which makes $(1 - k \cdot a_n \cdot \varepsilon) \cdot (x^*, 1)$ a feasible point for $\max_{\|(x,y)\|_p \le 1} h(x, y)$.

Given this estimation, we now make a case distinction: if G has a k-colored clique, then $\mathcal{L} = (k + \binom{k}{2}) \cdot \varepsilon$ and

$$L_p(h) \ge (1 - k \cdot a_n \cdot \varepsilon) \cdot (k + \binom{k}{2}) \cdot \varepsilon = (1 - \frac{1}{2(k + \binom{k}{2})}) \cdot (k + \binom{k}{2}) \cdot \varepsilon = (k + \binom{k}{2} - \frac{1}{2}) \cdot \varepsilon.$$

On the other hand, if G does not have a k-colored clique, then

$$L_p(h) \le \mathcal{L} \le (k + \binom{k}{2} - 1) \cdot \varepsilon.$$

Therefore, we have a separation of the L_p-Lipschitz constant $L_p(h)$ depending on whether G has a k-colored clique or not. With this, the 2-LAYER RELU L_p-LIPSCHITZ CONSTANT instance consisting of $L = (k + \binom{k}{2} - \frac{1}{2}) \cdot \varepsilon$ and the underlying network of h is a yes-instance if and only if G is a yes-instance of MULTICOLORED CLIQUE, which concludes the proof.

For every $p \in (0, 1) \cap \mathbb{Q}$, we can scale the network with $\varepsilon := \frac{1}{a_n \cdot k^N} \cdot \left(p \left(1 - \frac{k + \binom{k}{2} - \frac{1}{2}}{k + \binom{k}{2}} \right) \right)^N$, where $N = \lceil 1/p \rceil$. Note that since p is a fixed rational constant, ε is also rational and still polynomial in the input size. We estimate

$$\begin{aligned} \varepsilon = \frac{1}{a_n \cdot k^N} \cdot \left(p \left(1 - \frac{k + \binom{k}{2} - \frac{1}{2}}{k + \binom{k}{2}} \right) \right)^N &\le \frac{1}{a_n \cdot k^N} \cdot \left(1 - \left(\frac{k + \binom{k}{2} - \frac{1}{2}}{k + \binom{k}{2}} \right)^p \right)^N \\ &\le \frac{1}{a_n \cdot k^{1/p}} \cdot \left(1 - \left(\frac{k + \binom{k}{2} - \frac{1}{2}}{k + \binom{k}{2}} \right)^p \right)^{1/p} =: \varepsilon^*. \end{aligned}$$

Next, we can estimate $\mathcal{L} \geq L_p(h) \geq \mathcal{L}(1 - k \cdot (a_n \cdot \varepsilon)^p)^{1/p}$, where the second inequality follows from the fact that if x^* is a maximizer of $\max_{x \in \mathbb{R}^k} h(x, 1)$, then $|x_i^*| \leq a_n \cdot \varepsilon$ and thus

$$\|(1 - k \cdot (a_n \cdot \varepsilon)^p)^{1/p} \cdot (x^*, 1)\|_p = \left(1 - k \cdot (a_n \cdot \varepsilon)^p + (1 - k \cdot (a_n \cdot \varepsilon)^p) \sum_{i=1}^{k} |x_i^*|^p\right)^{1/p}$$
$$\leq (1 - k \cdot (a_n \cdot \varepsilon)^p + (1 - k \cdot (a_n \cdot \varepsilon)^p) \cdot k \cdot a_n \cdot \varepsilon)^{1/p} \leq 1$$

holds, which makes $(1 - k \cdot (a_n \cdot \varepsilon)^p)^{1/p} \cdot (x^*, 1)$ a feasible point for $\max_{\|(x,y)\|_p \leq 1} h(x, y)$.

We then proceed with the estimation for the case where G has a k-colored clique with

$$\begin{aligned} L_p(h) &\geq (1 - k \cdot (a_n \cdot \varepsilon)^p)^{1/p} \cdot (k + \binom{k}{2}) \cdot \varepsilon \\ &\geq (1 - k \cdot (a_n \cdot \varepsilon^*)^p)^{1/p} \cdot (k + \binom{k}{2}) \cdot \varepsilon \\ &= \frac{k + \binom{k}{2} - \frac{1}{2}}{k + \binom{k}{2}} \cdot (k + \binom{k}{2}) \cdot \varepsilon = (k + \binom{k}{2} - \frac{1}{2}) \cdot \varepsilon, \end{aligned}$$

which gives the same estimation as previously for $p \in [1, \infty]$ (note that we cannot directly use ε^* as scaling factor, since ε^* might not be rational). □

B.8 Proof of Theorem 7.1

Proof. Let $A = (a_1, \dots, a_n) \in \mathbb{R}^{d \times n}$ be a matrix and let $Z(A) = \sum_{i=1}^{n} \operatorname{conv}\{0, a_i\} \subset \mathbb{R}^d$ be the corresponding zonotope. Defining $c := \frac{1}{2} \sum_{i=1}^{n} a_i$ as the center of the zonotope, we have

$$Z - c = \sum_{i=1}^{n} \operatorname{conv}\{-\frac{a_i}{2}, \frac{a_i}{2}\} = \sum_{i=1}^{n} \operatorname{conv}\{0, -\frac{a_i}{2}\} + \sum_{i=1}^{n} \operatorname{conv}\{0, \frac{a_i}{2}\} \subset \mathbb{R}^d.$$

We now construct the matrix $B = (\frac{a_1}{2}, \dots, \frac{a_n}{2})^\top \in \mathbb{R}^{n \times d}$. Then, we have that

$$\|Bx\|_1 = \sum_{i=1}^{n} |\frac{a_i}{2}^\top x| = \sum_{i=1}^{n} \max\{0, -\frac{a_i}{2}^\top x\} + \sum_{i=1}^{n} \max\{0, \frac{a_i}{2}^\top x\}$$

is the support function of the zonotope $Z - c$.

Applying the polynomial algorithm of Cohen & Peng (2015), we find a matrix $B' = (b'_1, \dots, b'_r)^\top \in \mathbb{R}^{r \times d}$ with $r \in \mathcal{O}(d \log d \varepsilon^{-2})$ such that with high probability, $(1 + \varepsilon)^{-1} \|Bx\|_1 \leq \|B'x\|_1 \leq (1 + \varepsilon) \|Bx\|_1$ holds for all $x \in \mathbb{R}^d$. From the duality between zonotopes and their support function, this implies $(1 + \varepsilon)^{-1} Z((B^\top, -B^\top)) \subseteq Z - c \subseteq (1 + \varepsilon) Z((B^\top, -B^\top))$. Defining $A' = (2b'_1 + c, \dots, 2b'_r + c) \in \mathbb{R}^{d \times r}$, it follows that $(1 + \varepsilon)^{-1} \|Bx\|_1 \leq \|B'x\|_1 \leq (1 + \varepsilon) \|Bx\|_1$ implies $(1 + \varepsilon)^{-1} Z(A') \subseteq Z(A) \subseteq (1 + \varepsilon) Z(A')$, which implies the theorem. □