

COMPLEXITY OF INJECTIVITY AND VERIFICATION OF RELU NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks with ReLU activation play a key role in modern machine learning. Understanding the functions represented by ReLU networks is a major topic in current research as this enables a better interpretability of learning processes.

Injectivity plays a crucial role whenever invertibility of a neural network is necessary, such as, e.g., for inverse problems or generative models. The exact computational complexity of deciding injectivity was recently posed as an open problem (Puthawala et al. [JMLR 2022]). We answer this question by proving coNP-completeness. On the positive side, we show that the problem for a single ReLU-layer is still tractable for small input dimension; more precisely, we present a parameterized algorithm which yields fixed-parameter tractability with respect to the input dimension.

In addition, we study the network verification problem which is of great importance since neural networks are increasingly used in safety-critical systems. We prove that network verification is coNP-hard for a general class of input domains. Our result thus highlights that the hardness of network verification is intrinsic to the ReLU networks themselves, rather than specific input domains.

In this context, we also characterize surjectivity for ReLU networks with one-dimensional output which turns out to be the complement of a basic network verification task. We reveal interesting connections to computational convexity by formulating the surjectivity problem as a zonotope containment problem.

1 INTRODUCTION

Neural networks with rectified linear units (ReLUs) are a widely used model in deep learning. In practice, neural networks are trained on finite datasets and are expected to generalize to new, unseen inputs. However, they often exhibit unexpected and erroneous behavior in response to minor input perturbations; see, e.g., Szegedy et al. (2014). Hence, the certification of trained networks is of great importance and necessitates a thorough understanding of essential properties of the function computed by a ReLU network.

Network verification, that is, the question whether for all inputs from a given subset \mathcal{X} , the ReLU network outputs a value contained in a given set \mathcal{Y} , is a research field gaining high interest recently since neural networks are increasingly used in safety-critical systems like autonomous vehicles (Bajarski et al. (2016)) and collision avoidance for drones (Julian et al. (2019)), see, e.g., Weng et al. (2018); Rössig & Petkovic (2021); Kouvaros & Lomuscio (2021); Katz et al. (2022). Typically, \mathcal{X} and \mathcal{Y} are balls or defined by some linear constraints and the problem is known to be coNP-hard if the sets \mathcal{X} and \mathcal{Y} are part of the input; see Katz et al. (2022); Weng et al. (2018); Sälzer & Lange (2023).

Moreover, recent works focus on studying elementary properties of functions computed by ReLU networks such as injectivity; see, e.g., Puthawala et al. (2022); Haider et al. (2023); Furuya et al. (2023). Injectivity plays a crucial role in many applications where invertibility of a neural network is necessary. Such applications include, for example, generative models, inverse problems, or likelihood estimation; see Puthawala et al. (2022); Furuya et al. (2023) for a more detailed discussion. On the other hand, injectivity might cause privacy issues since the input might be inferred from the output.

Formally, for any number of layers $\ell \in \mathbb{N}$, given weights of a fully connected ReLU network $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$ with ℓ layers, the question is whether the map f is injective. Puthawala et al. (2022) give a mathematical characterization for injectivity of a single ReLU-layer (see Theorem 5), implying an exponential-time algorithm where the exponential part is upper bounded by $\min\{2^m, m^d\}$. In terms of *parameterized complexity*, the problem is *fixed-parameter tractable* with respect to the number m of ReLUs (that is, the superpolynomial part depends only on m), and in XP with respect to the input dimension d (that is, polynomial time for constant d). The complexity status of deciding injectivity of an ℓ -layer ReLU neural network (and therefore also, in particular, a single layer), however, is unresolved and posed as an open problem by Puthawala et al. (2022).

Another natural and fundamental property of functions to consider is surjectivity. Hence, to complete the picture alongside injectivity, we initiate the study of surjectivity for ReLU networks, which, to the best of our knowledge, has not been considered before. In fact, we show that deciding surjectivity can be considered as the complement of a certain network verification task.

Our Contributions. After some preliminaries in Section 2, we show in Section 3 that deciding injectivity of a ReLU network with ℓ layers is coNP-complete (Corollary 8) for any $\ell \in \mathbb{N}$ and thus not polynomial-time solvable, unless $P = NP$. Notably, our hardness reduction reveals some interesting connections between cut problems in (di)graphs and properties of ReLU networks via graphical hyperplane arrangements. We believe that these connections between seemingly unrelated areas are of independent interest. Moreover, our hardness result implies a running time lower bound of $2^{\Omega(m)}$ for a single ReLU-layer with m neurons based on the Exponential Time Hypothesis (Corollary 7). Hence, the running time 2^m is essentially optimal. As regards the input dimension d , however, in Section 4 we give an improved algorithm running in $O((d+1)^d \cdot \text{poly})$ time. This yields fixed-parameter tractability (Theorem 10) and settles the complexity status for injectivity of a ReLU-layer.

Katz et al. (2022) and Sälzer & Lange (2023) show that the network verification problem is coNP-hard for a single hidden layer. However, their hardness results are based on the unit cube as input set. Thus, it was not clear so far whether there are interesting special cases of input sets on which network verification is solvable in polynomial time. In Section 5, we provide the strongest hardness result for network verification to date (Corollary 13), proving coNP-hardness for every possible input domain \mathcal{X} that contains a ball (of possibly lower dimension under some mild conditions). In particular, our result implies that the computational intractability does not stem from choosing a particular set \mathcal{X} , but is rather an intrinsic property of a single hidden ReLU-layer. Since arguably any reasonable input domain contains some ball (this holds, e.g., for all polyhedral sets satisfying the same mild conditions), no special case of verification regarding the input domain is solvable in polynomial time in the worst case. In particular, our result implies hardness for all polyhedra and also all balls as input set, which is not covered by the previous hardness results. As a consequence, one must make (very) specific assumptions on the network in order to obtain tractable cases, e. g., by building special (approximating) networks that are efficiently verifiable (Baader et al. (2020), Wang et al. (2022), Baader et al. (2024)).

Moreover, we give a characterization of surjectivity for ReLU networks with one-dimensional output (Lemma 14) which implies a polynomial-time algorithm for constant input dimension d . We then proceed with proving NP-hardness of surjectivity by showing that it can be phrased as (the complement of) a special case of network verification. Finally, we also show that surjectivity can be formulated as a zonotope containment problem, which is of fundamental importance, e.g., in robotics (Kulmburg & Althoff (2021)).

Related Work. Puthawala et al. (2022) initiate the study of injectivity of ReLU networks. Their result implicitly yields an exponential-time algorithm. They also mention a connection between injectivity and the *spark* of the weight matrix (that is, the minimum number of linearly dependent rows). Computing the spark is known to be NP-hard (Tillmann & Pfetsch (2014)), and also known not to be fixed-parameter tractable with respect to d , unless $FPT = W[1]$ (Panolan et al. (2015)). Haider et al. (2023) use a frame-theoretic approach to study the injectivity of a ReLU-layer on a closed ball. They give an exponential-time algorithm to determine a bias vector for a given weight matrix such that the corresponding map is injective on a closed ball. Furuya et al. (2023) study injectivity of ReLU-layers with linear neural operators.

The network verification problem is known to be coNP-hard for networks with two layers and arbitrary output dimension (Katz et al. (2022) proved NP-hardness for the complement). Sälzer & Lange

(2023) corrected some flaws in the proof and showed coNP-hardness for one-dimensional output. The problem is also known to be coNP-hard to approximate in polynomial time for one-dimensional output if the number of layers is unbounded (Weng et al. (2018)) and for three hidden layers and squashable activation functions Wang et al. (2022). Efficient verifiable networks with heuristics such as the interval bound propagation Gowal et al. (2018) are shown to be universal approximators Wang et al. (2022); Baader et al. (2020), but the networks might have exponential size. On the other hand, obtaining exact representations for all functions computed by ReLU neural networks is not possible such that they are precisely verifiable with interval bound propagation Mirman et al. (2022) or single neuron convex relaxations Baader et al. (2024). Further heuristics include methods such as DeepPoly (Singh et al. (2019)), DeepZ (Wong et al. (2018)), general cutting planes (Zhang et al. (2024)), multi neuron verification (Ferrari et al. (2022)). Moreover, there exist libraries for verification (Xu et al. (2020); Mao et al. (2024)).

Parameterized complexity has also been studied for the training problem for ReLU neural networks by Froese et al. (2022); Froese & Hertrich (2023). In general, understanding the complexity/expressivity of ReLU neural networks is an important task (Hertrich et al. (2021); Arora et al. (2018)).

2 PRELIMINARIES

We introduce the basic definitions and concepts involving ReLU neural networks and their geometry which are relevant for this work.

Definition 1. A ReLU-layer with d inputs, m outputs, weights $\mathbf{W} \in \mathbb{R}^{m \times d}$, and biases $\mathbf{b} \in \mathbb{R}^m$ computes a map $\phi_{\mathbf{W}, \mathbf{b}}: \mathbb{R}^d \rightarrow \mathbb{R}^m$, $\mathbf{x} \mapsto [\mathbf{W}\mathbf{x} + \mathbf{b}]_+$, where $[\cdot]_+: \mathbb{R}^m \rightarrow \mathbb{R}^m$ is the *rectifier function* given by $[\mathbf{x}]_+ := (\max\{0, x_1\}, \dots, \max\{0, x_m\})$.

For most of our purposes, we can assume $\mathbf{b} = \mathbf{0}$ without loss of generality. In that cases, we omit \mathbf{b} and simplify notation $\phi_{\mathbf{W}} := \phi_{\mathbf{W}, \mathbf{0}}$. A deep ReLU network is just a concatenation of compatible ReLU-layers.

Definition 2. An ℓ -layer ReLU network of architecture $(n_0, n_1, \dots, n_{\ell-1}, n_\ell)$ with weights $\mathbf{W}_i \in \mathbb{R}^{n_{i-1} \times n_i}$ and biases $\mathbf{b}_i \in \mathbb{R}^{n_i}$ for $i \in \{1, \dots, \ell\}$ computes a map

$$f: \mathbb{R}^d \rightarrow \mathbb{R}^m, \quad \mathbf{x} \mapsto \mathbf{W}_\ell \cdot (\phi_{\mathbf{W}_{\ell-1}, \mathbf{b}_{\ell-1}} \circ \dots \circ \phi_{\mathbf{W}_1, \mathbf{b}_1})(\mathbf{x}) + \mathbf{b}_\ell,$$

where $d = n_0$ and $m = n_\ell$.

Whenever we consider an ℓ -layer ReLU neural network $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$ as the input to a decision problem in the paper, we implicitly consider its weights and biases as the input. For most of our purposes we can assume without loss of generality $\mathbf{b}_i = \mathbf{0}$ and hence the technical parts only deal with such ReLU-layers.

Geometry of ReLU-layers. We review basic definitions from polyhedral geometry; see Schrijver (1986) for more details. For a row vector \mathbf{w}_i , the hyperplane $H_{\mathbf{w}_i} := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}_i \mathbf{x} = 0\}$ subdivides \mathbb{R}^d into half-spaces $H_{\mathbf{w}_i}^+ := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}_i \mathbf{x} \geq 0\}$ and $H_{\mathbf{w}_i}^- := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}_i \mathbf{x} \leq 0\}$. 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 hyperplane *supports* P if it bounds a closed halfspace containing P , and any intersection of P with such a supporting hyperplane yields a *face* F of P . A *polyhedral complex* \mathcal{P} is a finite collection of polyhedra such that (i) $\emptyset \in \mathcal{P}$, (ii) if $P \in \mathcal{P}$ then all faces of P are in \mathcal{P} , and (iii) if $P, P' \in \mathcal{P}$, then $P \cap P'$ is a face of both P and P' . A *polyhedral fan* is a polyhedral complex in which every polyhedron is a cone. For a matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$, the set of full-dimensional polyhedral cones

$$\mathcal{C}_{\mathbf{W}} := \left\{ \bigcap_{i=1}^m H_{\mathbf{w}_i}^{s_i} \mid (s_1, \dots, s_m) \in \{+, -\}^m, \dim\left(\bigcap_{i=1}^m H_{\mathbf{w}_i}^{s_i}\right) = d \right\}$$

subdivides \mathbb{R}^d and the set $\{C \cap H_{\mathbf{w}_i} \mid C \in \mathcal{C}_{\mathbf{W}}, i \in [m]\}$ forms a polyhedral fan. A vector \mathbf{x} in the support $|\Sigma_{\mathbf{W}}| = \bigcup_{\sigma \in \Sigma_{\mathbf{W}}} \sigma$ of the fan $\Sigma_{\mathbf{W}}$ is called

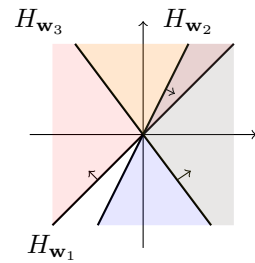


Figure 1: The polyhedral fan induced by an oriented linear hyperplane arrangement given by a matrix $\mathbf{W} \in \mathbb{R}^{3 \times 2}$.

a *breakpoint* of $\phi_{\mathbf{W}}$. See Figure 1 for an illustration of a 2-dimensional polyhedral fan arising from a ReLU-layer.

For $C \in \mathcal{C}_{\mathbf{W}}$, we define the *active set* $I_C := \{j \in [m] \mid \forall \mathbf{x} \in C : \mathbf{w}_j \mathbf{x} \geq 0\}$ and the matrix $\mathbf{W}_C \in \mathbb{R}^{m \times d}$, where $(\mathbf{W}_C)_j := \begin{cases} \mathbf{w}_j, & j \in I_C, \\ \mathbf{0}, & j \in [m] \setminus I_C. \end{cases}$ Note that the map $\phi_{\mathbf{W}}$ is linear on C , namely $\phi_{\mathbf{W}}(\mathbf{x}) = \mathbf{W}_C \cdot \mathbf{x}$ for $\mathbf{x} \in C$.

Geometry of ReLU Neural Networks. It is well-known that also a deep ReLU neural network f partitions its input space into polyhedra on which f is affine linear (Hanin & Rolnick (2019); Grigsby & Lindsey (2022)). More precisely, let $f_{i,j} = \pi_j \circ \phi_{\mathbf{W}_i, \mathbf{b}_i} \circ \dots \circ \phi_{\mathbf{W}_1, \mathbf{b}_1}$, where $\pi_j : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ is the projection onto the j -th coordinate. Then, every $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_{n_{\ell-1}}) \in \{-, +\}^{n_1} \times \dots \times \{-, +\}^{n_{\ell-1}}$ corresponds to a (possibly empty) polyhedron

$$P_{\mathbf{s}} = \bigcap_{(\mathbf{s}_i)_j=+} \{\mathbf{x} \in \mathbb{R}^d \mid f_{i,j}(\mathbf{x}) \geq 0\} \cap \bigcap_{(\mathbf{s}_i)_j=-} \{\mathbf{x} \in \mathbb{R}^d \mid f_{i,j}(\mathbf{x}) \leq 0\}.$$

The maps $f_{i,j}$ are affine linear on $P_{\mathbf{s}}$ and the coefficients are polynomially bounded in the weights and biases of the neural network. Hence, the polyhedron $P_{\mathbf{s}}$ arises as the intersection of at most $\sum_{i=1}^{\ell} n_i$ half-spaces whose encoding sizes are polynomially bounded in the weights and biases of f . All polyhedra $P_{\mathbf{s}}$ and their faces form a polyhedral complex which we denote by Σ_f .

A *ray* ρ is a one-dimensional pointed cone; a vector \mathbf{r} is a *ray generator* of ρ if $\rho = \{\lambda \mathbf{r} \mid \lambda \geq 0\}$. For each ray ρ of Σ_f , let \mathbf{r}_{ρ} be the unique unit ray generator of ρ having norm 1 and let $\mathcal{R} := \{\mathbf{r}_{\rho} \mid \rho \text{ is a ray in } \Sigma_f\}$. A ray $\rho \subseteq C$ of a cone C is an *extreme ray* if there do not exist $\lambda_1, \lambda_2 > 0$ and $\rho_1, \rho_2 \subseteq C$ such that $\rho = \lambda_1 \rho_1 + \lambda_2 \rho_2$. We make the following simple observation.

Observation 3. *Let C be a pointed polyhedral cone such that f is linear on C and let $\mathbf{r}_1, \dots, \mathbf{r}_{\ell}$ be ray generators of the extreme rays of C . Then, for each $\mathbf{x} \in C$, there are $\lambda_1, \dots, \lambda_{\ell} \in \mathbb{R}$ such that $\mathbf{x} = \sum_{i=1}^{\ell} \lambda_i \mathbf{r}_i$ and $f(\mathbf{x}) = \sum_{i=1}^{\ell} \lambda_i f(\mathbf{r}_i)$.*

Observation 3 implies that the map f is essentially determined by its values on \mathcal{R} , if all cones $C \in \Sigma_f$ are pointed.

(Parameterized) Complexity Theory. We assume the reader to be familiar with basic concepts from classical complexity theory like P, NP, and NP-completeness. The class coNP contains all decision problems whose complement is in NP. A decision problem is coNP-complete if and only if its complement is NP-complete. Clearly, a coNP-complete problem cannot be solved in polynomial time unless P = NP.

A *parameterized* problem consists of instances (x, k) , where x encodes the classical instance and $k \in \mathbb{N}$ is a *parameter*. A parameterized problem is in the class XP if it is polynomial-time solvable for every constant parameter value, that is, in $O(|x|^{f(k)})$ time for an arbitrary function f depending only on the parameter k . A parameterized problem is *fixed-parameter tractable* (contained in the class FPT) if it is solvable in $f(k) \cdot |x|^{O(1)}$ time, for an arbitrary function f . Clearly, $\text{FPT} \subseteq \text{XP}$; see Downey & Fellows (2013) for further details of parameterized complexity.

3 coNP-COMPLETENESS OF INJECTIVITY

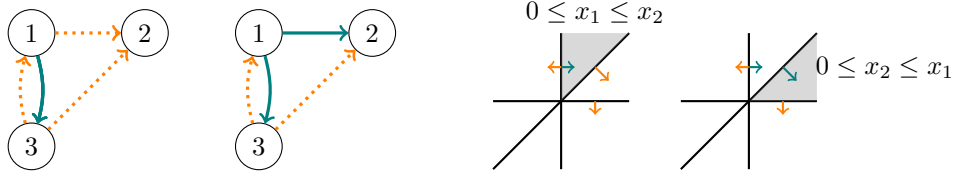
In this section we study the computational complexity of ℓ -RELU-LAYER INJECTIVITY, that is, deciding whether a ReLU network $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ with ℓ layers computes an injective map.

Proposition 4. *For every $\ell \in \mathbb{N}$, it holds that ℓ -RELU-LAYER INJECTIVITY is contained in coNP.*

Proof sketch. Two maximal polyhedra $P, Q \in \Sigma_f$ serve as a certificate for non-injectivity, since checking whether there are $\mathbf{x}_1 \in P$ and $\mathbf{x}_2 \in Q$ with $\mathbf{x}_1 \neq \mathbf{x}_2$ such that $f(\mathbf{x}_1) = f(\mathbf{x}_2)$ is simply checking feasibility of a linear program. For a rigorous proof one can apply Theorem 3.3 in Sälzer & Lange (2023). \square

To show that ℓ -RELU-LAYER INJECTIVITY is coNP-hard for any $\ell \in \mathbb{N}$, it suffices to show that deciding injectivity is already coNP-hard for a single ReLU-layer. Hence, in the remainder of this

216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269



(a) For the ordering $3 \leq 2 \leq 1$ (on the left) and the ordering $3 \leq 1 \leq 2$ (on the right), the arcs that respect the ordering are dotted and colored in orange.

(b) The resulting ReLU network represented by the corresponding oriented hyperplane arrangements. For the gray colored cones, the inactive neurons are colored in orange.

Figure 2: An illustration of the reduction from ACYCLIC 2-DISCONNECTION to RELU-LAYER NON-INJECTIVITY for a digraph with $n = 3$ nodes. An ordering π of the coordinates induces an acyclic subset A_π of arcs corresponding to inactive neurons on the cone $C_\pi = \{x_{\pi(1)} \leq \dots \leq x_{\pi(n)}\}$. The active neurons on C_π have full rank n if and only if removing the arcs A_π results in a weakly connected digraph.

section we study the following decision problem (RELU-LAYER INJECTIVITY): Given a matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$ and a vector $\mathbf{b} \in \mathbb{R}^m$, is the map $\phi_{\mathbf{W}, \mathbf{b}}$ injective?

Puthawala et al. (2022, Theorem 2) prove the following characterization of injectivity for a map computed by a single ReLU-layer.

Theorem 5 (Puthawala et al. (2022)). *A ReLU-layer $\phi_{\mathbf{W}}$ with $\mathbf{W} \in \mathbb{R}^{m \times d}$ is injective if and only if \mathbf{W}_C has (full) rank d for all $C \in \mathcal{C}_{\mathbf{W}}$.*

Since any hyperplane arrangement with m hyperplanes in d dimensions defines at most $O(m^d)$ (also clearly at most 2^m) cells (Zaslavsky (1975)), Theorem 5 implies an algorithm that solves RELU-LAYER INJECTIVITY in $O(\min\{2^m, m^d\} \cdot \text{poly}(S))$ time, where S denotes the input size.

We prove that RELU-LAYER INJECTIVITY is coNP-complete. To this end, we show NP-completeness for the complement problem RELU-LAYER NON-INJECTIVITY. Based on Theorem 5 and the fact that one can w.l.o.g. assume $\mathbf{b} = \mathbf{0}$ (Puthawala et al., 2022, Lemma 3), the RELU-LAYER NON-INJECTIVITY problem is, given a matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$, to decide if there is a cell $C \in \mathcal{C}_{\mathbf{W}}$ such that $\text{rank}(\mathbf{W}_C) < d$.

For the NP-hardness proof, we reduce from the following directed graph (*digraph*) problem. A digraph $D = (V, A)$ is called *acyclic* if it does not contain oriented cycles and *weakly disconnected* if the underlying graph, i.e., the graph where we have an unoriented edge for every arc, is connected. The ACYCLIC 2-DISCONNECTION problem is, given a digraph $D = (V, A)$, to decide if there is a subset $A' \subseteq A$ of arcs such that (V, A') is acyclic and $(V, A \setminus A')$ is not weakly connected.

The problem is a special case of ACYCLIC s -DISCONNECTION where the goal is to remove an acyclic arc set such that the remaining digraph contains at least s weakly connected components. This more general problem is known to be NP-hard if s is part of the input (Figueroa et al. (2017)). In Appendix A.1 we prove NP-hardness for our special case $s = 2$ (Theorem 18). We remark that our reduction implies that ACYCLIC 2-DISCONNECTION cannot be solved in $2^{o(|D|)}$ time unless the Exponential Time Hypothesis¹ fails (Corollary 19).

Theorem 6. RELU-LAYER NON-INJECTIVITY is NP-complete even if every row of \mathbf{W} contains at most two non-zero entries.

Proof sketch. Containment in NP is easy: The set I_C of a cell C with $\text{rank}(\mathbf{W}_C) < d$ serves as a certificate. To prove NP-hardness, we reduce from ACYCLIC 2-DISCONNECTION which is NP-hard by Theorem 18. We sketch the proof here, a detailed proof can be found in Appendix A.1.1. Moreover, the reduction is illustrated in Figure 2. Given a digraph $D = (V, A)$ with $V = [n]$, we construct a ReLU-layer $\phi: \mathbb{R}^{n-1} \rightarrow \mathbb{R}^{|A|}$ by $\phi(\mathbf{x}) = (\max\{0, \mathbf{x}_i - \mathbf{x}_j\})_{(i,j) \in A}$. For every permutation $\pi \in \mathcal{S}_n$, function ϕ is linear on the cone $C_\pi := \{\mathbf{x} \mid \mathbf{x}_{\pi(1)} \leq \dots \leq \mathbf{x}_{\pi(n)}\}$, where we

¹The Exponential Time Hypothesis asserts that 3-SAT cannot be solved in $2^{o(n)}$ time where n is the number of Boolean variables in the input formula (Impagliazzo & Paturi (2001)).

let $\mathbf{x}_n := \mathbf{0}$, since no hyperplane $\mathbf{x}_i = \mathbf{x}_j$ induced by a neuron $\max\{0, \mathbf{x}_i - \mathbf{x}_j\}$ intersects this cone. Let $A_\pi := \{(i, j) \in A \mid \pi(i) \leq \pi(j)\} \subseteq A$ be the (acyclic) subset of arcs respecting the (total) order on the nodes induced by $\pi \in \mathcal{S}_n$. The set of arcs in the complement $A \setminus A_\pi$ is therefore in bijection with the neurons that are active on C_π , i.e., neurons $\max\{0, \mathbf{x}_i - \mathbf{x}_j\}$ where $\pi(j) \leq \pi(i)$. In Appendix A.1.1, we prove that the digraph $D_\pi = (V, A \setminus A_\pi)$ is weakly connected if and only if

- there is a path in the underlying graph of D_π from node j to node n , for all $j \in [n - 1]$;
- if and only if the standard unit vector \mathbf{e}_j is contained in $\text{span}(\mathbf{W}_{C_\pi})$, for all $j \in [n - 1]$;
- if and only if \mathbf{W}_{C_π} has full rank.

The illustration for the 2-dimensional case in Figure 2 provides intuition for this chain of equivalent statements. Since the partial order on the nodes induced by an acyclic subset of arcs can be extended to a total order on the nodes, it holds that ϕ is not injective if and only if there is $A' \subseteq A$ such that (V, A') is acyclic and $(V, A \setminus A')$ is not weakly connected, proving the correctness of the reduction. \square

The lower bound from Corollary 19 actually transfers to RELU-LAYER INJECTIVITY since our polynomial-time reduction in the proof of Theorem 6 yields a matrix where the number of rows/columns is linear in the number arcs/nodes of the digraph.

Corollary 7. RELU-LAYER INJECTIVITY is coNP-complete even if every row of \mathbf{W} contains at most two non-zero entries (and $\mathbf{b} = \mathbf{0}$). Moreover, RELU-LAYER NON-INJECTIVITY and RELU-LAYER INJECTIVITY cannot be solved in $2^{o(m+d)}$ time, unless the ETH fails.

To prove coNP-hardness for an arbitrary number ℓ of layers, one can simply reduce RELU-LAYER INJECTIVITY to ℓ -RELU-LAYER INJECTIVITY by concatenating the layer with a ReLU network with $\ell - 1$ layers that computes the identity map. Hence, we obtain the following corollary.

Corollary 8. For every $\ell \geq 2$, it holds that ℓ -RELU-LAYER INJECTIVITY is coNP-complete.

4 AN FPT ALGORITHM FOR RELU-LAYER INJECTIVITY

The complexity results from the previous section exclude polynomial running times as well as running times subexponential in $m + d$ for deciding injectivity for a single ReLU-layer. Nevertheless, we show in this section that the previous upper bound of m^d can be improved to $(d + 1)^d$. We achieve this with a branching algorithm which searches for a “non-injective” cell C , that is, \mathbf{W}_C has rank strictly less than d . The algorithm branches on the ReLUs active in C , thus restricting the search space to some cone. The pseudocode is given in Algorithms 1 and 2.

The key idea to bound the running time of our search tree algorithm is to show that there are always at most $d + 1$ candidate ReLUs to set active. The candidates are those whose corresponding halfspaces cover the current search space; that is, at least one of them will be active in the sought cell (cf. Line 5 of Algorithm 2). Helly’s Theorem (Helly (1923)) ensures that there are always at most $d + 1$ such candidates. Using duality of linear programming, we show that these halfspaces can be found in polynomial time. A more detailed proof of the following lemma is given in Appendix A.2.1.

Lemma 9. Let $C \subseteq \mathbb{R}^d$ be a cone and let $\{\mathbf{w}_1, \dots, \mathbf{w}_n\} \subseteq \mathbb{R}^d$ be such that C is covered by the corresponding half-spaces, that is, $C \subseteq \bigcup_{i=1}^n H_{\mathbf{w}_i}^+$. Then, there exists a subset $A \subseteq [n]$ of size at most $d + 1$ computable in polynomial time, such that $C \subseteq \bigcup_{i \in A} H_{\mathbf{w}_i}^+$.

Algorithm 1: LayerInjectivity

Input : $W = \{\mathbf{w}_1, \dots, \mathbf{w}_m\} \subseteq \mathbb{R}^d$

Output: $\mathbf{x} \in \mathbb{R}^d$ with $\text{rank}(\{\mathbf{w}_i \in W \mid \mathbf{w}_i^T \mathbf{x} \geq 0\}) < d$ (if it exists); otherwise, “yes”

- 1 **if** $\text{rank}(W) < d$, **then return** $\mathbf{0}$;
 - 2 **if** $\exists \mathbf{x} \in \mathbb{R}^d : \forall i \in [m] : \mathbf{w}_i^T \mathbf{x} < 0$, **then return** \mathbf{x} ;
 - 3 $x \leftarrow \text{FindCell}(\emptyset, W)$;
 - 4 **if** $x \neq \text{“no”}$, **then return** x ;
 - 5 **return** “yes”
-

Algorithm 2: FindCell

Input : vectors $C = \{\mathbf{c}_1, \dots, \mathbf{c}_n\} \subseteq \mathbb{R}^d$ and $M = \{\mathbf{m}_1, \dots, \mathbf{m}_m\} \subseteq \mathbb{R}^d$
Output: vector $\mathbf{x} \in \mathbb{R}^d$ (if it exists) such that $\mathbf{c}_i^T \mathbf{x} \geq 0$, for all $i \in [n]$, and
 $\text{rank}(C \cup \{\mathbf{m}_i \mid \mathbf{m}_i^T \mathbf{x} \geq 0\}) < d$; otherwise, “no”

- 1 **if** $\text{rank}(C) = d \vee \{\mathbf{x} \mid \forall i \in [n] : \mathbf{c}_i^T \mathbf{x} \geq 0\} = \{\mathbf{0}\}$, **then return** “no”;
- 2 $I \leftarrow \{i \in \{1, \dots, m\} \mid \mathbf{m}_i \notin \text{span}(C)\}$;
- 3 $M \leftarrow \{\mathbf{m}_i \mid i \in I\}$;
- 4 **if** $\exists \mathbf{x} \in \mathbb{R}^d : ((\forall i \in [n] : \mathbf{c}_i^T \mathbf{x} \geq 0) \wedge (\forall i \in I : \mathbf{m}_i^T \mathbf{x} < 0))$, **then return** \mathbf{x} ;
- 5 **compute** $A \subseteq I$ such that $\{\mathbf{x} \mid \forall i \in [n] : \mathbf{c}_i^T \mathbf{x} \geq 0\} \subseteq \bigcup_{i \in A} H_{\mathbf{m}_i}^+$ and $|A| \leq d + 1$;
- 6 **foreach** $i \in A$ **do**
- 7 $x \leftarrow \text{FindCell}(C \cup \{\mathbf{m}_i\}, M \setminus \{\mathbf{m}_i\})$;
- 8 **if** $x \neq \text{“no”}$, **then return** x ;
- 9 **end**
- 10 **return** “no”

As regards the correctness of our algorithm, assume that there exists a vector $\mathbf{x} \in \mathbb{R}^d$ such that $\text{rank}(\{\mathbf{w}_i \mid \mathbf{w}_i^T \mathbf{x} \geq 0\}) = k < d$. Then, after every branching, \mathbf{x} is contained in some cone that is the search space of FindCell, that is, $\mathbf{c}_i^T \mathbf{x} \geq 0$ for all $i \in [n]$. Notice that every branching increases the number of linear independent neurons that are active in the current cone by one. Thus, after at most k branchings, FindCell finds a cone where the map is not injective. Conversely, if the algorithm outputs an $\mathbf{x} \in \mathbb{R}^d$, then the matrix $\mathbf{W}_{\bar{C}}$ corresponding to the cone $\bar{C} := \{\bar{\mathbf{x}} \mid \forall i \in [n] : \mathbf{c}_i^T \bar{\mathbf{x}} \geq 0\}$ containing \mathbf{x} cannot have full rank; otherwise, Line 1 in FindCell would have output “no”. Lemma 9 ensures that we can always branch on $d + 1$ neurons. Furthermore, the number of active neurons increases after every branching by one. Hence the search tree has at most $(d + 1)^d$ nodes. We conclude our findings in the following theorem for which we provide a detailed proof in Appendix A.2.2.

Theorem 10. RELU-LAYER INJECTIVITY is solvable in $O((d + 1)^d \cdot \text{poly}(S))$ time, where S denotes the input size.

Deciding injectivity for a deep neural network $f = \phi_\ell \circ \dots \circ \phi_1$ is even more involved. Clearly, if all layer maps ϕ_i are injective, then also Φ is injective. The converse does not hold, however, since ϕ_i only needs to be injective on the image $(\phi_{i-1} \circ \dots \circ \phi_1)(\mathbb{R}^d)$ for all $i \in [\ell] := \{1, \dots, \ell\}$. Hence, it is unclear whether the problem is still contained in FPT when parameterized by d .

5 VERIFICATION AND SURJECTIVITY

In this section, we study the network verification task and prove hardness for a very general class of input sets. To this end, we call a sequence $S = (S_d)_{d \in \mathbb{N}}$ of subsets $S_d \subseteq \mathbb{R}^d$ *reasonable* if there exists an algorithm that on input $k \in \mathbb{N}$ computes in $\text{poly}(k)$ time a $d \in \mathbb{N}$, $\mathbf{z} \in \mathbb{R}^d$ and a k -dimensional affine space A in \mathbb{R}^d (given by a basis and a translation vector) such that there is a k -dimensional ball with center \mathbf{z} contained in S_d that affinely spans A . For a sequence $S = (S_d)_{d \in \mathbb{N}}$ of reasonable sets and a polyhedron $Q_{\mathbf{t}} := \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x}_i \leq \mathbf{t}_i \text{ for all } i \in [m]\}$ where $\mathbf{t} \in \mathbb{R}^m$, the task to decide for a given ℓ -layer ReLU network $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ whether $f(S_d) \subseteq Q_{\mathbf{t}}$ we call ℓ -LAYER RELU $(S, Q_{\mathbf{t}})$ -VERIFICATION.

Even though the definition of a reasonable sequence is a bit involved, they include a wide range of set-sequences. For example, they trivially include all sequences of sets that contain an open neighborhood of the origin and therefore in particular all sequences of full-dimensional polyhedra containing the origin in the interior. This shows, in comparison to previous existing coNP-hardness proofs, that the hardness of verification does not rely on the complexity of the polyhedron in the input domain but is intrinsic to the ReLU networks themselves.

To show that the problem is coNP-hard, it suffices to show hardness for one hidden layer and one-dimensional output and hence in the remaining part we study the following decision problem for an arbitrary $t \in \mathbb{R}$.

2-LAYER RELU (S, t) -VERIFICATION

Input: Matrices $\mathbf{W}_1 \in \mathbb{R}^{n \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times n}$, $\mathbf{b}_1 \in \mathbb{R}^n$, and $b_2 \in \mathbb{R}$.
Question: Is $\mathbf{W}_2 \cdot \phi_{\mathbf{W}_1, \mathbf{b}_1}(\mathbf{x}) + b_2 \leq t$ for all $\mathbf{x} \in S_d$?

We will see in Theorem 12 that the hardness of 2-LAYER RELU (S, t) -VERIFICATION stems from the hardness of deciding whether a ReLU network computes a map that attains a positive output value. Thus, we will first show that the 2-LAYER RELU POSITIVITY problem is NP-complete: Given Matrices $\mathbf{W}_1 \in \mathbb{R}^{n \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times n}$, is there an $\mathbf{x} \in \mathbb{R}^d$ such that $\mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{x}) > 0$?

Afterwards, we give a reduction from the complement of 2-LAYER RELU POSITIVITY to 2-LAYER RELU (S, t) -VERIFICATION implying that 2-LAYER RELU (S, t) -VERIFICATION and thus ℓ -LAYER RELU (S, Q_t) -VERIFICATION are coNP-hard.

5.1 NP-COMPLETENESS OF 2-LAYER RELU POSITIVITY

We prove NP-completeness of 2-LAYER RELU POSITIVITY via reduction from POSITIVE CUT. For a graph $G = (V, E)$, we denote by $E(S, V \setminus S) := \{\{u, v\} \in E \mid u \in S, v \notin S\}$ the edges in the cut induced by $S \subseteq V$. The POSITIVE CUT problem is to decide for a given graph $G = (V, E)$ with edge weights $w: E \rightarrow \mathbb{Z}$ whether there is a subset $S \subseteq V$ such that $\sum_{e \in E(S, V \setminus S)} w(e) > 0$, which we prove to be NP-complete in Appendix A.3.4. A detailed proof of the following theorem is given in Appendix A.3.5.

Theorem 11. 2-LAYER RELU POSITIVITY is NP-complete.

Proof sketch. The problem is contained in NP since, by Observation 3, it is sufficient to identify an extreme ray ρ as a certificate for positivity.

To prove NP-hardness, we reduce from POSITIVE CUT. Given a weighted graph $(G = (V, E), w)$, we define a 2-layer ReLU neural network $f: \mathbb{R}^{|V|} \rightarrow \mathbb{R}$, where we have two hidden neurons for every $e \in E$ and the output weight of the neurons are the corresponding weights of the edges. More precisely,

$$f(\mathbf{x}) = \sum_{\{i, j\} \in E} w(\{i, j\}) \cdot ([\mathbf{x}_i - \mathbf{x}_j]_+ + [\mathbf{x}_j - \mathbf{x}_i]_+).$$

The weights of the cuts are now stored as function values on certain vectors. For a subset $S \subseteq V$, let $\mathbf{r}_S := \sum_{i \in S} \mathbf{e}_i \in \mathbb{R}^{|V|}$ and $\mathbf{r}'_S := -\sum_{i \in S \setminus V} \mathbf{e}_i \in \mathbb{R}^d$. It follows easily that

$$f(\mathbf{r}_S) = f(\mathbf{r}'_S) = \sum_{\{i, j\} \in E(S, V \setminus S)} w(\{i, j\}).$$

We proceed by showing that every \mathbf{x} is the conic combination of such \mathbf{r}_S and \mathbf{r}'_S . Hence, there is an $\mathbf{x} \in \mathbb{R}^{|V|}$ such that $f(\mathbf{x}) > 0$ if and only if there is an $S \subseteq V$ such that $f(\mathbf{r}_S) = \sum_{e \in E(S, V \setminus S)} w(e) > 0$, proving the correctness of the reduction. \square

5.2 coNP-HARDNESS OF VERIFICATION

We continue with the coNP-hardness for network verification using the NP-hardness of 2-LAYER RELU POSITIVITY.

Theorem 12. For every reasonable set-sequence S and $t \in \mathbb{R}$, it holds that 2-LAYER RELU (S, t) -VERIFICATION is coNP-hard.

Proof sketch. In Appendix A.3.1, we provide a reduction from the complement of 2-LAYER RELU POSITIVITY that we sketch here. Let $f: \mathbb{R}^k \rightarrow \mathbb{R}$ be the function computed by an instance of 2-LAYER RELU POSITIVITY and let the dimension d , the affine space A and the point $\mathbf{z} \in \mathbb{R}^d$ be the output of the algorithm that exists due to the fact that $(S_d)_{d \in \mathbb{N}}$ is a reasonable sequence of sets. Let $T: \mathbb{R}^d \rightarrow \mathbb{R}^d$ be the affine map that maps \mathbf{z} to the origin composed with the affine map that projects orthogonal to A and afterwards maps isomorphic to \mathbb{R}^k . Then we prove that $\mathbf{x} \mapsto (f \circ T)(\mathbf{x}) + t$ is the function computed by a “no”-instance of 2-LAYER RELU (S, t) -VERIFICATION if and only if f attains a positive output value. \square

To prove coNP-hardness for an arbitrary number $\ell \geq 2$ of layers and arbitrary output dimension m , one can simply provide a reduction from 2-LAYER RELU (S, t) -VERIFICATION to ℓ -LAYER RELU (S, Q_t) -VERIFICATION by concatenating the 2-layer ReLU network with a ReLU network with $\ell - 1$ layers that computes m many maps in parallel that translate t to $\mathbf{t} \in \mathbb{R}^m$ coordinate-wise. Hence, we obtain the following corollary.

Corollary 13. *For every reasonable set-sequence S , every polyhedron $Q_t \subseteq \mathbb{R}^m$ with $\mathbf{t} \in \mathbb{R}^m$ and every $\ell \geq 2$, it holds that ℓ -LAYER RELU (S, Q_t) -VERIFICATION is coNP-hard.*

5.3 SURJECTIVITY

In this section, we study the task to decide whether a given ReLU network $f: \mathbb{R}^d \rightarrow \mathbb{R}$ with ℓ layers computes a surjective map. We will call this task ℓ -LAYER RELU SURJECTIVITY. We start with some simple observations characterizing surjectivity; see Appendix A.3.2 for a proof.

Lemma 14. *For a ReLU network $f = \mathbf{W}_\ell \circ \phi_{\mathbf{W}_{\ell-1}, \mathbf{b}_{\ell-1}} \circ \cdots \circ \phi_{\mathbf{W}_1, \mathbf{b}_1} + b_\ell$ we denote by $f_0 := \mathbf{W}_\ell \circ \phi_{\mathbf{W}_{\ell-1}} \circ \cdots \circ \phi_{\mathbf{W}_1}$ the corresponding ReLU network without biases. Then the following holds:*

- a) f_0 is surjective if and only if there are $\mathbf{v}^+, \mathbf{v}^- \in \mathbb{R}^d$ such that $f_0(\mathbf{v}^+) > 0$ and $f_0(\mathbf{v}^-) < 0$.
- b) f is surjective if and only if f_0 is surjective
- c) The map f is surjective if and only if there exist two ray generators $\mathbf{r}^+, \mathbf{r}^-$ of two rays $\rho^+, \rho^- \in \Sigma_{f_0}$ such that $f_0(\mathbf{r}^+) > 0$ and $f_0(\mathbf{r}^-) < 0$.

Lemma 14 b) implies that we can assume that the ReLU network has no biases without loss of generality. Furthermore, Lemma 14 c) implies an exponential-time algorithm for ℓ -LAYER RELU SURJECTIVITY, since Σ_{f_0} contains at most $O(\binom{\ell}{d-1} n_i^d)$ many rays. In particular, for $\ell = 2$, the fan $\Sigma_{\mathbf{W}_1}$ contains at most $\min\{2n_1^{d-1}, 2^{n_1+1}\}$ rays; that is, the problem is in XP when parameterized by d and in FPT when parameterized by n_1 . Moreover, since two ray generators $\mathbf{r}^+, \mathbf{r}^-$ of two rays $\rho^+, \rho^- \in \Sigma_{f_0}$ serve as a certificate, we obtain the following proposition.

Proposition 15. *For every $\ell \in \mathbb{N}$, it holds that ℓ -LAYER RELU SURJECTIVITY is in NP.*

Again, to show that the problem is NP-hard, it suffices to show hardness for one hidden layer and one-dimensional output. Hence, in the remainder we study 2-LAYER RELU SURJECTIVITY, where we have one hidden layer with n neurons (with weights $\mathbf{W}_1 \in \mathbb{R}^{n \times d}$ and bias $\mathbf{b}_1 \in \mathbb{R}^n$) with ReLU activation, and an output layer with one output neuron (with weights $\mathbf{W}_2 \in \mathbb{R}^{1 \times n}$) without activations. The network then computes the map $f: \mathbb{R}^d \rightarrow \mathbb{R}$ with $f(\mathbf{x}) := \mathbf{W}_2 \cdot \phi_{\mathbf{W}_1, \mathbf{b}_1}(\mathbf{x})$.

In fact, to decide surjectivity, it is actually enough to find one (say the positive) ray generator since it is easy to find some point \mathbf{x} where f is non-zero (we argue below that w.l.o.g. $f(\mathbf{x}) < 0$). To find \mathbf{x} , we choose an arbitrary full-dimensional cone and determine whether \mathbf{W}_C is the zero map or not. In the former case, one can pick an arbitrary full-dimensional cone C' that shares a facet with C . Then, $\mathbf{W}_{C'} \neq \mathbf{0}$ since the neuron defining the facet must be active in C' . Hence, one finds a point $\mathbf{x} \in C'$ with $f(\mathbf{x}) \neq 0$. We give a more detailed proof of the following lemma in Appendix A.3.3.

Lemma 16. *One can check in polynomial time whether $f = 0$, and otherwise find a point $\mathbf{x}^* \in \mathbb{R}^d$ such that $f(\mathbf{x}^*) \neq 0$.*

Lemma 16 implies that 2-LAYER RELU SURJECTIVITY is polynomially equivalent to 2-LAYER RELU POSITIVITY (if $f(\mathbf{x}^*) > 0$, then replace \mathbf{W}_2 with $-\mathbf{W}_2$ such that $f(\mathbf{x}^*) < 0$) and hence we obtain the following corollary. Moreover,

Corollary 17. *For every $\ell \geq 2$, it holds that ℓ -LAYER RELU SURJECTIVITY is NP-complete.*

Clearly, the above result implies NP-hardness for the general case where the output dimension m is part of the input. It is unclear, however, whether containment in NP also holds for larger output dimension.²

²We believe that the problem might be Π_2^P -complete for $m \geq 2$.

5.4 ZONOTOPE FORMULATION

We conclude this section with an alternative formulation of 2-LAYER RELU POSITIVITY based on a duality of convex piecewise linear functions and polytopes. Interestingly, this yields a close connection to zonotope problems which arise in areas such as robotics and control. Let \mathcal{F}_d be the set of convex piecewise linear functions from \mathbb{R}^d to \mathbb{R} and let \mathcal{P}_d be the set of polytopes in \mathbb{R}^d . For every $f \in \mathcal{F}_d$, there are $\{\mathbf{a}_i \in \mathbb{R}^d\}_{i \in I}$ such that $f(\mathbf{x}) = \max_{i \in I} \{\mathbf{a}_i^T \mathbf{x}\}$ and there is a bijection $\varphi: \mathcal{F}_d \rightarrow \mathcal{P}_d$ given by $\varphi(\max_{i \in I} \{\mathbf{a}_i^T \mathbf{x}\}) = \text{conv} \{\mathbf{a}_i \mid i \in I\}$ where the inverse is the *support function* $\varphi^{-1}: \mathcal{P}_d \rightarrow \mathcal{F}_d$ given by $\varphi^{-1}(P)(\mathbf{x}) = \max\{\langle \mathbf{x}, \mathbf{y} \rangle \mid \mathbf{y} \in P\}$.

Furthermore, φ is a semi-ring isomorphism between the semi-rings $(\mathcal{F}_d, \max, +)$ and $(\mathcal{P}_d, \text{conv}, +)$, where $+$ is either the pointwise addition or the Minkowski sum, respectively.³

A *zonotope* is a Minkowski sum of line segments, i.e., given a matrix $\mathbf{G} \in \mathbb{R}^{n \times d}$, the corresponding zonotope is given by $Z(\mathbf{G}) := \left\{ \mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} \in \sum_{i \in [n]} \text{conv}\{\mathbf{0}, \mathbf{g}_i\} \right\}$.

We now show that 2-LAYER RELU POSITIVITY is equivalent to deciding (non-)containment of certain zonotopes. Given the map $f(\mathbf{x}) = \mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{x})$ (where $\mathbf{W}_2 \in \{-1, 1\}^{1 \times n}$), we define the sets $I^+ := \{i \in [n] \mid (\mathbf{W}_2)_i = 1\}$ and $I^- := [n] \setminus I^+$ and let $\mathbf{W} := \mathbf{W}_1$. Note that we have

$$f(\mathbf{x}) = \sum_{i \in I^+} \max\{0, \mathbf{w}_i \mathbf{x}\} - \sum_{i \in I^-} \max\{0, \mathbf{w}_i \mathbf{x}\},$$

and therefore $f = \varphi^{-1}(Z^+) - \varphi^{-1}(Z^-)$, where $Z^+ := \sum_{i \in I^+} \text{conv}\{\mathbf{0}, \mathbf{w}_i\} = Z(\mathbf{W}_{I^+})$, and $Z^- := \sum_{i \in I^-} \text{conv}\{\mathbf{0}, \mathbf{w}_i\} = Z(\mathbf{W}_{I^-})$. Note that the support functions $\varphi^{-1}(Z^+)$ and $\varphi^{-1}(Z^-)$ can only attain nonnegative values and $Z^+ \subseteq Z^-$ implies that $\varphi^{-1}(Z^+) \leq \varphi^{-1}(Z^-)$. Moreover, if for a $\mathbf{v} \in Z^+$, it holds that $\varphi^{-1}(Z^+)(\mathbf{v}) > \varphi^{-1}(Z^-)(\mathbf{v})$, then $\mathbf{v} \notin Z^-$. Therefore, there exists a $\mathbf{v} \in \mathbb{R}^d$ such that $\varphi^{-1}(Z^+)(\mathbf{v}) > \varphi^{-1}(Z^-)(\mathbf{v})$ if and only if $Z^+ \not\subseteq Z^-$.

Notably, Kulmburg & Althoff (2021) already showed that ZONOTOPE CONTAINMENT (that is, the question whether $Z^+ \subseteq Z^-$) is coNP-hard, which implies our NP-hardness of 2-LAYER RELU POSITIVITY (Theorem 11). Nevertheless, we believe that our reduction is more accessible and direct and provides a different perspective on the computational hardness. The question whether ZONOTOPE CONTAINMENT is fixed-parameter tractable with respect to d is, to the best of our knowledge, open.

6 CONCLUSION

We showed the strongest hardness result for network verification known so far and thereby excluded polynomial-time algorithms (in the worst case) for almost all restrictions on the input set. Moreover, we initiate the complexity-theoretic study of deciding the two elementary properties injectivity and surjectivity for functions computed by ReLU networks. We exclude polynomial-time algorithms for solving both problems, and prove fixed-parameter tractability for injectivity of a single layer. It turned out that surjectivity is a special case of network verification and is also equivalent to zonotope containment. Our results build new bridges between seemingly unrelated areas, and yield new insights into the complexity and expressiveness of ReLU neural networks. We close with some open questions:

- Can the running time for RELU-LAYER INJECTIVITY be improved? Or is it possible to prove a lower bound of $2^{\Omega(d \log d)}$?
- What is the (parameterized) complexity of deciding injectivity for a ReLU neural network with two hidden layers?
- Is 2-LAYER RELU SURJECTIVITY (or equivalently ZONOTOPE CONTAINMENT) fixed-parameter tractable with respect to the input dimension d ?
- How can surjectivity be characterized for output dimension $m \geq 2$? What is the complexity of the decision problem?
- What is the complexity of bijectivity for 2-layer ReLU networks?

³See, e.g., Zhang et al. (2018) for more details on this correspondence.

REFERENCES

- 540
541
542 Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural
543 networks with rectified linear units. In *6th International Conference on Learning Representations,
544 ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
545 OpenReview.net, 2018. URL https://openreview.net/forum?id=BlJ_rgWRW. 3
- 546 Maximilian Baader, Matthew Mirman, and Martin Vechev. Universal approximation with certified
547 networks. In *International Conference on Learning Representations*, 2020. URL [https://
548 openreview.net/forum?id=BlgX8kBTPr](https://openreview.net/forum?id=BlgX8kBTPr). 2, 3
- 549
550 Maximilian Baader, Mark Niklas Mueller, Yuhao Mao, and Martin Vechev. Expressivity of reLU-
551 networks under convex relaxations. In *The Twelfth International Conference on Learning Repre-
552 sentations*, 2024. URL <https://openreview.net/forum?id=awHTL3Hpto>. 2, 3
- 553
554 Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon
555 Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao,
556 and Karol Zieba. End to end learning for self-driving cars, 2016. URL [https://arxiv.org/
557 abs/1604.07316](https://arxiv.org/abs/1604.07316). 1
- 558 Paul S. Bonsma, Hajo Broersma, Viresh Patel, and Artem V. Pyatkin. The complexity status of
559 problems related to sparsest cuts. In *21st International Workshop on Combinatorial Algorithms
560 (IWOC '10)*, volume 6460 of *LNCS*, pp. 125–135. Springer, 2010. 19
- 561
562 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer,
563 2013. 4
- 564
565 Claudio Ferrari, Mark Niklas Mueller, Nikola Jovanović, and Martin Vechev. Complete verification
566 via multi-neuron relaxation guided branch-and-bound. In *International Conference on Learning
567 Representations*, 2022. URL https://openreview.net/forum?id=l_amHf1oaK. 3
- 568
569 Ana Paulina Figueroa, César Hernández-Cruz, and Mika Olsen. The minimum feedback arc set
570 problem and the acyclic disconnection for graphs. *Discrete Mathematics*, 340(7):1514–1521,
571 2017. 5
- 572
573 Vincent Froese and Christoph Hertrich. Training neural networks is NP-hard in fixed dimension. In
574 *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Informa-
575 tion Processing Systems (NeurIPS '23)*, pp. 44039–44049, 2023. 3
- 576
577 Vincent Froese, Christoph Hertrich, and Rolf Niedermeier. The computational complexity of ReLU
578 network training parameterized by data dimensionality. *Journal of Artificial Intelligence Re-
579 search*, 74:1775–1790, 2022. 3
- 580
581 Takashi Furuya, Michael Puthawala, Matti Lassas, and Maarten V. de Hoop. Globally injective and
582 bijective neural operators. In *Advances in Neural Information Processing Systems 36: Annual
583 Conference on Neural Information Processing Systems (NeurIPS '23)*, pp. 57713–57753, 2023.
584 1, 2
- 585
586 Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Ue-
587 sato, Relja Arandjelović, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval
588 bound propagation for training verifiably robust models. *ArXiv*, abs/1810.12715, 2018. URL
589 <https://api.semanticscholar.org/CorpusID:53112003>. 3
- 590
591 J. Elisenda Grigsby and Kathryn Lindsey. On transversality of bent hyperplane arrangements and
592 the topological expressiveness of relu neural networks. *SIAM Journal on Applied Algebra and
593 Geometry*, 6(2):216–242, 2022. doi: 10.1137/20M1368902. URL [https://doi.org/10.
1137/20M1368902](https://doi.org/10.1137/20M1368902). 4
- 594
595 Daniel Haider, Martin Ehler, and Péter Balázs. Convex geometry of ReLU-layers, injectivity on the
596 ball and local reconstruction. In *Proceedings of the 40th International Conference on Machine
597 Learning (ICML '23)*, volume 202 of *PMLR*, pp. 12339–12350, 2023. 1, 2

- 594 Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns. In
595 Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox,
596 and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Con-
597 ference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019,
598 Vancouver, BC, Canada*, pp. 359–368, 2019. URL [https://proceedings.neurips.cc/
599 paper/2019/hash/9766527f2b5d3e95d4a733fcfb77bd7e-Abstract.html](https://proceedings.neurips.cc/paper/2019/hash/9766527f2b5d3e95d4a733fcfb77bd7e-Abstract.html). 4
- 600 Eduard Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht der
601 Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923. 6
- 602
603 Christoph Hertrich, Amitabh Basu, Marco Di Summa, and Martin Skutella. Towards lower bounds
604 on the depth of relu neural networks. In *Advances in Neural Information Processing Systems 34:
605 Annual Conference on Neural Information Processing Systems (NeurIPS’21)*, pp. 3336–3348,
606 2021. 3, 18
- 607 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k -SAT. *Journal of Computer
608 and System Sciences*, 62(2):367–375, 2001. 5, 14
- 609
610 Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. Deep neural network compres-
611 sion for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics*,
612 42(3):598–608, 2019. doi: 10.2514/1.G003724. URL [https://doi.org/10.2514/1.
613 G003724](https://doi.org/10.2514/1.G003724). 1
- 614 Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: a
615 calculus for reasoning about deep neural networks. *Formal Methods in System Design*, 60(1):
616 87–116, 2022. 1, 2
- 617
618 Panagiotis Kouvaros and Alessio Lomuscio. Towards scalable complete verification of ReLU neural
619 networks via dependency-based branching. In *Proceedings of the Thirtieth International Joint
620 Conference on Artificial Intelligence (IJCAI ’21)*, pp. 2643–2650, 2021. 1
- 621 Dieter Kratsch and Van Bang Le. Algorithms solving the matching cut problem. *Theoretical Com-
622 puter Science*, 609:328–335, 2016. 14
- 623
624 Adrian Kulmburg and Matthias Althoff. On the co-NP-completeness of the zonotope containment
625 problem. *European Journal of Control*, 62:84–91, 2021. 2, 10
- 626
627 László Lovász. Coverings and colorings of hypergraphs. In *Proceedings of the 4th Southeastern
628 Conference of Combinatorics, Graph Theory, and Computing*, pp. 3–12, 1973. 13
- 629 Yuhao Mao, Mark Niklas Müller, Marc Fischer, and Martin Vechev. Understanding certified training
630 with interval bound propagation, 2024. URL <https://arxiv.org/abs/2306.10426>. 3
- 631 Matthew B Mirman, Maximilian Baader, and Martin Vechev. The fundamental limits of neural
632 networks for interval certified robustness. *Transactions on Machine Learning Research*, 2022.
633 ISSN 2835-8856. URL <https://openreview.net/forum?id=fsacLLU35V>. 3
- 634
635 Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. On the parameterized complexity of girth and
636 connectivity problems on linear matroids. In *Proceedings of the 14th International Symposium
637 on Algorithms and Data Structures (WADS ’15)*, volume 9214 of *LNCS*, pp. 566–577. Springer,
638 2015. 2
- 639 Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanic, and Maarten V. de Hoop. Globally
640 injective ReLU networks. *Journal of Machine Learning Research*, 23:105:1–105:55, 2022. 1, 2,
641 5
- 642
643 Ansgar Rössig and Milena Petkovic. Advances in verification of ReLU neural networks. *Journal of
644 Global Optimization*, 81(1):109–152, 2021. 1
- 645
646 Marco Sälzer and Martin Lange. Reachability in simple neural networks, 2023. URL <https://arxiv.org/abs/2203.07941>. 1, 2, 4
- 647
A. Schrijver. *Theory of Linear and Integer programming*. Wiley-Interscience, 1986. 3, 16

- 648 Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying
649 neural networks. *Proc. ACM Program. Lang.*, 3(POPL), January 2019. doi: 10.1145/3290354.
650 URL <https://doi.org/10.1145/3290354>. 3
- 651 Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Good-
652 fellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and
653 Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014,
654 Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL [http:
655 //arxiv.org/abs/1312.6199](http://arxiv.org/abs/1312.6199). 1
- 656 Andreas M. Tillmann and Marc E. Pfetsch. The computational complexity of the restricted isometry
657 property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions
658 on Information Theory*, 60(2):1248–1259, 2014. 2
- 659 Zi Wang, Aws Albarghouthi, Gautam Prakriya, and Somesh Jha. Interval universal approximation
660 for neural networks. *Proc. ACM Program. Lang.*, 6(POPL), January 2022. doi: 10.1145/3498675.
661 URL <https://doi.org/10.1145/3498675>. 2, 3
- 662 Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane S.
663 Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for ReLU net-
664 works. In *Proceedings of the 35th International Conference on Machine Learning (ICML’18)*,
665 volume 80 of *PMLR*, pp. 5273–5282, 2018. 1, 3
- 666 Eric Wong, Frank R. Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial
667 defenses. In *Proceedings of the 32nd International Conference on Neural Information Processing
668 Systems, NIPS’18*, pp. 8410–8419, Red Hook, NY, USA, 2018. Curran Associates Inc. 3
- 669 Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya
670 Kaikhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified ro-
671 bustness and beyond. Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
672 3
- 673 Thomas Zaslavsky. Facing up to arrangements: face-count formulas for partitions of space by
674 hyperplanes. *Memoirs of the American Mathematical Society*, 1(154), 1975. 5
- 675 Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter.
676 General cutting planes for bound-propagation-based neural network verification. In *Proceedings
677 of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red
678 Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088. 3
- 679 Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In
680 *Proceedings of the 35th International Conference on Machine Learning (ICML’18)*, volume 80
681 of *PMLR*, pp. 5824–5832, 2018. 10

686 A APPENDIX

687 A.1 APPENDIX TO SECTION 3

688 To prove that ACYCLIC 2-DISCONNECTION is NP-hard, we reduce from the following problem.

690 3-UNIFORM HYPERGRAPH 2-COLORING

691 **Input:** A 3-uniform hypergraph $H = (V, E)$, that is, $|e| = 3$ for all $e \in E$.

692 **Question:** Is there a 2-coloring of the nodes V such that no hyperedge is monochromatic?

693 **Theorem 18.** ACYCLIC 2-DISCONNECTION is NP-hard.

694 *Proof.* We give a reduction from 3-UNIFORM HYPERGRAPH 2-COLORING which is NP-
695 hard Lovász (1973). Let $H = (V, E)$ be a 3-uniform hypergraph with $|V| = n$ and $|E| = m$.
696 We construct a digraph $D = (U, A)$ as follows: For each $i \in \{0, 1\}$, we define a node set U_i with
697 the $n + 2m$ nodes

$$698 U_i := \{v_i \mid v \in V\} \cup \bigcup_{e \in E} \{e_i, e'_i\}.$$

Moreover, for each node $v \in V$, we define a node set X_v with the $2 \deg(v) + 2$ nodes

$$X_v := \{x_v, x'_v\} \cup \{x_{v,e,0}, x_{v,e,1} \mid e \in E, v \in e\}.$$

Finally, we define the node set $Q := \{q_{e,i}, q'_{e,i} \mid e \in E, i \in \{0, 1\}\}$ and let

$$U := U_0 \cup U_1 \cup \bigcup_{v \in V} X_v \cup Q.$$

The arc set A is defined as follows: For $i \in \{0, 1\}$, we connect the nodes in U_i with $2(|U_i| - 1)$ arcs to a strongly connected path, that is, for an arbitrary ordering $b_1, \dots, b_{|U_i|}$ of the nodes in U_i , we insert the arcs (b_j, b_{j+1}) and (b_{j+1}, b_j) for each $j \in [|U_i| - 1]$. Analogously, we also connect all nodes in X_v for each $v \in V$ to a strongly connected path. Moreover, for each $v \in V$, we insert the cyclic arcs (v_0, x_v) , (x_v, v_1) , (v_1, x'_v) , and (x'_v, v_0) . Finally, for each hyperedge $e = \{u, v, w\} \in E$ and each $i \in \{0, 1\}$, we insert the arcs

- $(q_{e,i}, q'_{e,i})$ and $(q'_{e,i}, q_{e,i})$,
- $(x_{u,e,i}, e_i)$, $(e_i, q_{e,i})$, $(q_{e,i}, e'_i)$, and $(e'_i, x_{u,e,i})$, and
- $(x_{v,e,i}, q_{e,i})$, $(q_{e,i}, x_{w,e,i})$, $(x_{w,e,i}, q'_{e,i})$, and $(q'_{e,i}, x_{v,e,i})$.

Overall, the constructed digraph D contains $O(n + m)$ many nodes and arcs.

For the correctness, assume first that there is a 2-coloring of the nodes of H such that no hyperedge is monochromatic and let $V_i \subseteq V$ denote the set of nodes with color i . We construct a solution $A' \subseteq A$ for D as follows: For each $v \in V_0$, A' contains the arcs (x_v, v_1) and (v_1, x'_v) , and for each $v \in V_1$, A' contains the arcs (v_0, x_v) and (x'_v, v_0) . Clearly, these arcs are acyclic. Further, consider a hyperedge $e = \{u, v, w\} \in E$. Since e is not monochromatic, it follows that exactly one of its nodes is colored with one of the two colors, say 0 (the other case is analogous), and the two other nodes have color 1. If u is colored 0, then A' contains the arcs $(x_{u,e,1}, e_1)$, $(e'_1, x_{u,e,1})$, $(e_0, q_{e,0})$, and $(q_{e,0}, e'_0)$. If v has color 0 (the case where w has color 0 is analogous), then A' contains the arcs $(x_{u,e,0}, e_0)$, $(e'_0, x_{u,e,0})$, $(x_{v,e,1}, q_{e,1})$, $(q_{e,1}, x_{v,e,1})$, $(x_{w,e,0}, q'_{e,0})$, and $(q_{e,0}, x_{w,e,0})$. It can easily be verified that A' is acyclic. Moreover, $(U, A \setminus A')$ is not weakly connected since e.g. all nodes in X_v with $v \in V_0$ are disconnected from all nodes in U_1 .

Conversely, assume that there is a solution $A' \subseteq A$ for D . First, observe that in $D' := (U, A \setminus A')$ all nodes in U_0 are weakly connected (since they are strongly connected in D). The same holds for U_1 and for each X_v , $v \in V$. Moreover, for each $v \in V$, the set X_v is weakly connected to exactly one of the sets U_0 or U_1 . To see this, note that X_v cannot be disconnected from both U_0 and U_1 due to the cycle involving the nodes x_v and x'_v . It follows that also no X_v can be weakly connected to both U_0 and U_1 since then D' would be weakly connected because also each node in Q is connected to some X_v due to its cyclic connections. Hence, we assign each node $v \in V$ the color $i \in \{0, 1\}$ if and only if X_v is weakly connected to U_i .

It remains to check that each hyperedge $e = \{u, v, w\} \in E$ is not monochromatic. Assume the contrary, that is, X_u , X_v , and X_w are weakly connected to (wlog) U_0 . Then, by construction, also $q_{e,1}$ and $q'_{e,1}$ are weakly connected to U_0 . Since X_u is not weakly connected to U_1 and therefore A' contains $(x_{u,e,1}, e_1)$ and $(e'_1, x_{u,e,1})$, it follows that e_1 is weakly connected to $q_{e,1}$, since otherwise A' would not be acyclic. Therefore, e_1 is also weakly connected to U_0 , which yields a contradiction since then D' would be weakly connected. \square

We remark that our reduction implies a running time lower bound based on the Exponential Time Hypothesis⁴ (ETH). As discussed by Kratsch & Le (2016), there is no algorithm solving a 3-UNIFORM HYPERGRAPH 2-COLORING-instance (V, E) in $2^{o(|E|)}$ time assuming ETH. Our polynomial-time reduction in the proof of Theorem 18 constructs a digraph D of size $O(|V| + |E|)$. Notice that $O(|V| + |E|) \subseteq O(|E|)$ since we can assume $|V| \leq 3|E|$ (isolated nodes can trivially be removed). Hence, any algorithm solving ACYCLIC 2-DISCONNECTION in $2^{o(|D|)}$ time would imply a $2^{o(|E|)}$ -time algorithm for 3-UNIFORM HYPERGRAPH 2-COLORING.

Corollary 19. ACYCLIC 2-DISCONNECTION cannot be solved in $2^{o(|D|)}$ time unless the ETH fails.

⁴The Exponential Time Hypothesis asserts that 3-SAT cannot be solved in $2^{o(n)}$ time where n is the number of Boolean variables in the input formula (Impagliazzo & Paturi (2001)).

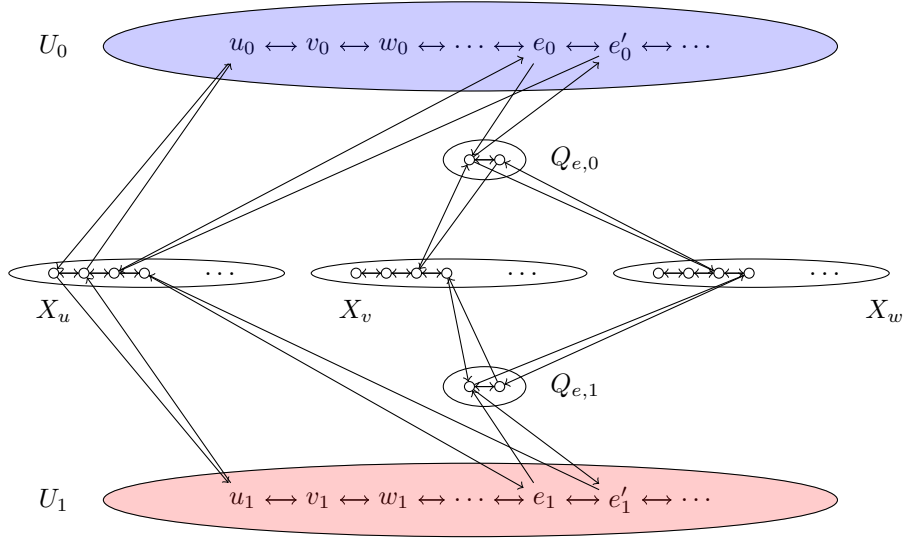


Figure 3: The encoding of one hyperedge $\{u, v, w\} \in E$ in the digraph D . The cyclic directed paths connecting X_v respectively X_w with U_0 and U_1 are not drawn in order to not overload the figure.

A.1.1 PROOF OF THEOREM 6

Proof. Containment in NP is easy: The set I_C of a cell C with $\text{rank}(\mathbf{W}_C) < d$ serves as a certificate.

For the NP-hardness, we reduce from ACYCLIC 2-DISCONNECTION which is NP-hard by Theorem 18. For a given digraph $D = (V = \{v_1, \dots, v_n\}, A = \{a_1, \dots, a_m\})$, we construct the matrix $\mathbf{W} \in \mathbb{R}^{m \times (n-1)}$ as follows: For every arc $a = (v_i, v_j) \in A$, we add the row vector $\mathbf{w}_a \in \mathbb{R}^{n-1}$ with

$$(\mathbf{w}_a)_\ell = \begin{cases} 1, & \ell = i \\ -1, & \ell = j \\ 0, & \text{otherwise} \end{cases}.$$

For the correctness, assume first that there is a solution $A' \subseteq A$ for D . Let $I' := \{i \mid a_i \in A'\} \subseteq [m]$. We claim that there is a cell $C \in \mathcal{C}_{\mathbf{W}}$ with $I_C \subseteq [m] \setminus I'$. To see this, let $v_{\pi_1}, v_{\pi_2}, \dots, v_{\pi_n}$ be a topological ordering of V such that all arcs in A' point “from left to right”. Such an ordering exists since A' is acyclic. Let $\pi_k = n$ and let $\mathbf{x} \in \mathbb{R}^{n-1}$ be such that

$$\mathbf{x}_{\pi_1} < \dots < \mathbf{x}_{\pi_{k-1}} < 0 < \mathbf{x}_{\pi_{k+1}} < \dots < \mathbf{x}_{\pi_n}.$$

Then, no neuron $i \in I'$ corresponding to an arc $a_i = (v_j, v_\ell) \in A'$ is active at \mathbf{x} since

$$\mathbf{w}_{a_i} \mathbf{x} = \begin{cases} \mathbf{x}_j - \mathbf{x}_\ell, & j \neq n \wedge \ell \neq n \\ -\mathbf{x}_\ell, & j = n \\ \mathbf{x}_j, & \ell = n \end{cases} < 0.$$

Hence, \mathbf{x} is contained in some cell C with $I_C \subseteq [m] \setminus I'$. Now, since $(V, A \setminus A')$ is not weakly connected, it follows that $\text{rank}(\mathbf{W}_C) < n - 1$. To see this, note that there must be a node $v_i \in V$ that is not weakly connected to v_n , that is, there is no undirected path from v_i to v_n . If $\text{rank}(\mathbf{W}_C) = n - 1$, then there exists a linear combination $\sum_{j \in I_C} c_j \mathbf{w}_{a_j} = \mathbf{e}_i$ of the i -th unit vector in \mathbb{R}^{n-1} . But this implies the existence of an undirected path from v_i to v_n corresponding to some arcs in $\{a_j \mid c_j \neq 0\}$, which yields a contradiction. To see this, note first that $(\mathbf{e}_i)_i = 1$ implies that $c_j \neq 0$ for some $j \in I_C$ such that $(\mathbf{w}_{a_j})_i \neq 0$. Clearly, $a_j = (v_i, v_n)$ or $a_j = (v_n, v_i)$ is not possible. Hence, a_j must be an arc between v_i and some $v_k \neq v_n$. But then, we have $(c_j \mathbf{w}_{a_j})_k \neq 0$, whereas $(\mathbf{e}_i)_k = 0$. Therefore, there exists another $j' \in I_C$ such that $c_{j'} \neq 0$ and $(\mathbf{w}_{a_{j'}})_k \neq 0$. Again, it is not possible that $a_{j'} = (v_k, v_n)$ or $a_{j'} = (v_n, v_k)$. However, since I_C is finite, repeating this argument yields a contradiction.

For the reverse direction, let $P \in \mathcal{C}_{\mathbf{W}}$ be a cell with $\text{rank}(\mathbf{W}_C) < n - 1$. Then, there exists a point $\mathbf{x} \in P$. Now, let $i \in [m] \setminus I_C$ be a neuron corresponding to arc $a_i = (v_j, v_\ell)$ that is not active

at \mathbf{x} . Then, this implies that $\mathbf{x}_j - \mathbf{x}_\ell < 0$ if $j < n$ and $\ell < n$. If $j = n$, then this implies $\mathbf{x}_\ell > 0$ and if $\ell = n$, then this implies $\mathbf{x}_j < 0$. Hence, (V, A') with $A' := \{a_i \mid i \in [m] \setminus I_C\}$ is acyclic since any cycle would lead to a contradiction. Now, we claim that $(V, A \setminus A')$ cannot be weakly connected. Otherwise, there exists an undirected path from each v_i , $i \in [n-1]$ to v_n . Let $v_i = v_{j_0}, v_{j_1}, \dots, v_{j_t} = v_n$ be the nodes of such a path along the arcs $a_{\ell_1}, \dots, a_{\ell_t}$. Then, there exists the linear combination $\sum_{k=1}^t c_k \mathbf{w}_{a_{\ell_k}} = \mathbf{e}_i$, where $c_k := 1$ if $a_{\ell_k} = (v_{j_{k-1}}, v_{j_k})$ and $c_k := -1$ if $a_{\ell_k} = (v_{j_k}, v_{j_{k-1}})$. But this implies $\text{rank}(\mathbf{W}_C) = n-1$, which yields a contradiction. \square

A.2 APPENDIX TO SECTION 4

A.2.1 PROOF OF LEMMA 9

Proof. We use strong duality of linear programming. To that end, let $\mathbf{A} \in \mathbb{R}^{m \times d}$ such that $C = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\}$ and let $\mathbf{W} := (\mathbf{w}_1, \dots, \mathbf{w}_n)^T \in \mathbb{R}^{n \times d}$. Since $C \subseteq \bigcup_{i=1}^n H_{\mathbf{w}_i}^+$, it follows that the set

$$\{\mathbf{x} \in \mathbb{R}^d \mid -\mathbf{W}\mathbf{x} \geq \mathbf{1}, \mathbf{A}\mathbf{x} \geq \mathbf{0}\} \subseteq \{\mathbf{x} \in \mathbb{R}^d \mid -\mathbf{W}\mathbf{x} > \mathbf{0}, \mathbf{A}\mathbf{x} \geq \mathbf{0}\}$$

is empty and hence the following linear program does not admit a feasible solution.

$$\begin{aligned} \min_{\mathbf{x}} \quad & 0 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{0} \\ & -\mathbf{W}\mathbf{x} \geq \mathbf{1} \end{aligned}$$

By strong duality, the dual linear program

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{1}^T \mathbf{y} \\ \text{s.t.} \quad & \mathbf{z}^T \mathbf{A} - \mathbf{y}^T \mathbf{W} = \mathbf{0} \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0} \end{aligned}$$

has either no feasible solution or its objective value is unbounded. Since $\mathbf{y} = \mathbf{0}$ and $\mathbf{z} = \mathbf{0}$ yields a feasible solution, the latter is the case. In particular, there is a ray ρ of the cone $\{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n+m} \mid \mathbf{z}^T \mathbf{A} - \mathbf{y}^T \mathbf{W} = \mathbf{0}, \mathbf{y}, \mathbf{z} \geq \mathbf{0}\}$ such that the objective value is unbounded on ρ . The dimension of the subspace $\{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n+m} \mid \mathbf{z}^T \mathbf{A} - \mathbf{y}^T \mathbf{W} = \mathbf{0}\}$ is at least $n+m-d$. Therefore, ρ as a 1-dimensional subspace lies in the intersection of at least $n+m-d-1$ many hyperplanes of the form $\{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n+m} \mid (\mathbf{y}, \mathbf{z})_i = 0\}$ and hence in the intersection of at least $n-d-1$ many hyperplanes of the form $\{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n+m} \mid \mathbf{y}_i = 0\}$. Let $B \subseteq [n]$ be the set of size at least $n-d-1$ such that $\rho \subseteq \bigcap_{i \in B} \{(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{n+m} \mid \mathbf{y}_i = 0\}$ (this can be computed in polynomial time; see, e.g., (Schrijver, 1986, Corollary 14.1g)) and $A := [n] \setminus B$ its complement.

It follows that $|A| \leq d+1$ and that the objective value of the following LP is still unbounded.

$$\begin{aligned} \max_{\mathbf{y}} \quad & \mathbf{1}^T \mathbf{y} \\ \text{s.t.} \quad & \mathbf{z}^T \mathbf{A} - \mathbf{y}_A^T \mathbf{W}_A = \mathbf{0} \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0} \end{aligned}$$

Again, by strong duality, this implies that the set

$$S := \{\mathbf{x} \in \mathbb{R}^d \mid -\mathbf{W}_A \mathbf{x} \geq \mathbf{1}, \mathbf{A}\mathbf{x} \geq \mathbf{0}\}$$

is empty. Assume that the cone $\{\mathbf{x} \in \mathbb{R}^d \mid -\mathbf{W}_A \mathbf{x} > \mathbf{0}, \mathbf{A}\mathbf{x} \geq \mathbf{0}\}$ contains an element \mathbf{x} . Let $k := \min_{i \in [d]} \{(-\mathbf{W}_A \mathbf{x})_i\}$. Then $\frac{1}{k} \cdot \mathbf{x} \in S$, which is a contradiction. Hence, the set

$$\{\mathbf{x} \in \mathbb{R}^d \mid -\mathbf{W}_A \mathbf{x} > \mathbf{0}, \mathbf{A}\mathbf{x} \geq \mathbf{0}\}$$

is empty, which means that $C \subseteq \bigcup_{i \in A} H_{\mathbf{w}_i}^+$, proving the claim. \square

A.2.2 PROOF OF THEOREM 10

Lemma 20. *Algorithm 1 is correct.*

864 *Proof.* Assume that the algorithm outputs some $\mathbf{x} \in \mathbb{R}^d$. If \mathbf{x} was returned in Line 1 or Line 2, then
 865 this is clearly correct since either no cell has rank d or there is a rank-0 cell. If \mathbf{x} was returned by
 866 some call of FindCell (Algorithm 2) in Line 4, then this is correct since $\text{rank}(W_{\mathbf{x}}) \leq \text{rank}(C) < d$,
 867 where $W_{\mathbf{x}} := \{\mathbf{w}_i \in W \mid \mathbf{w}_i^T \mathbf{x} \geq 0\}$. The second inequality holds since otherwise the algorithm
 868 would have returned “No” in Line 1. For the first inequality, we first observe the invariant that
 869 $W \setminus \text{span}(C) \subseteq M \subseteq W$ holds at any time during execution of FindCell. This is clear for the initial
 870 call of FindCell in Line 3 in Algorithm 1. Also, within FindCell the property holds after Line 3 and
 871 for the recursive calls in Line 7. Now, $\text{rank}(W_{\mathbf{x}}) \leq \text{rank}(C)$ holds since $\mathbf{m}_i^T \mathbf{x} < 0$ for all $i \in [m]$
 872 implies that $W_{\mathbf{x}} \cap M = \emptyset$, and thus, by our invariant, we have $C \subseteq W_{\mathbf{x}} \subseteq \text{span}(C)$.

873 For the opposite direction, assume that there is an $\mathbf{x} \in \mathbb{R}^d$ such that $k := \text{rank}(W_{\mathbf{x}}) < d$. If $k = 0$,
 874 then Algorithm 1 correctly returns some point of a rank-0 cell in Line 2. If $k > 0$, then $W_{\mathbf{x}}$ contains
 875 at least one vector \mathbf{w}_i . We claim that the call of FindCell in Line 3 will correctly return some point
 876 from a cell with rank at most k . To this end, we show that FindCell(C, M) (Algorithm 2) returns a
 877 correct point whenever $C \subseteq W_{\mathbf{x}}$. Clearly, if $\text{rank}(C) = k$, then \mathbf{x} satisfies the conditions in Line 4
 878 since $C \subseteq W_{\mathbf{x}}$, and thus $W_{\mathbf{x}} \subseteq \text{span}(C)$ while $M \subseteq W \setminus \text{span}(C)$ (due to Line 3 and the invariant
 879 on M). Hence, FindCell returns a correct point in this case.

880 Now consider the case $\text{rank}(C) < k$. If some point is returned in Line 4, then this is correct
 881 (as already shown above). If no point satisfies the conditions in Line 4, then it holds $\{\mathbf{x} \mid \forall i \in$
 882 $[n] : \mathbf{c}_i^T \mathbf{x} \geq 0\} \subseteq \bigcup_{i \in I} H_{\mathbf{m}_i}^+$ and by Lemma 9, we can compute the set A in Line 5. Note that
 883 $W_{\mathbf{x}} \cap \{\mathbf{m}_i \mid i \in A\} \neq \emptyset$. Hence, for at least one of the recursive calls in Line 7, it holds that
 884 $C \cup \{\mathbf{m}_i\} \subseteq W_{\mathbf{x}}$. Moreover, $\text{rank}(C \cup \{\mathbf{m}_i\}) = \text{rank}(C) + 1$ since $\mathbf{m}_i \notin \text{span}(C)$ (due to Line 3).
 885 Hence, by induction, this call will return a correct point. \square

886 **Lemma 21.** *Algorithm 1 runs in $O((d+1)^d \cdot \text{poly}(S))$ time, where S denotes the input size.*

888 *Proof.* Let S be the bit-length of W . Clearly, Line 1 and Line 2 can be done in $\text{poly}(S)$ time via
 889 linear programming. As regards the running time of FindCell, note first that the recursion depth is
 890 at most d since every recursive call increases the rank of C (as already discussed in the proof of
 891 Lemma 21) and the recursion terminates when rank d is reached. Moreover, each call of FindCell
 892 branches into at most $d+1$ recursive calls, that is, the search tree has size at most $(d+1)^d$. Since
 893 all other computations within FindCell can be done in $\text{poly}(S)$ time (using linear programming and
 894 Lemma 9), we obtain the desired running time. \square

895 A.3 APPENDIX TO SECTION 5

897 A.3.1 PROOF OF THEOREM 12

899 *Proof.* We reduce from the complement of 2-LAYER RELU POSITIVITY. Let $\mathbf{W}_1 \in \mathbb{R}^{n \times k}$, $\mathbf{W}_2 \in$
 900 $\mathbb{R}^{1 \times n}$ be an instance of 2-LAYER RELU POSITIVITY. Let the dimension d , the affine space A (as
 901 a basis) and the point $\mathbf{z} \in \mathbb{R}^d$ be the output of the algorithm (on input k) that exists due to the fact
 902 that $(S_d)_{d \in \mathbb{N}}$ is a reasonable sequence of sets. More precisely, let $\varepsilon > 0$ be chosen such that

$$903 B = B_\varepsilon(\mathbf{z}) := \{\mathbf{x} \in A \mid \|\mathbf{x} - \mathbf{z}\|_2 < \varepsilon\} \subseteq S_d.$$

904 Let $P: \mathbb{R}^d \rightarrow A$ be the orthogonal projection to A and $T: A \rightarrow \mathbb{R}^k$ an isometric isomorphism
 905 obtained by mapping the normalized basis of A to the standard basis of \mathbb{R}^k . The composition
 906 $(T \circ P): \mathbb{R}^d \rightarrow \mathbb{R}^k$ is an affine map and let it be given by a matrix $\mathbf{A} \in \mathbb{R}^{k \times d}$ and a vector $\mathbf{b} \in \mathbb{R}^k$.
 907 Then, $\widetilde{\mathbf{W}}_2 := \mathbf{W}_2$, $\widetilde{\mathbf{W}}_1 := \mathbf{W}_1 \cdot \mathbf{A}$, $\widetilde{\mathbf{b}}_1 := \mathbf{W}_1 \cdot \mathbf{b} - \mathbf{W}_1 \cdot \mathbf{A} \cdot \mathbf{z}$, $\widetilde{b}_2 := t$ form an instance of
 908 2-LAYER RELU (S, t)-VERIFICATION and it holds that

$$909 \mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{A}\mathbf{x} + \mathbf{b}) > 0 \iff \widetilde{\mathbf{W}}_2 \cdot \phi_{\widetilde{\mathbf{W}}_1, \widetilde{\mathbf{b}}_1}(\mathbf{z} + \mathbf{x}) + \widetilde{b}_2 > t. \quad (1)$$

911 For the correctness of the reduction, assume that there is a $\mathbf{y} \in \mathbb{R}^k$ such that $\mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{y}) > 0$.
 912 Then, by the positive homogeneity of the map $\mathbf{x} \mapsto \mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{x})$, we also have that $\mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{y}') >$
 913 0 for $\mathbf{y}' := \frac{\varepsilon}{2} \frac{\mathbf{y}}{\|\mathbf{y}\|}$. Since T is an isometric isomorphism, there is an $\mathbf{x} \in A$ with $\|\mathbf{x}\| = \frac{\varepsilon}{2}$ such that
 914 $T(\mathbf{x}) = \mathbf{y}'$. Hence, $\mathbf{z} + \mathbf{x} \in B \subseteq S_d$ and Equation (1) implies that $\widetilde{\mathbf{W}}_2 \cdot \phi_{\widetilde{\mathbf{W}}_1, \widetilde{\mathbf{b}}_1}(\mathbf{z} + \mathbf{x}) + \widetilde{b}_2 > t$.
 915

916 Conversely, if there is an $\mathbf{x} \in S_d$ such that $\widetilde{\mathbf{W}}_2 \cdot \phi_{\widetilde{\mathbf{W}}_1, \widetilde{\mathbf{b}}_1}(\mathbf{x}) + \widetilde{b}_2 > t$, then Equation (1) implies that
 917 $\mathbf{W}_2 \cdot \phi_{\mathbf{W}_1}(\mathbf{A}(\mathbf{x} - \mathbf{z}) + \mathbf{b}) > 0$, concluding the proof. \square

A.3.2 PROOF OF LEMMA 14

Proof. a) If there is a $\mathbf{v}^+ \in \mathbb{R}^d$ with $f_0(\mathbf{v}^+) = a > 0$, then for all $b \in [0, \infty)$ it holds that $f_0(\frac{b}{a}\mathbf{v}^+) = b$ due to positive homogeneity of f_0 (analogously for all $b \in (-\infty, 0]$ with \mathbf{v}^-). The other direction is trivial.

b) We start with some preliminary observations. First, there is a constant $C \in \mathbb{R}$ that only depends on the weights and biases of f such that $\|f - f_0\|_\infty \leq C$ (Hertrich et al. (2021) Proposition 2.3). Moreover, due to continuity of f it holds that f is surjective if and only if for every $a < b \in \mathbb{R}$ there are $a', b' \in \mathbb{R}$ with $a' < a$ and $b' > b$ such that $a', b' \in f(\mathbb{R}^d)$.

Now, for the first direction, assume that f is surjective. Then there are $a < -C$ and $b > C$ such that $a, b \in f(\mathbb{R}^d)$. Since $\|f - f_0\|_\infty \leq C$, it follows that there are \mathbf{v}^+ and \mathbf{v}^- with $\phi_0(\mathbf{v}^+) > 0$ and $\phi_0(\mathbf{v}^-) < 0$, implying with a) that ϕ_0 is surjective.

For the converse direction, let $a < 0 < b$. By surjectivity of f_0 , we have that $a - 2C \in f_0(\mathbb{R}^d)$ and $b + 2C \in f_0(\mathbb{R}^d)$. Hence, since $\|f - f_0\|_\infty \leq C$, it follows that there are $a' < a$ and $b' > b$ such that $a', b' \in f(\mathbb{R}^d)$, implying surjectivity of f .

c) Follows directly from a) and Observation 3. \square

A.3.3 PROOF OF LEMMA 16

Proof. We define the sets $I^+ := \{i \in [n] \mid (\mathbf{W}_2)_i = 1\}$ and $I^- := [n] \setminus I^+$ and let $\mathbf{w}_1, \dots, \mathbf{w}_n$ be the rows of \mathbf{W}_1 .

First, we can assume that for any $\mathbf{v} \in \mathbb{R}^d$ there is at most one $i \in [n]$ such that $\mathbf{w}_i \in \text{pos}(\mathbf{v}) := \{\lambda \mathbf{v} \mid \lambda \geq 0\}$. To see this, let $\mathbf{w}_i \in \text{pos}(\mathbf{w}_j)$ for some $i, j \in [n]$. If $i, j \in I^+$, then we can simply delete the rows \mathbf{w}_i and \mathbf{w}_j and add a new row $\mathbf{w}_i + \mathbf{w}_j$ without changing the map f (clearly the same works for $i, j \in I^-$). If $i \in I^+$ and $j \in I^-$, then we can delete the rows \mathbf{w}_i and \mathbf{w}_j and, if $\|\mathbf{w}_j\|_2 \leq \|\mathbf{w}_i\|_2$, add a new row $\mathbf{w}_i - \mathbf{w}_j$ with output weight 1 or add a new row $\mathbf{w}_j - \mathbf{w}_i$ with output weight -1 if $\|\mathbf{w}_i\|_2 \leq \|\mathbf{w}_j\|_2$ without changing the map f . Note that we can transform any matrix \mathbf{W}_1 to such a form in polynomial time.

Now, if for every row \mathbf{w}_j there is a row $\mathbf{w}_{j'}$ such that $\mathbf{w}_j = -\mathbf{w}_{j'}$ and $(\mathbf{W}_2)_j = -(\mathbf{W}_2)_{j'}$, it follows that f is a linear map and hence we can easily check whether it is the zero map. Otherwise, assume that for a row \mathbf{w}_j there is no such row $\mathbf{w}_{j'}$. Then \mathbf{w}_j induces a hyperplane $H_j := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}_j \mathbf{x} = 0\}$ such that ϕ is not linear in every open neighborhood of any $\mathbf{x} \in H_j$ and hence the map f is not linear and in particular cannot be the zero map. Thus, we can check in polynomial time whether $f = 0$.

Now in the case of $f \neq 0$, let $I_j := \{i \in [n] \mid \mathbf{w}_i \in \text{span}(\mathbf{w}_j)\}$ and note that $|I_j| \leq 2$. For $i \in [n]$, we define the hyperplane $H_i := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{w}_i \mathbf{x} = 0\}$. By definition, there exists an $\mathbf{x} \in H_j \setminus \left(\bigcup_{i \in [n] \setminus I_j} H_i\right)$. Now, for

$$\varepsilon := \min\{1, 1/2 \cdot \min_{i \in [n] \setminus I_j} \min_{\mathbf{y} \in H_i} \|\mathbf{x} - \mathbf{y}\|_2 > 0\}$$

it holds that $\{\mathbf{x} + \delta \mathbf{w}_j, \mathbf{x} - \delta \mathbf{w}_j\} \subset \mathbb{R}^d \setminus \left(\bigcup_{i \in [n] \setminus I_j} H_i\right)$ for all $\delta \in (0, \varepsilon)$. Let $\mathbf{x}' := \mathbf{x} + \varepsilon \mathbf{w}_j$ and $\mathbf{x}'' := \mathbf{x} - \varepsilon \mathbf{w}_j$ and let $I' := \{i \in [n] \mid \mathbf{w}_i \mathbf{x}' > 0\}$ and $I'' := \{i \in [n] \mid \mathbf{w}_i \mathbf{x}'' > 0\}$.

We will argue now that either in the cell $C' \in \Sigma_{\mathbf{W}_1}$ containing \mathbf{x}' or in the cell $C'' \in \Sigma_{\mathbf{W}_1}$ containing \mathbf{x}'' we find the desired \mathbf{x}^* with $f(\mathbf{x}^*) \neq 0$. Note that it is sufficient to prove that f cannot be the zero map on C' and C'' . We prove this by showing that

$$\mathbf{W}_2 \circ (\mathbf{W}_1)_{C'} - \mathbf{W}_2 \circ (\mathbf{W}_1)_{C''} \neq \mathbf{0}.$$

Note that $(I' \cup I_j) \setminus \{j\} = I''$. If $I_j = \{j\}$, then

$$\left(\sum_{i \in I' \cap I^+} \mathbf{w}_i - \sum_{i \in I' \cap I^-} \mathbf{w}_i \right) - \left(\sum_{i \in I'' \cap I^+} \mathbf{w}_i - \sum_{i \in I'' \cap I^-} \mathbf{w}_i \right) = \pm \mathbf{w}_j \neq \mathbf{0}.$$

972 If $I_j = \{j, j'\}$, then we have $\mathbf{w}_{j'} = -\lambda \mathbf{w}_j$ for some $\lambda > 0$ and hence

$$973 \left(\sum_{i \in I' \cap I^+} \mathbf{w}_i - \sum_{i \in I' \cap I^-} \mathbf{w}_i \right) - \left(\sum_{i \in I'' \cap I^+} \mathbf{w}_i - \sum_{i \in I'' \cap I^-} \mathbf{w}_i \right) = \pm \mathbf{w}_j \pm \mathbf{w}_{j'}$$

974 equals $\mathbf{0}$ if and only if $\lambda = 1$ and $(\mathbf{W}_2)_j = -(\mathbf{W}_2)_{j'}$, which we assumed not to be the case. \square

975 A.3.4 PROPOSITION 22

976 In order to prove that POSITIVE CUT is NP-complete, we reduce from the following NP-hard problem (Bonsma et al. (2010)).

977 DENSEST CUT

978 **Input:** A graph $G = (V, E)$ and $t \in \mathbb{Q} \cap [0, 1]$.

979 **Question:** Is there a subset $S \subseteq V$ such that $\frac{|E(S, V \setminus S)|}{|S| \cdot |V \setminus S|} > t$?

980 **Proposition 22.** POSITIVE CUT is NP-complete.

981 *Proof.* We reduce from DENSEST CUT. Let (V, E) be a graph and $t = \frac{a}{b}$ with $a < b \in \mathbb{N}$ and let $w(F) := \sum_{e \in F} w(e)$. We construct the complete graph $K_{|V|} = (V, E' := \binom{V}{2})$ with edge weights $w: E' \rightarrow \mathbb{Z}$ by

$$982 w(\{i, j\}) = \begin{cases} -ab & \{i, j\} \notin E \\ (b-a)b & \{i, j\} \in E \end{cases}$$

983 Note that for any $S \subseteq V$, it holds that

$$984 w(E'(S, V \setminus S)) = (b-a)b \cdot |E(S, V \setminus S)| - ab(|S| \cdot |V \setminus S| - |E(S, V \setminus S)|)$$

$$985 = b^2 \cdot |E(S, V \setminus S)| - ab \cdot |S| \cdot |V \setminus S|.$$

986 Hence, we have

$$987 w(E'(S, V \setminus S)) > 0 \iff \frac{b^2 \cdot |E(S, V \setminus S)|}{ab \cdot |S| \cdot |V \setminus S|} > 1 \iff \frac{|E(S, V \setminus S)|}{|S| \cdot |V \setminus S|} > \frac{a}{b} = t,$$

988 proving the correctness of the reduction. \square

1004 A.3.5 PROOF OF THEOREM 11

1005 *Proof.* The problem is contained in NP since, by Lemma 14, it is sufficient to define a ray ρ as a certificate for positivity. Since rays are one-dimensional subspaces, they are the intersection of $d-1$ hyperplanes corresponding to rows of \mathbf{W}_1 . Hence, the $2(d-1)$ rows of \mathbf{W}_1 that determine ρ form a polynomial-time verifiable certificate.

1006 For the NP-hardness, we reduce from POSITIVE CUT. Given a weighted graph $(G = (V, E), w)$ with $V = [d]$ and $|E| = n$, we define the matrices $\mathbf{W}_1 \in \mathbb{R}^{2n \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times 2n}$ as follows: For each $e = \{i, j\} \in E$, \mathbf{W}_1 contains two rows \mathbf{w}_e and \mathbf{w}'_e , where

$$1007 (\mathbf{w}_e)_\ell := \begin{cases} 1, & \ell = i \\ -1, & \ell = j \text{ and } \mathbf{w}'_e := -\mathbf{w}_e. \\ 0, & \text{else} \end{cases}$$

1008 The corresponding entries of \mathbf{W}_2 are set to $w(e)$. Thus, the 2-layer ReLU neural network computes the map $f: \mathbb{R}^d \rightarrow \mathbb{R}$ with

$$1009 f(\mathbf{x}) = \sum_{\{i, j\} \in E} w(\{i, j\}) \cdot ([\mathbf{x}_i - \mathbf{x}_j]_+ + [\mathbf{x}_j - \mathbf{x}_i]_+).$$

1010 For the correctness, we start with some preliminary observations. For a subset $S \subseteq V$, let $\mathbf{r}_S := \sum_{i \in S} \mathbf{e}_i \in \mathbb{R}^d$ and $\mathbf{r}'_S := -\sum_{i \in S \setminus V} \mathbf{e}_i \in \mathbb{R}^d$ and note that

$$1011 [(\mathbf{r}_S)_i - (\mathbf{r}_S)_j]_+ + [(\mathbf{r}_S)_j - (\mathbf{r}_S)_i]_+ = \begin{cases} 0, & \{i, j\} \notin E(S, V \setminus S) \\ 1, & \{i, j\} \in E(S, V \setminus S) \end{cases}$$

1026 as well as

$$1027 \quad [(\mathbf{r}'_S)_i - (\mathbf{r}'_S)_j]_+ + [(\mathbf{r}'_S)_j - (\mathbf{r}'_S)_i]_+ = \begin{cases} 0, & \{i, j\} \notin E(S, V \setminus S) \\ 1, & \{i, j\} \in E(S, V \setminus S) \end{cases}$$

1029 and therefore

$$1031 \quad f(\mathbf{r}_S) = f(\mathbf{r}'_S) = \sum_{\{i, j\} \in E(S, V \setminus S)} w(\{i, j\}) = w(E(S, V \setminus S)).$$

1034 As regards the correctness, if (G, w) is a yes-instance, that is, there exists a subset $S \subseteq V$
1035 with $w(E(S, V \setminus S)) > 0$, then $f(\mathbf{r}_S) > 0$.

1036 Conversely, assume that there is a $\mathbf{v} \in \mathbb{R}^d$ with $f(\mathbf{v}) > 0$. Let $\pi \in \mathcal{S}_{d+1}$ be a permutation such
1037 that $\mathbf{v}_{\pi(1)} \leq \mathbf{v}_{\pi(2)} \leq \dots \leq \mathbf{v}_{\pi(d+1)}$, where $\mathbf{v}_{d+1} := 0$. Since all breakpoints \mathbf{x} of $f_{\mathbf{W}_1}$ satisfy
1038 $\mathbf{x}_i = \mathbf{x}_j$ for some i and j , the map $f_{\mathbf{W}_1}$ is linear within the pointed d -dimensional cone

$$1040 \quad C := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x}_{\pi(1)} \leq \dots \leq \mathbf{x}_{\pi(d+1)}\} = \bigcap_{i=1}^d \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}_{\pi(i)} \leq \mathbf{x}_{\pi(i+1)}\},$$

1043 where again $\mathbf{x}_{d+1} := 0$. Hence, by linearity of the output layer and Observation 3, the value $\phi(\mathbf{v})$ is
1044 a conical combination of the values of f on the ray generators of the rays of C and hence there is ray
1045 generator \mathbf{r} of C such that $f(\mathbf{r}) > 0$. All rays of C are the intersection of C and $n - 1$ hyperplanes
1046 of the form $\{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x}_{\pi(i)} = \mathbf{x}_{\pi(i+1)}\}$. We denote these rays by

$$1047 \quad \rho_k := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x}_{\pi(1)} = \dots = \mathbf{x}_{\pi(k-1)} \leq \mathbf{x}_{\pi(k)} = \dots = \mathbf{x}_{\pi(d+1)}\}$$

1049 for $k \in [d]$. Let $k \in [d]$ and $S = \{\pi(k), \dots, \pi(d)\}$. If $\pi^{-1}(d+1) < k$, then \mathbf{r}_S generates ρ_k and
1050 otherwise \mathbf{r}'_S generates ρ_k . Since ϕ is positive on one of these ray generators, we can conclude that
1051 there is a $S \subseteq V$ such that $f(\mathbf{r}_S) = f(\mathbf{r}'_S) > 0$ which implies $w(E(S, V \setminus S)) > 0$. \square

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079