Strong mixed-integer programming formulations for trained neural networks

Ross Anderson · Joey Huchette · Will Ma · Christian Tjandraatmadja · Juan Pablo Vielma

Abstract We present strong mixed-integer programming (MIP) formulations for high-dimensional piecewise linear functions that correspond to trained neural networks. These formulations can be used for a number of important tasks, such as verifying that an image classification network is robust to adversarial inputs, or solving decision problems where the objective function is a machine learning model. We present a generic framework, which may be of independent interest, that provides a way to construct sharp or ideal formulations for the maximum of d affine functions over arbitrary polyhedral input domains. We apply this result to derive MIP formulations for a number of the most popular nonlinear operations (e.g. ReLU and max pooling) that are strictly stronger than other approaches from the literature. We corroborate this computationally, showing that our formulations are able to offer substantial improvements in solve time on verification tasks for image classification networks.

Keywords Mixed-integer programming, Formulations, Deep learning

Mathematics Subject Classification (2010) 90C11

Acknowledgements The authors gratefully acknowledge Yeesian Ng and Ondřej Sýkora for many discussions on the topic of this paper, and for their work on the development of the `tf.opt` package used in the computational experiments.

R. Anderson
E-mail: rander@google.com

J. Huchette
E-mail: jhuchette@google.com

W. Ma
E-mail: willma@google.com

C. Tjandraatmadja
E-mail: ctjandra@google.com

J. P. Vielma
E-mail: jvielma@mit.edu

1 Introduction

Deep learning has proven immensely powerful at solving a number of important predictive tasks arising in areas such as image classification, speech recognition, machine translation, and robotics and control [31,46]. The workhorse model in deep learning is the feedforward network $\texttt{NN}(x^0) = x^s$, where

$$x^i_j = \texttt{NL}^{i,j}(w^{i,j} \cdot x^{i-1} + b^{i,j}) \tag{1}$$

for each layer $i \in [\![s]\!] \overset{\text{def}}{=} \{1, \ldots, s\}$ and $j \in [\![m_i]\!]$. Note that the input $x^0 \in \mathbb{R}^{m_0}$ might be high-dimensional, and that the output $x^s \in \mathbb{R}^{m_s}$ may be multivariate.

In this recursive description, $\texttt{NL}^{i,j}$ is some simple nonlinearity, and $w^{i,j}$ and $b^{i,j}$ are the weights and bias of an affine function which is learned during the training procedure. In its simplest and most common form, the nonlinearity would be the rectified linear unit (ReLU), defined as $\texttt{ReLU}(v) = \max\{0, v\}$. Each equation in (1) corresponds to a single *neuron* in the network, coupling together a high-dimensional affine function and a simple nonlinearity.

Many standard nonlinearities $\texttt{NL}$, such as the ReLU, are *piecewise linear*: that is, there exists a partition $\{S^i \subseteq D\}^d_{i=1}$ of the domain and affine functions $\{f^i\}^d_{i=1}$ such that $\texttt{NL}(x) = f^i(x)$ for all $x \in S^i$. If all nonlinearities describing $\texttt{NN}$ are piecewise linear, then the entire network $\texttt{NN}$ is piecewise linear as well.

There are numerous contexts in which one may want to solve an optimization problem containing a trained neural network such as $\texttt{NN}$. For example, this paradigm arises in deep reinforcement learning problems with high dimensional action spaces and where any of the cost-to-go function, immediate cost, or the state transition functions are learned by a neural network [4,23,52,58,73]. Alternatively, there has been significant recent interest in verifying the robustness of trained neural networks deployed in systems like self-driving cars that are incredibly sensitive to unexpected behavior from the machine learning model [19,55,62]. Relatedly, a string of recent work has used optimization over neural networks trained for visual perception tasks to *generate* new images which are “most representative” for a given class [54], are “dream-like” [53], or adhere to a particular artistic style via neural style transfer [30].

1.1 MIP formulation preliminaries

In this work, we study mixed-integer programming (MIP) approaches for optimization problems containing trained neural networks. In contrast to heuristic or local search methods often deployed for the applications mentioned above, MIP offers a framework for producing provably optimal solutions. This is particularly important, for example, in verification problems, where rigorous dual bounds can guarantee robustness in a way that purely primal methods cannot.

We focus on constructing MIP formulations for the *graph* of individual neurons:

$$\text{gr}(\texttt{NL} \circ f; D) \overset{\text{def}}{=} \{\, (x, (\texttt{NL} \circ f)(x)) \mid x \in D \,\}, \tag{2}$$

where $\circ$ is the standard function composition operator $(g \circ f)(x) = g(f(x))$. This substructure consists of a single nonlinear activation function $\texttt{NL}$, taking as input an affine function f over a η-dimensional input domain D. The nonlinearity is handled by introducing auxiliary binary variables z to select among the various pieces of the piecewise linear function. We focus on these particular substructures because we can readily produce a MIP formulation for the entire network as the composition of formulations for each individual neuron.

Definition 1 Throughout, we will notationally use the convention that $x \in \mathbb{R}^\eta$ are *input variables*, $y \in \mathbb{R}$ is the *output variable*, $v \in \mathbb{R}^p$ are any potential auxiliary continuous variables, and $z \in \mathbb{R}^q$ are auxiliary binary variables. The orthogonal projection operator Proj will be subscripted by the variables to project onto, e.g. $\operatorname{Proj}_{x,y}(R) = \{\, (x,y) \mid \exists v, z \text{ s.t. } (x,y,v,z) \in R \,\}$ is the orthogonal projection of R onto the "original space" of x and y variables. We denote by $\operatorname{ext}(Q)$ the set of extreme points of a polyhedron Q.

Take the set $S \subseteq \mathbb{R}^{\eta+1}$ we want to model (for example, $\operatorname{gr}(\texttt{NL} \circ f; D)$), and a polyhedron $Q \subset \mathbb{R}^{\eta+1+p+q}$. Then:

- A *(valid) mixed-integer programming (MIP) formulation* of S consists of the linear constraints on $(x,y,v,z) \in \mathbb{R}^{\eta+1+p+q}$ which define a polyhedron Q, combined with the integrality constraints $z \in \{0,1\}^q$, such that
$$S = \operatorname{Proj}_{x,y}\left(Q \cap \left(\mathbb{R}^{\eta+1+p} \times \{0,1\}^q\right)\right).$$
We refer to Q as the *linear programming (LP) relaxation* of the formulation.
- A MIP formulation is *sharp* if $\operatorname{Proj}_{x,y}(Q) = \operatorname{Conv}(S)$.
- A MIP formulation is *hereditarily sharp* if fixing any subset of binary variables z to 0 or 1 preserves sharpness.[1]
- A MIP formulation is *ideal* (or *perfect*) if $\operatorname{ext}(Q) \subseteq \mathbb{R}^{\eta+1+p} \times \{0,1\}^q$.
- The *separation* problem for a family of inequalities is to find a valid inequality violated by a given point or certify that no such inequality exists.
- An inequality is *valid* for the formulation if each integer feasible point in $\underline{Q} = \{\, (x,y,v,z) \in Q \mid z \in \{0,1\}^q \,\}$ satisfies the inequality. Moreover, a valid inequality is *facet-defining* if the dimension of the points in $\underline{Q}$ satisfying the inequality at equality is exactly one less than the dimension of $\underline{Q}$ itself.

Note that ideal formulations are sharp, but the converse is not necessarily the case [66, Proposition 2.4]. In this sense, ideal formulations offer the tightest possible relaxation, and the integrality property in Definition 1 tends to lead

[1] More formally, for any disjoint subsets $I^0, I^1 \subseteq [\![q]\!]$, take
$$Q'(I^0,I^1) = \left\{ (x,y,v,z) \in Q \;\middle|\; \begin{array}{l} z_k = 0 \ \forall k \in I^0 \\ z_k = 1 \ \forall k \in I^1 \end{array} \right\} \text{ and}$$
$$S'(I^0,I^1) = \operatorname{Proj}_{x,y}\left(\{\, (x,y,v,z) \in Q'(I^0,I^1) \mid z \in \{0,1\}^q \,\}\right).$$
A MIP formulation is *hereditarily sharp* if, for each disjoint $I^0, I^1 \subseteq [\![q]\!]$, $\operatorname{Proj}_{x,y}(Q'(I^0,I^1)) = \operatorname{Conv}(S'(I^0,I^1))$.

to superior computational performance. Furthermore, note that a hereditarily sharp formulation is a formulation which retains its sharpness at every node in a branch-and-bound tree, and as such is potentially superior to a formulation which only guarantees sharpness at the root node [39,40]. Additionally, it is important to keep in mind that modern MIP solvers will typically require an explicit, finite list of inequalities defining Q.

1.2 Our contributions

We highlight the contributions of this work as follows.

1. *Generic recipes for building strong MIP formulations of the maximum of d affine functions for any bounded polyhedral input domain.*
   - [**Propositions 3 and 4**] We derive both primal and dual characterizations for ideal formulations via the *Cayley embedding.*
   - [**Propositions 5 and 6**] We relax the Cayley embedding in a particular way, and use this to derive simpler primal and dual characterizations for (hereditarily) sharp formulations.
   - We discuss how to separate both dual characterizations via subgradient descent.
2. *Simplifications for common special cases.*
   - [**Corollary 1**] We show the equivalence of the ideal and sharp characterizations when $d = 2$.
   - [**Proposition 7**] We show that, if the input domain is a product of simplices, the separation problem of the sharp formulation can be reframed as a series of transportation problems.
   - [**Corollaries 2 and 3, and Proposition 9**] When the input domain is a product of simplices, and either (1) $d = 2$, or (2) each simplex is two-dimensional, we provide an explicit, finite description for the sharp formulation. Furthermore, none of these inequalities are redundant.
3. *Application of these results to construct MIP formulations for neural network nonlinearities.*
   - [**Proposition 14**] We derive an explicit ideal formulation for the ReLU nonlinearity over a box input domain, the most common case. Separation over this ideal formulation can be performed in time linear in the input dimension.
   - [**Corollary 5**] We derive an explicit ideal formulation for the ReLU nonlinearity where some (or all) of the input domain is *one-hot encoded* categorical or discrete data. Again, the separation can be performed efficiently, and none of the inequalities are redundant.
   - [**Proposition 10**] We derive a tightened big-M formulation for the maximum of d affine functions.
   - [**Proposition 12**] We present an explicit sharp formulation for the maximum of d affine functions over a box input domain, and provide an efficient separation routine.

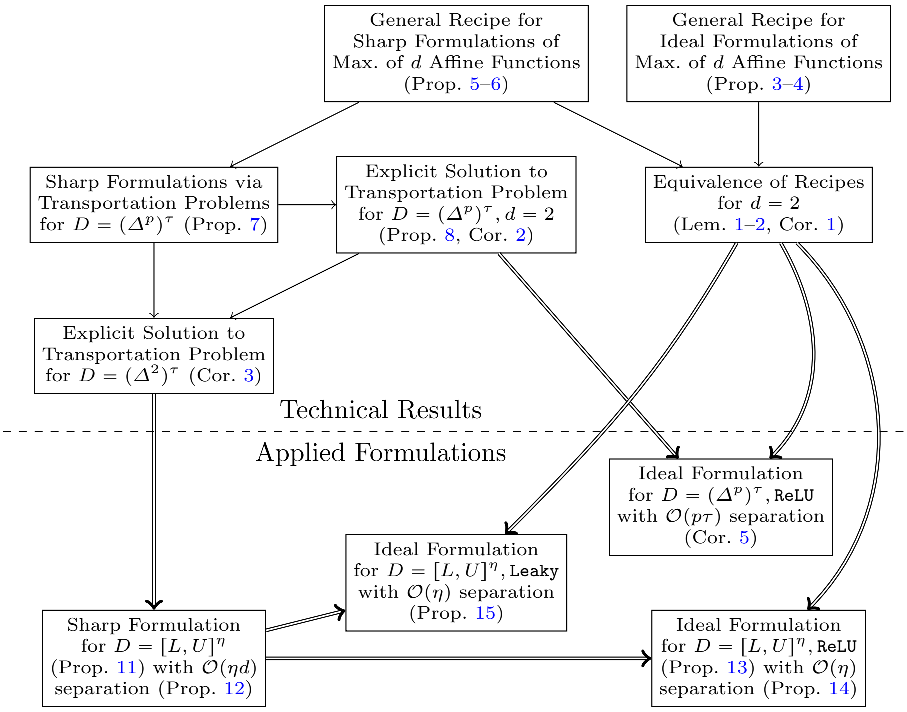

General Recipe for Sharp Formulations of Max. of d Affine Functions (Prop. 5–6)
General Recipe for Ideal Formulations of Max. of d Affine Functions (Prop. 3–4)
Sharp Formulations via Transportation Problems for $D = (\Delta^p)^\tau$ (Prop. 7)
Explicit Solution to Transportation Problem for $D = (\Delta^p)^\tau, d = 2$ (Prop. 8, Cor. 2)
Equivalence of Recipes for $d = 2$ (Lem. 1–2, Cor. 1)
Explicit Solution to Transportation Problem for $D = (\Delta^2)^\tau$ (Cor. 3)
Technical Results
Applied Formulations
Ideal Formulation for $D = (\Delta^p)^\tau$, ReLU with $\mathcal{O}(p\tau)$ separation (Cor. 5)
Ideal Formulation for $D = [L, U]^\eta$, Leaky with $\mathcal{O}(\eta)$ separation (Prop. 15)
Sharp Formulation for $D = [L, U]^\eta$ (Prop. 11) with $\mathcal{O}(\eta d)$ separation (Prop. 12)
Ideal Formulation for $D = [L, U]^\eta$, ReLU (Prop. 13) with $\mathcal{O}(\eta)$ separation (Prop. 14)

Fig. 1: A roadmap of our results. The single arrows depict dependencies between our technical results, while the double arrows depict the way in which we use these results to establish our applied formulations in the paper. Notationally, $[L, U]^\eta$ is an η-dimensional box, and $(\Delta^p)^\tau$ refers to a product of τ simplices, each with p components. Additional results not depicted in the graph include: most of our formulations do not contain redundant constraints (Prop. 9), and a big-M formulation for the maximum of d functions (Prop. 10).

  - [**Propositions 15 and 16**] We produce similar results for more exotic nonlinearities: the leaky ReLU, the clipped ReLU, and the hard tanh.
4. *Computational experiments on verification problems arising from image classification networks trained on the MNIST digit data set.*
  - We observe that our new formulations, along with just a few rounds of separation over our families of cutting planes, can improve the solve time of Gurobi on verification problems by orders of magnitude.

Our contributions are depicted in Figure 1. It serves a roadmap of the paper.

1.3 Relevant prior work

In recent years a number of authors have used MIP formulations to model trained neural networks [18,20,24,29,38,45,49,58,60,61,64,73,74], mostly applying big-M formulation techniques to ReLU-based networks. When applied to a single neuron of the form (2), these big-M formulations will not be ideal or offer an exact convex relaxation; see Example 1 for an illustration. Additionally, a stream of literature in the deep learning community has studied convex relaxations [10,26,27,56,57], primarily for verification tasks. Moreover, some authors have investigated how to use convex relaxations within the training procedure in the hopes of producing neural networks with a priori robustness guarantees [25,71,72].

Beyond MIP and convex relaxations, a number of authors have investigated other algorithmic techniques for modeling trained neural networks in optimization problems, drawing primarily from the satisfiability, constraint programming, and global optimization communities [8,9,41,48,59]. Another intriguing direction studies restrictions to the space of models that may make the optimization problem over the network inputs simpler: for example, the classes of binarized [42] or input convex [2] neural networks.

Broadly, our work fits into a growing body of research in prescriptive analytics and specifically the "predict, then optimize" framework, which considers how to embed trained machine learning models into optimization problems [12,14,21,22,28,34,51]. Additionally, the formulations presented below have connections with existing structures studied in the MIP and constraint programming community like indicator variables, on/off constraints, and convex envelopes [5,11,16,35,36,63].

1.4 Starting assumptions and notation

We use the following notational conventions throughout the paper.

- The nonnegative orthant: $\mathbb{R}_{\geqslant 0} \overset{\text{def}}{=} \{\, x \in \mathbb{R} \mid x \geqslant 0 \,\}$.
- The d-dimensional simplex: $\Delta^d \overset{\text{def}}{=} \left\{\, x \in \mathbb{R}^d_{\geqslant 0} \;\middle|\; \sum_{i=1}^d x_i = 1 \,\right\}$.
- The set of integers from 1 to n: $[\![n]\!] = \{1, \ldots, n\}$.
- "Big-M" coefficients: $M^+(f; D) \overset{\text{def}}{=} \max_{x \in D} f(x)$ and $M^-(f; D) \overset{\text{def}}{=} \min_{x \in D} f(x)$.
- The dilation of a set: if $z \in \mathbb{R}_{\geqslant 0}$ and $D \subseteq \mathbb{R}^\eta$, then $z \cdot D \overset{\text{def}}{=} \{\, zx \mid x \in D \,\}$.[2]

Furthermore, throughout we will make the following simplifying assumptions.

Assumption 1 *The input domain D is a bounded polyhedra.*

While a bounded input domain assumption will make the formulations and analysis considerably more difficult than the unbounded setting (see [5] for a similar phenomenon), it ensures that standard MIP representability conditions are satisfied (e.g. [66, Section 11]). Furthermore, variable bounds are natural for many applications (for example in verification problems), and are absolutely essential for ensuring reasonable dual bounds.

[2] If $D = \{\, x \in \mathbb{R}^\eta \mid Ax \leqslant b \,\}$ is polyhedral, then $z \cdot D = \{\, x \in \mathbb{R}^\eta \mid Ax \leqslant bz \,\}$.

Assumption 2 *Each neuron is* irreducible*: for any $k \in [\![d]\!]$, there exists some $x \in D$ where $f^k(x) > f^\ell(x)$ for each $\ell \neq k$.*

Observe that if a neuron is not irreducible, this means that it is unnecessarily complex, and one or more of the affine functions can be completely removed. Moreover, the assumption can be verified in polynomial time by solving d LPs, since it is equivalent to the condition

$$\max_{x,\Delta} \left\{ \Delta \;\middle|\; x \in D,\; \Delta \leqslant f^k(x) - f^\ell(x) \;\forall \ell \neq k \right\} > 0$$

for all $k \in [\![d]\!]$. In the special case where $d = 2$ (e.g. ReLU) and D is a box, this can be checked in linear time. Finally, if the assumption does not hold, it will not affect the validity of the formulations or cuts derived in this work, though certain results pertaining to non-redundancy or facet-defining properties may no longer hold.

2 Motivating example: The ReLU nonlinearity over a box domain

The ReLU is the workhorse of deep learning models: it is easy to reason about, introduces little computational overhead, and despite its simple structure is nonetheless capable of articulating complex nonlinear relationships.

2.1 A big-M formulation

To start, we will consider the ReLU in the simplest possible setting: where the input is univariate. Take the two-dimensional set $\operatorname{gr}(\texttt{ReLU}; [l, u])$, where $[l, u]$ is some interval in $\mathbb{R}$ containing zero. It is straightforward to construct an ideal formulation for this univariate ReLU.

Proposition 1 *An ideal formulation for* $\operatorname{gr}(\texttt{ReLU}; [l, u])$ *is:*

$$y \geqslant x \tag{3a}$$
$$y \leqslant x - l(1 - z) \tag{3b}$$
$$y \leqslant uz \tag{3c}$$
$$(x, y, z) \in \mathbb{R} \times \mathbb{R}_{\geqslant 0} \times [0, 1] \tag{3d}$$
$$z \in \{0, 1\}. \tag{3e}$$

Proof Follows from inspection, or as a special case of Proposition 13 to be presented in Section 5.2. □

A more realistic setting would have a ReLU nonlinearity whose input is some affine function $f : [L, U] \to \mathbb{R}$ with box constrained input $[L, U] \subset \mathbb{R}^\eta$. The box input corresponds to known (finite) bounds on each component, which can typically be efficiently computed via interval arithmetic or other standard methods.

Observe that we can model the multivariate ReLU as a simple composition of a univariate ReLU and an affine function:

$$\left\{ (x,y,z) \;\middle|\; \begin{array}{c} (f(x),y,z) \in \operatorname{gr}\left(\texttt{ReLU}; [M^-(f;[L,U]), M^+(f;[L,U])]\right) \\ L \leqslant x \leqslant U \end{array} \right\}. \tag{4}$$

Using formulation (3) as a submodel, we can write a formulation for the ReLU over a box domain as:

$$\begin{aligned} y &\geqslant f(x) && \text{(5a)}\\ y &\leqslant f(x) - M^-(f;[L,U]) \cdot (1-z) && \text{(5b)}\\ y &\leqslant M^+(f;[L,U]) \cdot z && \text{(5c)}\\ (x,y,z) &\in [L,U] \times \mathbb{R}_{\geqslant 0} \times [0,1] && \text{(5d)}\\ z &\in \{0,1\}. && \text{(5e)} \end{aligned}$$

This is the approach taken recently in the bevy of papers referenced in Section 1.3. Unfortunately, after the composition with the affine function f over a box input domain, this formulation is no longer sharp.

Example 1 If $f(x) = x_1 + x_2 - 1.5$, formulation (5) for $\operatorname{gr}(\texttt{ReLU} \circ f; [0,1]^2)$ is

$$\begin{aligned} y &\geqslant x_1 + x_2 - 1.5 && \text{(6a)}\\ y &\leqslant x_1 + x_2 - 1.5 + 1.5(1-z) && \text{(6b)}\\ y &\leqslant 0.5z && \text{(6c)}\\ (x,y,z) &\in [0,1]^2 \times \mathbb{R}_{\geqslant 0} \times [0,1] && \text{(6d)}\\ z &\in \{0,1\}. && \text{(6e)} \end{aligned}$$

The point $(\hat{x},\hat{y},\hat{z}) = ((1,0),0.25,0.5)$ is feasible for the LP relaxation (6a-6d). However, observe that the inequality $y \leqslant 0.5x_2$ is valid for $\operatorname{gr}(\texttt{ReLU} \circ f; [0,1]^2)$, but is violated by $(\hat{x},\hat{y})$. Therefore, the formulation does not offer an exact convex relaxation (and, hence, is not ideal). See Figure 2 for an illustration: on the left, of the big-M formulation projected to (x,y)-space, and on the right, the tightest possible convex relaxation.

Moreover, the integrality gap of (5) can be arbitrarily bad, even in fixed dimension η.

Example 2 Fix $\gamma \in \mathbb{R}_{\geqslant 0}$ and even $\eta \in \mathbb{N}$. Take the affine function $f(x) = \sum_{i=1}^{\eta} x_i$, the input domain $[L,U] = [-\gamma,\gamma]^\eta$, and $\hat{x} = \gamma \cdot (1,-1,\cdots,1,-1)$ as a scaled vector of alternating positive and negative ones. We can check that $(\hat{x},\hat{y},\hat{z}) = (\hat{x}, \frac{1}{2}\gamma\eta, \frac{1}{2})$ is feasible for the LP relaxation of the big-M formulation (5). Additionally, $f(\hat{x}) = 0$, and for any $\tilde{y}$ such that $(\hat{x},\tilde{y}) \in \operatorname{Conv}(\operatorname{gr}(\texttt{ReLU} \circ f; [L,U]))$, then $\tilde{y} = 0$ necessarily. Therefore, there exists a fixed point $\hat{x}$ in the input domain where the tightest possible convex relaxation (for example, from a sharp formulation) is exact, but the big-M formulation deviates from this value by at least $\frac{1}{2}\gamma\eta$.

Intuitively, this example suggests that the big-M formulation can be particularly weak around the boundary of the input domain, as it cares only about the value $f(x)$ of the affine function, and not the particular input value x.

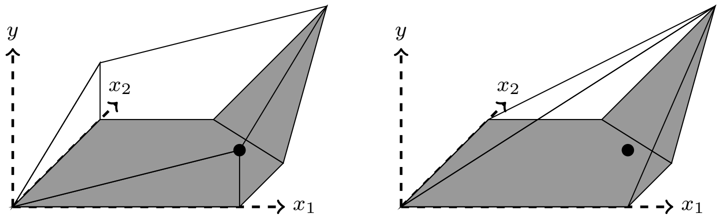


Fig. 2: For $f(x) = x_1 + x_2 - 1.5$: **(Left)** $\text{Conv}(\text{gr}(\texttt{ReLU} \circ f; [0,1]^2))$, and **(Right)** the projection of the big-M formulation (5) to (x,y)-space, where we mark the point $(x,y) = ((1,0), 0.25)$ that is not in the convex hull, but is valid for the projection of the big-M LP relaxation (6a-6d).

2.2 An ideal extended formulation

It is possible to produce an ideal *extended* formulation for the ReLU neuron by introducing a modest number of auxiliary continuous variables. The "multiple choice" formulation

$$
\begin{aligned}
(x,y) &= (x^0,y^0) + (x^1,y^1) && \text{(7a)}\\
y^0 &= 0 \geqslant w \cdot x^0 + b(1-z) && \text{(7b)}\\
y^1 &= w \cdot x^1 + bz \geqslant 0 && \text{(7c)}\\
L(1-z) \leqslant x^0 &\leqslant U(1-z) && \text{(7d)}\\
Lz \leqslant x^1 &\leqslant Uz && \text{(7e)}\\
z &\in \{0,1\}, && \text{(7f)}
\end{aligned}
$$

is an ideal extended formulation for piecewise linear functions [69]. It can alternatively be derived from techniques introduced by Balas [6,7]. We can interpret the LP relaxation of (7) as expressing (x,y) as a convex combination of points $\frac{1}{1-z}(x^0,y^0)$ and $\frac{1}{z}(x^1,y^1)$, one in each piece of the ReLU.

Although the multiple choice formulation offers the tightest possible convex relaxation for a single neuron, it requires a copy x^0 of the input variables (note that it is straightforward to use equations (7a) to eliminate the second copy x^1). This means that when the multiple choice formulation is applied to every neuron in the network to formulate $\texttt{NN}$, the total number of continuous variables required is $m_0 + \sum_{i=1}^r (m_{i-1}+1)m_i$ (using the notation of (1), where m_i is the number of neurons in layer i). In contrast, the big-M formulation requires only $m_0 + \sum_{i=1}^r m_i$ continuous variables to formulate the entire network. As we will see in Section 6, the quadratic growth in size of the extended formulation can quickly become burdensome. Additionally, a folklore observation in the MIP community is that multiple choice formulations tend to not perform as well as expected in simplex-based branch-and-bound algorithms, likely due to degeneracy introduced by the block structure of the formulation [68].

2.3 An ideal MIP formulation without auxiliary continuous variables

In this work, our most broadly useful contribution is the derivation of an ideal MIP formulation for the ReLU nonlinearity over a box domain that is non-extended; that is, it does not require additional auxiliary variables as in formulation (7). We informally summarize our main result in this regard as follows.

Main result for ReLU networks (informal)

There exists an explicit ideal nonextended formulation for the ReLU nonlinearity over a box domain, i.e. it requires only a single auxiliary binary variable. It has an exponential (in η) number of inequality constraints, each of which are facet-defining. However, it is possible to separate over this family of inequalities in time scaling linearly in η.

We defer the formal statement and proof to Section 5.2, for after we have derived the requisite machinery, for this and a number of related results. However, we do note that Theorem 2.3 serves as the main result for the extended abstract version of this work [3], where it is derived through alternative means.

3 Our general machinery: Formulations for the maximum of *d* affine functions

We will state our main structural results in the following generic setting. Take the maximum operator $\texttt{Max}(v_1, \dots, v_d) = \max_{i=1}^d v_i$ over d scalar inputs. We will study the composition of this multivariate nonlinearity with d affine functions $f^i : D \to \mathbb{R}$ with $f^i(x) = w^i \cdot x + b^i$, all sharing some bounded polyhedral domain D:

$$S_{\max} \overset{\text{def}}{=} \operatorname{gr}(\texttt{Max} \circ (f^1, \dots, f^d); D) \equiv \left\{ (x, y) \in D \times \mathbb{R} \;\middle|\; y = \max_{i=1}^{d} f^i(x) \right\},$$

This setting subsumes the ReLU over box input domain presented in Section 2 as a special case with $d = 2$, $f^2(x) = 0$, and $D = [L, U]$. It also covers a number of other settings arising in modern deep learning, either by making D more complex (e.g. one-hot encodings for categorical features), or by increasing d (e.g. max pooling neurons often used in convolutional networks for image classification tasks [17], or in maxout networks [32]).

In this section, we present structural results that characterize the *Cayley embedding* [37,67,68,70] of $S_{\max}$. Take the set

$$S_{\text{cayley}} \overset{\text{def}}{=} \bigcup_{k=1}^{d} \left\{ (x, f^k(x), \mathbf{e}^k) \;\middle|\; x \in D_{|k} \right\},$$

where $\mathbf{e}^k$ is the unit vector where the k-th element is 1, and for each $k \in [\![d]\!]$,

$$\begin{aligned} D_{|k} &\overset{\text{def}}{=} \left\{ x \in D \;\middle|\; k \in \arg\max_{\ell=1}^{d} f^\ell(x) \right\} \\ &\equiv \left\{ x \in D \;\middle|\; w^k \cdot x + b^k \geqslant w^\ell \cdot x + b^\ell \quad \forall \ell \neq k \right\} \end{aligned}$$

is the portion of the input domain D where f^k attains the maximum.[3] The *Cayley embedding* of $S_{\max}$ is the convex hull of this set: $R_{\text{cayley}} \overset{\text{def}}{=} \text{Conv}(S_{\text{cayley}})$.

The following straightforward observation holds directly from the definition of R_{cayley}.

Observation 1 *The set R_{cayley} is a bounded polyhedron, and an ideal formulation for $S_{\max}$ is given by the system $\left\{ (x, y, z) \in R_{\text{cayley}} \;\middle|\; z \in \{0,1\}^d \right\}$.*

Therefore, if we can produce an explicit inequality description for R_{cayley}, we immediately have an ideal MIP formulation for $S_{\max}$. Indeed, we have already seen an extended representation in (7) when $d = 2$, which we now state in the more general setting (projecting out the superfluous copies of the y output variable).

Proposition 2 *An ideal MIP formulation for $S_{\max}$ is:*

$$(x, y) = \sum_{k=1}^{d} (\tilde{x}^k, \tilde{y}^k) \tag{8a}$$

$$\tilde{x}^k \in z_k \cdot D_{|k} \qquad \forall k \in [\![d]\!] \tag{8b}$$

$$z \in \Delta^d \tag{8c}$$

$$z \in \{0,1\}^d. \tag{8d}$$

Denote its LP relaxation by $R_{\text{extended}} = \left\{ (x, y, z, \tilde{x}^1, \ldots, \tilde{x}^k) \;\middle|\; (8a - 8c) \right\}$. Then $\text{Proj}_{x,y,z}(R_{\text{extended}}) = R_{\text{cayley}}$.

Although this formulation is ideal and polynomially-sized, as noted in Section 2.2 and corroborated in the computational experiments in Section 6, this extended formulation can exhibit poor practical performance. Observe that, from definition, constraint (8b) for a given $k \in [\![d]\!]$ is equivalent to the set of constraints $\tilde{x}^k \in z_k \cdot D$ and $w^k \cdot \tilde{x}^k + b^k z_k \geqslant w^\ell \cdot \tilde{x}^k + b^\ell z_k$ for each $\ell \neq k$.

3.1 A recipe for constructing ideal formulations

Our goal in this section is to derive generic tools that allow us to build ideal formulations for $S_{\max}$ via the Cayley embedding.

[3] In fact, much of the analysis in Section 3.1 will carry over for nonconvex continuous piecewise linear functions by setting $D_{|k}$ as the domain for each corresponding piece. We return to this generalization in Section 5.5.

3.1.1 A primal characterization

Our first structural result provides a characterization for the Cayley embedding. Although it is not an explicit polyhedral characterization, we will subsequently see how it can be massaged into a more practically amenable form.

Take the system

$$y \leqslant \overline{g}(x,z) \tag{9a}$$
$$y \geqslant \underline{g}(x,z) \tag{9b}$$
$$(x,y,z) \in D \times \mathbb{R} \times \Delta^d, \tag{9c}$$

where

$$\overline{g}(x,z) \overset{\text{def}}{=} \max_{\tilde{x}^1,\ldots,\tilde{x}^d} \left\{ \sum_{k=1}^{d} w^k \cdot \tilde{x}^k + b^k z_k \;\middle|\; \begin{array}{l} x = \sum_k \tilde{x}^k \\ \tilde{x}^k \in z_k \cdot D_{|k} \quad \forall k \in [\![d]\!] \end{array} \right\}$$
$$\underline{g}(x,z) \overset{\text{def}}{=} \min_{\tilde{x}^1,\ldots,\tilde{x}^d} \left\{ \sum_{k=1}^{d} w^k \cdot \tilde{x}^k + b^k z_k \;\middle|\; \begin{array}{l} x = \sum_k \tilde{x}^k \\ \tilde{x}^k \in z_k \cdot D_{|k} \quad \forall k \in [\![d]\!] \end{array} \right\}.$$

and define the set $R_{\text{ideal}} \overset{\text{def}}{=} \{ (x,y,z) \mid (9) \}$.

Proposition 3 *The set R_{ideal} is polyhedral, and $R_{\text{ideal}} = R_{\text{cayley}}$.*

Proof By Proposition 2, it suffices to show that $R_{\text{ideal}} = \text{Proj}_{x,y,z}(R_{\text{extended}})$. We start by observing that, as $\overline{g}$ (respectively $\underline{g}$) is concave (resp. convex) in its imputs as it is the value function of a linear program (cf. [13, Theorem 5.1]). Therefore, the set of points satisfying (9) is convex.

Let $(\hat{x},\hat{y},\hat{z})$ be an extreme point of R_{ideal}. Then it must satisfy either (9a) or (9b) at equality, as otherwise it is a convex combination of $(\hat{x},\hat{y}-\epsilon,\hat{z})$ and $(\hat{x},\hat{y}+\epsilon,\hat{z})$ for some $\epsilon > 0$. Take $\tilde{x}^1,\ldots,\tilde{x}^d$ that optimizes $\overline{g}(x,z)$ or $\underline{g}(x,z)$, depending on which constraint is satisfied at equality. Then $(\hat{x},\hat{y},\hat{z},\tilde{x}^1,\ldots,\tilde{x}^d) \in R_{\text{extended}}$. In other words, $\text{ext}(R_{\text{ideal}}) \subseteq \text{Proj}_{x,y,z}(R_{\text{extended}})$, and thus $R_{\text{ideal}} \subseteq \text{Proj}_{x,y,z}(R_{\text{extended}})$ by convexity.

Conversely, let $(\hat{x},\hat{y},\hat{z},\tilde{x}^1,\ldots,\tilde{x}^d) \in R_{\text{extended}}$. Then $\hat{y} = \sum_{k=1}^{d}(w^k \cdot \tilde{x}^k + b^k \hat{z}_k) \leqslant \overline{g}(\hat{x},\hat{z})$, as $(\{\tilde{x}^k\}_{k=1}^{d})$ is feasible for the optimization problem in $\overline{g}(x,z)$. Likewise, $\hat{y} \geqslant \underline{g}(\hat{x},\hat{z})$, and the remaining constraints are also satisfied. Therefore, $(\hat{x},\hat{y},\hat{z}) \in R_{\text{ideal}}$.

Polyhedrality of R_{ideal} then follows immediately, as R_{cayley} is itself a polyhedron. □

Note that the input domain constraint $x \in D$ is implied by the constraints (9a)–(9b), and therefore do not need to be explicitly included in this description. However, we include it here in our description for clarity. Moreover, observe that $\overline{g}$ is a function from $D \times \Delta^d$ to $\mathbb{R} \cup \{-\infty\}$, since the optimization problem may be infeasible, but is always bounded from above since D is bounded. Likewise, $\underline{g}$ is a function from $D \times \Delta^d$ to $\mathbb{R} \cup \{+\infty\}$.

3.1.2 A dual characterization

From Proposition 3, we can derive a more useful characterization by applying Lagrangian duality to the LPs describing the envelopes $\overline{g}(x,z)$ and $\underline{g}(x,z)$.

Proposition 4 *The Cayley embedding R_{cayley} is equal to all (x,y,z) satisfying*

$$y \leqslant \overline{\alpha} \cdot x + \sum_{k=1}^{d} \left(\max_{x^k \in D_{|k}} \{(w^k - \overline{\alpha}) \cdot x^k\} + b^k \right) z_k \quad \forall \overline{\alpha} \in \mathbb{R}^\eta \tag{10a}$$

$$y \geqslant \underline{\alpha} \cdot x + \sum_{k=1}^{d} \left(\min_{x^k \in D_{|k}} \{(w^k - \underline{\alpha}) \cdot x^k\} + b^k \right) z_k \quad \forall \underline{\alpha} \in \mathbb{R}^\eta \tag{10b}$$

$$(x,y,z) \in D \times \mathbb{R} \times \Delta^d. \tag{10c}$$

Proof Consider the upper bound inequalities for y in Proposition 3, that is, (9a). Now apply the change of variables $x^k \leftarrow \frac{\tilde{x}^k}{z_k}$ for each $k \in [\![d]\!]$ and, for all (x,z), take the Lagrangian dual of the optimization problem in $\overline{g}(x,z)$ with respect to the constraint $x = \sum_k x^k z_k$. Note that the duality gap is zero since the problem is an LP. We then obtain that

$$\begin{aligned}\overline{g}(x,z) &= \min_{\overline{\alpha}} \max_{x^k \in D_{|k}} \sum_{k=1}^{d} \left(w^k \cdot x^k + b^k\right) z_k + \overline{\alpha} \cdot \left(x - \sum_{k=1}^{d} x^k z_k \right) \\ &= \min_{\overline{\alpha}} \ \overline{\alpha} \cdot x + \sum_{k=1}^{d} \left(\max_{x^k \in D_{|k}} \{(w^k - \overline{\alpha}) \cdot x^k\} + b^k \right) z_k. \end{aligned} \tag{11}$$

In other words, we can express the constraint (9a) as the family of inequalities (10a). The same can be done with the constraint (9b), yielding the set of inequalities (10b). Therefore, $\{ \, (x,y,z) \mid (10) \, \} = \text{Proj}_{x,y,z}(R_{\text{extended}}) = R_{\text{cayley}}$. □

This gives an exact outer description for the Cayley embedding in terms of an infinite number of linear inequalities. Despite this, the formulation enjoys a simple, interpretable form: we can view the inequalities as choosing coefficients on x and individually tightening the coefficients on z according to explicitly described LPs. In later sections, we will see that this decoupling is helpful to simplify (a variant of) this formulation for special cases.

Separating a point $(\hat{x},\hat{y},\hat{z})$ over (10a) can be done by evaluating $\overline{g}(\hat{x},\hat{z})$ in the form (11) (and in the analogous form of $\underline{g}(\hat{x},\hat{z})$ for (10b)). As typically done when using Lagrangian relaxation, this optimization problem can be solved via a subgradient or bundle method, where each subgradient can be computed by solving the inner LP in (11) for all $x^k \in D_{|k}$. Observe that any feasible solution $\overline{\alpha}$ for the optimization problem in (11) yields a valid inequality. However, this optimization problem is unbounded when $(\hat{x},\hat{z}) \notin \text{Proj}_{x,z}(R_{\text{cayley}})$ (i.e. when the primal form of $\overline{g}(\hat{x},\hat{z})$ is infeasible). In other words, as illustrated in Figure 3, $\overline{g}$ is an *extended* real valued function such that $\overline{g}(x,z) \in \mathbb{R} \cup \{-\infty\}$, so care must be taken to avoid numerical instabilities when separating a point $(\hat{x}, \hat{z})$ where $\overline{g}(\hat{x}, \hat{z}) = -\infty$.[4]

3.2 A recipe for constructing hereditarily sharp formulations

Although Proposition 4 gives a separation-based way to optimize over $S_{\max}$, there are two potential downsides to this approach. First, it does not give us a explicit, finite description for a MIP formulation that we can directly pass to a MIP solver. Second, the separation problem requires optimizing over $D_{|k}$, which may be substantially more complicated than optimizing over D (for example, if D is a box).

Therefore, in this section we set our sights slightly lower and present a similar technique to derive *sharp* MIP formulations for $S_{\max}$. Furthermore, we will see that our formulations trivially satisfy the hereditary sharpness property. In the coming sections, we will see how we can deploy these results in a practical manner, and study settings in which the simpler sharp formulation will also, in fact, be ideal.

3.2.1 A primal characterization

Consider the system

$$\begin{aligned} y &\leqslant \overline{h}(x, z) && \text{(12a)} \\ y &\geqslant w^k \cdot x + b^k \quad \forall k \in [\![d]\!] && \text{(12b)} \\ (x, y, z) &\in D \times \mathbb{R} \times \Delta^d, && \text{(12c)} \end{aligned}$$

where

$$\overline{h}(x, z) \stackrel{\text{def}}{=} \max_{\tilde{x}^1, \ldots, \tilde{x}^d} \left\{ \sum_{k=1}^{d} (w^k \cdot \tilde{x}^k + b^k z_k) \;\middle|\; \begin{array}{l} x = \sum_k \tilde{x}^k \\ \tilde{x}^k \in z_k \cdot D \quad \forall k \in [\![d]\!] \end{array} \right\}.$$

Take the set $R_{\text{sharp}} \stackrel{\text{def}}{=} \{ (x, y, z) \mid (12) \}$.

It is worth dwelling on the differences between the systems (9) and (12). First, we have completely replaced the constraint (9b) with d explicit linear inequalities (12b). Second, when replacing $\overline{g}$ with $\overline{h}$ we have replaced the inner maximization over $D_{|k}$ with an inner maximization over D (modulo constant scaling factors). As we will see in Section 5.1.2, this is particularly advantageous when D is trivial to optimize over (for example, a simplex or a box), allowing us to write these constraints in closed form, whereas optimizing over $D_{|k}$ may be substantially more difficult (i.e. requiring an LP solve).

Furthermore, we will show that while (12) is not ideal, it does enjoy the hereditary sharpness property, which in general may offer a strictly stronger

[4] As is standard in a Benders' decomposition approach, we can address this by adding a *feasibility cut* describing the domain of $\overline{g}$ (the region where it is finite valued) instead of an *optimality cut* of the form (10a).

relaxation than a standard sharp formulation. In particular, the formulation may be stronger than a sharp formulation constructed by composing a big-M formulation, along with an exact convex relaxation in the (x, y)-space produced, for example, by studying the upper concave envelope of the function $\texttt{Max} \circ (f^1, \dots, f^d)$.

Proposition 5 *The set R_{sharp} is polyhedral, and $\left\{ (x, y, z) \in R_{\text{sharp}} \;\middle|\; z \in \{0,1\}^d \right\}$ is a hereditarily sharp MIP formulation of $S_{\max}$.*

Proof For the result, we must show four properties: polyhedrality of R_{sharp}, validity of the formulation whose LP relaxation is R_{sharp}, sharpness, and then hereditary sharpness. We proceed in that order.

To show polyhedrality, consider a fixed value $(\hat{x}, \hat{y}, \hat{z})$ feasible for (12), and presume that we express the domain via the linear inequality constraints $D = \{ x \mid Ax \leqslant c \}$. First, observe that due to (12a) and (12b), $\overline{h}(\hat{x}, \hat{z})$ is bounded from below. Now, using LP duality, we may rewrite

$$\begin{aligned} \overline{h}(\hat{x}, \hat{z}) &= \max_{\tilde{x}^1,\dots,\tilde{x}^d} \left\{ \sum_{k=1}^d w^k \cdot \tilde{x}^k \;\middle|\; \begin{array}{l} \hat{x} = \sum_k \tilde{x}^k \\ A\tilde{x}^k \leqslant \hat{z}_k c \quad \forall k \in [\![d]\!] \end{array} \right\} + \sum_{k=1}^d b^k \hat{z}_k \\ &= \min_{(\alpha,\beta^1,\dots,\beta^k)\in R} \left\{ \alpha \cdot \hat{x} + \sum_{k=1}^d c \cdot \beta^k \hat{z}_k \right\} + \sum_{k=1}^d b^k \hat{z}_k, \end{aligned}$$

where R is a polyhedron that is *independent of $\hat{x}$ and $\hat{z}$*. Therefore, as (a) the above optimization problem is linear with $\hat{x}$ and $\hat{z}$ fixed, and (b) $\overline{h}(\hat{x}, \hat{z})$ is bounded from below,

$$\overline{h}(\hat{x}, \hat{z}) = \min_{(\alpha,\beta^1,\dots,\beta^k)\in \operatorname{ext}(R)} \left\{ \alpha \cdot \hat{x} + \sum_{k=1}^d c \cdot \beta^k \hat{z}_k \right\} + \sum_{k=1}^d b^k \hat{z}_k.$$

In other words, $\overline{h}(\hat{x}, \hat{z})$ is equal to the minimum of a finite number of alternatives which are affine in $\hat{x}$ and $\hat{z}$. Therefore, $\overline{h}$ is a concave continuous piecewise linear function, and so R_{sharp} is polyhedral.

To show validity, we must have that $\operatorname{Proj}_{x,y} \left(R_{\text{sharp}} \cap (\mathbb{R}^\eta \times \mathbb{R} \times \{0,1\}^d) \right) = S_{\max}$. Observe that if $(\hat{x}, \hat{y}, \hat{z}) \in R_{\text{sharp}} \cap (\mathbb{R}^\eta \times \mathbb{R} \times \{0,1\}^d)$, then $\hat{z} = \mathbf{e}^\ell$ for some $\ell \in [\![d]\!]$, and

$$\begin{aligned} \overline{h}(\hat{x}, \hat{z}) &= \max_{\tilde{x}^1,\dots,\tilde{x}^d} \left\{ \sum_{k=1}^d w^k \cdot \tilde{x}^k + b^\ell \;\middle|\; \begin{array}{l} \hat{x} = \sum_k \tilde{x}^k \\ \tilde{x}^\ell \in D \\ \tilde{x}^k = \mathbf{0}^\eta \quad \forall k \neq \ell \end{array} \right\} \\ &= \max_{\tilde{x}_\ell} \left\{ w^\ell \tilde{x}^\ell + b^\ell \;\middle|\; \begin{array}{l} \hat{x} = \tilde{x}^\ell \\ \tilde{x}^\ell \in D \end{array} \right\} \\ &= w^\ell \hat{x} + b^\ell, \end{aligned}$$

where in the first line we infer that $\tilde{x}^k = \mathbf{0}^\eta$ for each $k \neq \ell$ from the boundedness of D (recall that if D is bounded, then $0 \cdot D = \{ x \in \mathbb{R}^\eta \mid Ax \leqslant 0 \} = \{\mathbf{0}^\eta\}$).

Along with (12b), this implies that $\hat{y} = w^\ell \cdot \hat{x} + b^\ell$, and that $\hat{y} \geqslant w^k \cdot \hat{x} + b^k$ for each $k \neq \ell$, giving the result.

To show sharpness, we must prove that $\text{Proj}_{x,y}(R_{\text{sharp}}) = \text{Conv}(S_{\max})$. First, recall from Proposition 2 that $\text{Conv}(S_{\max}) = \text{Proj}_{x,y}(R_{\text{extended}})$; thus, we state our proof in terms of R_{extended}. We first show that $\text{Proj}_{x,y}(R_{\text{extended}}) \subseteq \text{Proj}_{x,y}(R_{\text{sharp}})$. Take $(\hat{x}, \hat{y}, \hat{z}, \{\hat{x}^k\}_{k=1}^d) \in R_{\text{extended}}$. Then $\hat{y} = \sum_{k=1}^d (w^k \cdot \hat{x}^k + b^k \hat{z}_k) \leqslant \overline{h}(\hat{x}, \hat{z})$, as $(\{\hat{x}^k\}_{k=1}^d)$ is feasible for the optimization problem in $\overline{h}(x, z)$. It also holds that $\hat{y} \geqslant w^k \cdot \hat{x} + b^k$ for all $k \in [\![d]\!]$ and $\hat{x} \in D$ directly from the definition of $S_{\max}$, giving the result.

Next, we show that $\text{Proj}_{x,y}(R_{\text{sharp}}) \subseteq \text{Proj}_{x,y}(R_{\text{extended}})$. This proof is similar to the proof of Proposition 3, except that we choose z that simplifies the constraints. It suffices to show that $\text{ext}(\text{Proj}_{x,y}(R_{\text{sharp}})) \subseteq \text{Proj}_{x,y}(R_{\text{extended}})$. Let $(\hat{x}, \hat{y}) \in \text{ext}(\text{Proj}_{x,y}(R_{\text{sharp}}))$. Define $\overline{h}(x) \overset{\text{def}}{=} \max_z \left\{ \overline{h}(x, z) \mid z \in \Delta^d \right\}$. Then either $(\hat{x}, \hat{y})$ satisfies $\hat{y} = \overline{h}(\hat{x})$, or it satisfies (12b) at equality for some $k \in [\![d]\!]$, since otherwise $(\hat{x}, \hat{y})$ is a convex combination of the points $(\hat{x}, \hat{y} - \epsilon)$ and $(\hat{x}, \hat{y} + \epsilon)$ feasible for $\text{Proj}_{x,y}(R_{\text{sharp}})$ for some $\epsilon > 0$. We show that in either case, $(\hat{x}, \hat{y}) \in \text{Proj}_{x,y}(R_{\text{extended}})$.

Case 1: Suppose that for some $j \in [\![d]\!]$, $(\hat{x}, \hat{y})$ satisfies the corresponding inequality in (12b) at equality; that is, $\hat{y} = w^j \cdot \hat{x} + b^j$. Then the point $(\hat{x}, \hat{y}, \mathbf{e}_j, \{\hat{x}^k\}_{k=1}^d) \in R_{\text{extended}}$, where $\hat{x}^j = x$ and $\hat{x}^\ell = 0$ if $\ell \neq j$. Hence, $(\hat{x}, \hat{y}) \in \text{proj}_{x,y}(R_{\text{extended}})$.

Case 2: Suppose $(\hat{x}, \hat{y})$ satisfies $\hat{y} = \overline{h}(\hat{x})$. Let $\hat{z}$ be an optimal solution for the optimization problem defining $\overline{h}(\hat{x})$, and $\{\hat{x}^k\}_{k=1}^d$ be an optimal solution for $\overline{h}(\hat{x}, \hat{z})$. By design, $(\hat{x}, \hat{y}, \hat{z}, \{\hat{x}^k\}_{k=1}^d)$ satisfies all constraints in R_{extended}, except potentially constraint (8b).

We show that constraint (8b) is satisfied as well. Suppose not for contradiction; that is, $w^k \cdot \hat{x}^k + b^k \hat{z}_k < w^\ell \cdot \hat{x}^k + b^\ell \hat{z}_k$ for some pair $k, \ell \in [\![d]\!], \ell \neq k$. Consider the solution $(\{\overline{x}^k\}_{k=1}^d, \overline{z})$ identical to $(\{\hat{x}^k\}_{k=1}^d, \hat{z})$ except that $\overline{z}_k = 0$, $\overline{x}^k = \mathbf{0}^\eta$, $\overline{z}_\ell = \hat{z}_\ell + \hat{z}_k$, and $\overline{x}^\ell = \hat{x}^\ell + \hat{x}^k$. By inspection, this solution is feasible for $\overline{h}(\hat{x})$. The objective value changes by $-(w^k \cdot \hat{x}^k + b^k \hat{z}_k) + (w^\ell \cdot \hat{x}^k + b^\ell \hat{z}_k) > 0$, contradicting the optimality of $(\{\hat{x}^k\}_{k=1}^d, \hat{z})$. Therefore, $(\hat{x}, \hat{y}, \hat{z}, \{\hat{x}^k\}_{k=1}^d) \in R_{\text{extended}}$, and thus $(\hat{x}, \hat{y}) \in \text{Proj}_{x,y}(R_{\text{extended}})$.

Finally, we observe that the hereditary sharpness property follows immediately from the definition of $\overline{h}$. In particular, fixing any $z_k = 0$ necessarily implies that $\tilde{x}^k = \mathbf{0}^\eta$ in the maximization problem defining $\overline{h}$. In other words, the variables $\tilde{x}^k$ and z_k drop completely from the maximization problem defining $\overline{h}(x, z)$, meaning that it is equal to the corresponding version of $\overline{h}$ with the function k completely dropped as input. Additionally, if any $z_k = 1$, as $z \in \Delta^d$ this implies that for each $\ell \neq k$, $z_\ell = 0$ and hence $\tilde{x}^\ell = \mathbf{0}^\eta$. In this case, $\overline{h}(x, z) = w^k \cdot x + b^k$, which gives the result. □

3.2.2 A dual characterization

We can apply a duality-based approach to produce an (albeit infinite) linear inequality description for the set R_{sharp}, analogous to Proposition 4.

Proposition 6 *The set R_{sharp} is equal to all (x, y, z) such that*

$$y \leqslant \alpha \cdot x + \sum_{k=1}^{d} \left(\max_{x^k \in D} \{(w^k - \alpha) \cdot x^k\} + b^k \right) z_k \quad \forall \alpha \in \mathbb{R}^\eta \tag{13a}$$

$$y \geqslant w^k \cdot x + b^k \quad \forall k \in [\![d]\!] \tag{13b}$$

$$(x, y, z) \in D \times \mathbb{R} \times \Delta^d. \tag{13c}$$

Proof Follows in an analogous manner as in Proposition 4. □

Figure 3 depicts slices of the functions $\overline{g}(x, z)$, $\underline{g}(x, z)$, and $\overline{h}(x, z)$, created by fixing some value of z and varying x. Observe that $\overline{g}(x, z)$ can be viewed as the largest value for y such that (x, y) can be written as a convex combination of points in the graph using convex multipliers z. Likewise, $\underline{g}(x, z)$ can be interpreted as the minimum value for y. In $\overline{h}(x, z)$, we relax $D_{|k}$ to D, and thus we can interpret it similarly to $\overline{g}(x, z)$, except that we may take convex combinations of points constructed by evaluating the affine functions at any point the domain, not only those where the given function attains the maximum. Figure 3b shows that, in general, $\overline{h}(x, z)$ can be strictly looser than $\overline{g}(x, z)$ for $(x, z) \in \text{Proj}_{x,z}(R_{\text{cayley}})$. A similar situation occurs for the lower envelopes as illustrated by Figure 3d. However, we prove in the following section that this does not occur for $d = 2$, along with other desirable properties in special cases.

4 Simplifications to our machinery under common special cases

In this section we study how our dual characterizations in Propositions 4 and 6 simplify under common special cases with the number of input affine functions and the input domain.

4.1 Simplifications when $d = 2$

When we consider taking the maximum of only two affine functions (i.e. $d = 2$), we can prove that R_{sharp} is, in fact, ideal.

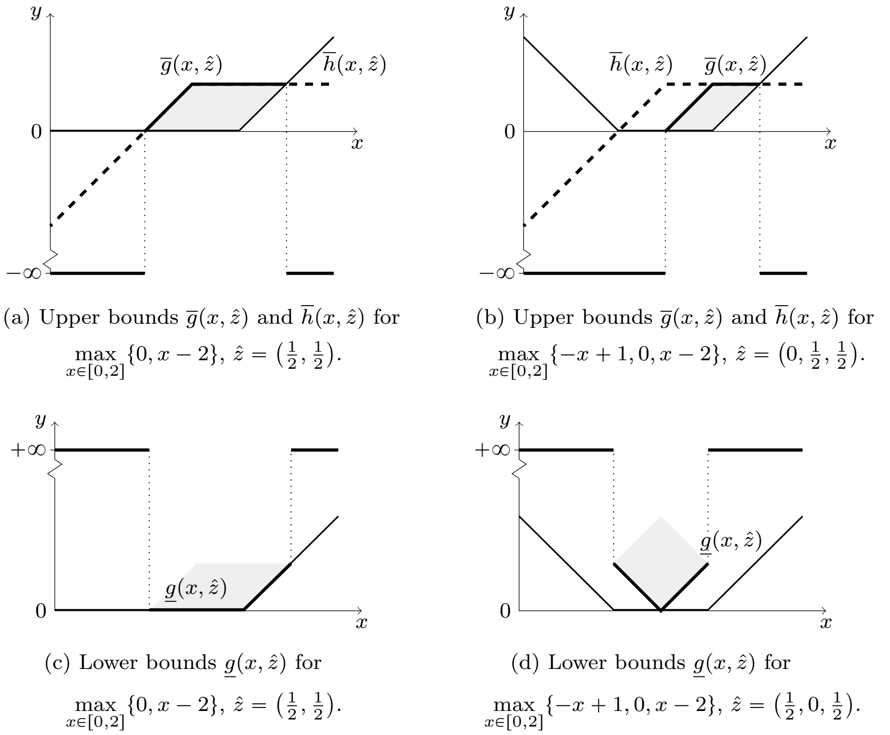
We start by returning to Proposition 3 and show that it can be greatly simplified when $d = 2$. We first show $\underline{g}(x, z)$ can be replaced by the maximum of the affine functions as illustrated in Figure 3c, although it is not possible for $d > 2$ as seen in Figure 3d.

Lemma 1 *If $d = 2$, then at any values of (x, z) where $\overline{g}(x, z) \geqslant \underline{g}(x, z)$ (i.e. there exists a y such that $(x, y, z) \in R_{\text{cayley}}$), we have*

$$\underline{g}(x, z) = \max\{w^1 \cdot x + b^1, w^2 \cdot x + b^2\}.$$

Proof See Appendix A. □

Moreover, we show that when $d = 2$ we can replace $\overline{g}$ with $\overline{h}$, as illustrated in Figure 3a. This property may not hold when $d > 2$ as shown in Figure 3b.



(a) Upper bounds $\overline{g}(x,\hat{z})$ and $\overline{h}(x,\hat{z})$ for $\max\limits_{x\in[0,2]}\{0, x-2\}$, $\hat{z} = \left(\frac{1}{2},\frac{1}{2}\right)$.

(b) Upper bounds $\overline{g}(x,\hat{z})$ and $\overline{h}(x,\hat{z})$ for $\max\limits_{x\in[0,2]}\{-x+1, 0, x-2\}$, $\hat{z} = \left(0,\frac{1}{2},\frac{1}{2}\right)$.

(c) Lower bounds $\underline{g}(x,\hat{z})$ for $\max\limits_{x\in[0,2]}\{0, x-2\}$, $\hat{z} = \left(\frac{1}{2},\frac{1}{2}\right)$.

(d) Lower bounds $\underline{g}(x,\hat{z})$ for $\max\limits_{x\in[0,2]}\{-x+1, 0, x-2\}$, $\hat{z} = \left(\frac{1}{2},0,\frac{1}{2}\right)$.

Fig. 3: Examples of the functions $\overline{g}(x,z)$, $\overline{h}(x,z)$, and $\underline{g}(x,z)$ defined in (9) and (12) with some fixed value for z. Note that $\overline{g}(x,z)$ and $\overline{h}(x,z)$ coincide in (a) for $x \in [1, 2.5]$, and in (b) for $x \in [2, 2.5]$. The thick solid lines represent $\overline{g}(x,z)$ in (a)–(b) and $\underline{g}(x,z)$ in (c)–(d), whereas the dashed lines correspond to $\overline{h}(x,z)$. The thin solid lines represent $S_{\max}$ and the shaded region is the slice of R_{cayley} with $z = \hat{z}$.

Lemma 2 *If $d = 2$, then at any values of (x,z) where $\overline{g}(x,z) \geqslant \underline{g}(x,z)$ (i.e. there exists a y such that $(x,y,z) \in R_{\text{cayley}}$), we have*

$$\overline{g}(x,z) = \max_{\widetilde{x}^1,\widetilde{x}^2}\left\{ w^1\cdot\widetilde{x}^1 + b^1 z_1 + w^2\cdot\widetilde{x}^2 + b^2 z_2 \;\middle|\; \begin{array}{l} x = \widetilde{x}^1 + \widetilde{x}^2 \\ \widetilde{x}^1 \in z_1\cdot D \\ \widetilde{x}^2 \in z_2\cdot D \end{array} \right\} \tag{14}$$

Proof We need to show that although we have expanded the feasible region in the optimization problem on the right-hand side of (14), the objective value is no greater when (x,z) is such that $\overline{g}(x,z) \geqslant \underline{g}(x,z)$. Without loss of generality, we want to show that $w^1\cdot\widetilde{x}^1 + b^1 z_1 \geqslant w^2\cdot\widetilde{x}^1 + b^2 z_1$ is implied for any optimal $\widetilde{x}^1,\widetilde{x}^2$. We use the constraint $y \geqslant w^2\cdot x + b^2$ implied by Lemma 1. Combining this with the constraint $y \leqslant \overline{g}(x,z)$, which implies the existence of some optimal $\widetilde{x}^1,\widetilde{x}^2$, we have $w^2\cdot(x-\widetilde{x}^1) + b^2 z_2 + w^1\cdot\widetilde{x}^1 + b^1 z_1 \geqslant w^2\cdot x + b^2$, which is equivalent to $w^1\cdot\widetilde{x}^1 + b^1 z_1 \geqslant w^2\cdot\widetilde{x}^1 + b^2 z_1$. □

After observing that these simplifications are identical to those presented in Proposition 6, we obtain the following corollary promised at the beginning of the section.

Corollary 1 *When* $d = 2$*,* $\left\{ (x, y, z) \in R_{\text{sharp}} \;\middle|\; z \in \{0,1\}^d \right\}$ *is an ideal MIP formulation of* $S_{\max}$*.*

Proof Lemmas 1 and 2 imply that $R_{\text{sharp}} = R_{\text{ideal}}$, while Proposition 3 implies that $R_{\text{ideal}} = R_{\text{cayley}}$, completing the chain and giving the result. □

In later sections, we will study conditions under which we can produce an explicit inequality description for R_{sharp}.

4.2 Simplifications when D is the product of simplices

In this section, we consider another important special case: when the input domain is the Cartesian product of simplices. Indeed, the box domain case introduced in Section 2 can be viewed as a product of two-dimensional simplices, and we will also see in Section 5.3 that this structure naturally arises in machine learning settings with categorical or discrete features.

When D is the product of simplices, we can derive a finite representation for the the set (13) (i.e. a finite representation for the infinite family of linear inequalities (13a)) through an elegant connection with the transportation problem. To do so, we introduce the following notation.

Definition 2 Suppose the input domain is $D = \prod_{i=1}^{\tau} \Delta^{p_i}$, with $p_1 + \cdots + p_\tau = \eta$. For notational simplicity, we re-organize the indices of x and refer to its entries via $x_{i,j}$, where $i \in [\![\tau]\!]$ is the simplex index, and $j \in [\![p_i]\!]$ refers to the coordinate within simplex i. The domain for x is then

$$D = \left\{ ((x_{i,j})_{j=1}^{p_i})_{i=1}^{\tau} \;\middle|\; (x_{i,j})_{j=1}^{p_i} \in \Delta^{p_i} \quad \forall i \in [\![\tau]\!] \right\}, \tag{15}$$

where the rows of x correspond to each simplex.

Correspondingly, we re-index the weights of the affine functions so that for each function $k \in [\![d]\!]$, we have $f^k(x) = \sum_{i=1}^{\tau} \sum_{j=1}^{p_i} w_{i,j}^k x_{i,j} + b^k$.

Using the notation from Definition 2, constraints (13a) can be written as

$$y \leqslant \sum_{i=1}^{\tau} \sum_{j=1}^{p_i} \alpha_{i,j} x_{i,j} + \sum_{k=1}^{d} \left(\max_{x^k \in D} \sum_{i=1}^{\tau} \sum_{j=1}^{p_i} (w_{i,j}^k - \alpha_{i,j}) x_{i,j}^k + b^k \right) z_k \quad \forall \alpha \in \mathbb{R}^\eta.$$

Since D is a product of simplices, the maximization over $x^k \in D$ appearing in the right-hand side above is separable over each simplex $i \in [\![\tau]\!]$. Moreover, for each simplex i, the maximum value of $\sum_{j=1}^{p_i} (w_{ij}^k - \alpha_{ij}) x_{ij}^k$, subject to the constraint $x^k \in D$, is obtained when $x^k_{ij} = 1$ for some $j \in [\![p_i]\!]$. Therefore, the family of constraints (13a) is equivalent to

$$\begin{aligned} y &\leqslant \min_{\alpha} \left(\sum_{i=1}^{\tau} \sum_{j=1}^{p_i} \alpha_{i,j} x_{i,j} + \sum_{k=1}^{d} \left(\sum_{i=1}^{\tau} \max_{j=1}^{p_i} (w^k_{i,j} - \alpha_{i,j}) + b^k \right) z_k \right) \\ &= \sum_{i=1}^{\tau} \min_{\alpha_{i,1},\dots,\alpha_{i,p_i}} \left(\sum_{j=1}^{p_i} \alpha_{i,j} x_{i,j} + \sum_{k=1}^{d} z_k \cdot \max_{j=1}^{p_i} (w^k_{i,j} - \alpha_{i,j}) \right) + \sum_{k=1}^{d} b^k z_k. \quad (16) \end{aligned}$$

We show that the minimization problem in (16), for any i, is equivalent to a transportation problem defined as follows.

Definition 3 For any values $x \in \Delta^p$ and $z \in \Delta^d$, and arbitrary weights $w^k_j \in \mathbb{R}$ for all $j \in [\![p]\!]$ and $k \in [\![d]\!]$, define the *max-weight transportation problem* to be

$$\begin{aligned} \mathsf{Transport}(x, z; w^1, \dots, w^d) \overset{\text{def}}{=} \max_{\beta} \quad & \sum_{k=1}^{d} \sum_{j=1}^{p} w^k_j \beta^k_j \\ \text{s.t.} \quad & \sum_{k=1}^{d} \beta^k_j = x_j \qquad \forall j \in [\![p]\!] \\ & \sum_{j=1}^{p} \beta^k_j = z_k \qquad \forall k \in [\![d]\!] \\ & \beta^k_j \geqslant 0 \qquad \forall j \in [\![p]\!], k \in [\![d]\!] \end{aligned}$$

In the transportation problem, since $\sum_j x_j = 1 = \sum_k z_k$, it follows that $\beta^k_j \in [0, 1]$, and so this value can be interpreted as the percent of total flow "shipped" between j and k. The relation to equation (16) is now established through LP duality.

Proposition 7 *For any fixed $x \in \Delta^p$ and $z \in \Delta^d$,*

$$\min_{\alpha} \left(\sum_{j=1}^{p} \alpha_j x_j + \sum_{k=1}^{d} z_k \cdot \max_{j=1}^{p} (w^k_j - \alpha_j) \right) = \mathsf{Transport}(x, z; w^1, \dots, w^d). \quad (17)$$

Therefore, when D is a product of simplices, the constraints (13a) can be replaced in (13) with the single inequality

$$y \leqslant \sum_{i=1}^{\tau} \mathsf{Transport}((x_{i,j})_{j=1}^{p_i}, z; (w^1_{i,j})_{j=1}^{p_i}, \dots, (w^d_{i,j})_{j=1}^{p_i}) + \sum_{k=1}^{d} b^k z_k. \quad (18)$$

Proof By using a variable γ_k to model the value of $\max_{j=1}^{p}(w^k_j - \alpha_j)$ for each $k \in [\![d]\!]$, the minimization problem on the LHS of (17) is equivalent to

$$\begin{aligned} \min_{\alpha,\gamma} \quad & \sum_{j=1}^{p} \alpha_j x_j + \sum_{k=1}^{d} \gamma_k z_k \\ \text{s.t.} \quad & \gamma_k \geqslant w^k_j - \alpha_j \qquad \forall k \in [\![d]\!], j \in [\![p]\!] \end{aligned}$$

which is a minimization LP with free variables α_j and γ_k. Applying LP duality, this completes the proof of equation (17). (18) then arises by substituting equation (17) into (16), for every simplex $i = 1, \dots, \tau$. □

4.3 Simplifications when both $d = 2$ and D is the product of simplices

Proposition 7 shows that, when the input domain is a product of simplices, the tightest upper bound on y can be computed through a series of transportation problems. We now leverage the fact that if either side of the transportation problem from Definition 3 (i.e. p or d) has only two entities, then it reduces to a simpler fractional knapsack problem. Later, this will allow us to represent (16), in either of the cases $d = 2$ or $p_1 = \cdots = p_\tau = 2$, using an explicit finite family of linear inequalities in x and z which has a greedy linear-time separation oracle.

Proposition 8 *Given data $w^1, w^2 \in \mathbb{R}^p$, take $\widetilde{w}_j = w^1_j - w^2_j$ for all $j \in [\![p]\!]$, and suppose the indices have been sorted so that $\widetilde{w}_1 \leqslant \cdots \leqslant \widetilde{w}_p$. Then*

$$\mathsf{Transport}(x, z; w^1, w^2) = \min_{J=1}^{p} \left(\widetilde{w}_J z_1 + \sum_{j=J+1}^{p} (\widetilde{w}_j - \widetilde{w}_J) x_j \right) + \sum_{j=1}^{p} w^2_j x_j. \quad (19)$$

Moreover, a $J \in [\![p]\!]$ that attains the minimum in the right-hand side of (19) can be found in $\mathcal{O}(p)$ time.

Proof When $d = 2$, the transportation problem becomes

$$\begin{aligned} \max_{\beta^1, \beta^2} \quad & \sum_{j=1}^{p} (w^1_j \beta^1_j + w^2_j \beta^2_j) \\ \text{s.t.} \quad & \beta^1_j + \beta^2_j = x_j && \forall j \in [\![p]\!] \\ & \sum_{j=1}^{p} \beta^1_j = z_1 \\ & \beta^1_j, \beta^2_j \geqslant 0 && \forall j \in [\![p]\!], \end{aligned}$$

where the constraint $\sum_{j=1}^{p} \beta^2_j = z_2$ is implied because

$$\sum_{j=1}^{p} \beta^2_j = \sum_{j=1}^{p} (x_j - \beta^1_j) = 1 - \sum_{j=1}^{p} \beta^1_j = 1 - z_1 = z_2.$$

Substituting $\beta^2_j = x_j - \beta^1_j$ for all $j \in [\![p]\!]$ and then omitting the superscript "1", the above optimization problem becomes

$$\begin{aligned} \max_{\beta} \quad & \sum_{j=1}^{p} (w^1_j - w^2_j) \beta_j + \sum_{j=1}^{p} w^2_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{p} \beta_j = z_1 \\ & 0 \leqslant \beta_j \leqslant x_j && \forall j \in [\![p]\!], \end{aligned}$$

which is a fractional knapsack problem that can be solved greedily.

An optimal solution to the fractional knapsack LP above, assuming the sorting $w^1_1 - w^2_1 \leqslant \cdots \leqslant w^1_p - w^2_p$, takes the following greedy form. Let $J \in \llbracket p \rrbracket$ be the *maximum* index at which $\sum_{j=J}^p x_j \geqslant z_1$. We set

$$\beta_j = \begin{cases} 0 & j < J \\ z_1 - \sum_{j=J+1}^p x_j & j = J \\ x_j & j > J \end{cases}.$$

The optimal objective value is

$$\sum_{j=J+1}^{p} (w^1_j - w^2_j)x_j + (w^1_J - w^2_J)\left(z_1 - \sum_{j=J+1}^{p} x_j\right) + \sum_{j=1}^{p} w^2_j x_j,$$

which yields the desired expression after substituting $\widetilde{w}_j = w^1_j - w^2_j$ for all $j \geqslant J$. Moreover, the index J above can be found in $\mathcal{O}(p)$ time by storing a running total for $\sum_{j=J}^p x_j$, completing the proof. □

Observe that the $\mathcal{O}(p)$ runtime in Proposition 8 is non-trivial, as a naïve implementation would run in time $\mathcal{O}(p^2)$, as the inner summation is linear in p.

Combining Propositions 7 and 8 immediately yields the following result.

Corollary 2 *Suppose that $d = 2$ and that D is a product of simplices. Let $z = z_1 \equiv 1 - z_2$. For each simplex $i \in \llbracket \tau \rrbracket$, take $\widetilde{w}_{i,j} = w^1_{i,j} - w^2_{i,j}$ for all $j = 1, \ldots, p_i$ and relabel the indices so that $\widetilde{w}_{i,1} \leqslant \cdots \leqslant \widetilde{w}_{i,p_i}$. Then, in the context of* (13)*, the upper-bound constraints* (13a) *are equivalent to*

$$y \leqslant \sum_{i=1}^{\tau} \left(\widetilde{w}_{i,J(i)} z + \sum_{j=J(i)+1}^{p_i} (\widetilde{w}_{i,j} - \widetilde{w}_{i,J(i)}) x_{i,j} + \sum_{j=1}^{p_i} w^2_{i,j} x_{i,j} \right) + (b^1 - b^2)z + b^2$$
$$\forall \textit{ mappings } J : \llbracket \tau \rrbracket \to \mathbb{Z} \textit{ with } J(i) \in \llbracket p_i \rrbracket \ \forall i \in \llbracket \tau \rrbracket. \quad (20)$$

Moreover, given any point $(x, y, z) \in D \times \mathbb{R} \times [0, 1]$, feasibility can be verified or a most violated constraint can be found in $\mathcal{O}(p_1 + \cdots + p_\tau)$ time.

Corollary 2 gives an explicit finite family of linear inequalities equivalent to (13). Moreover, we have already shown in Corollary 1 that R_{sharp} yields an ideal formulation when $d = 2$. Hence, we have an ideal nonextended formulation whose exponentially-many inequalities can be separated in $\mathcal{O}(p_1 + \cdots + p_\tau)$ time, where the initial sorting requires $\mathcal{O}(p_1 \log p_1 + \cdots + p_\tau \log p_\tau)$ time. Note that this sorting can be avoided: we may instead solve the fractional knapsack problem in the separation via weighted median in $\mathcal{O}(p_1 + \cdots + p_\tau)$ time [44, Chapter 17.1].

We can also show that none of the constraints in (20) are redundant.

Proposition 9 *Consider the polyhedron P defined as the intersection of all halfspaces corresponding to the inequalities* (20). *Consider some arbitrary mapping $J : [\![\tau]\!] \to \mathbb{Z}$ with $J(i) \in [\![p_i]\!]$ for each $i \in [\![\tau]\!]$. Then the inequality in* (20) *for J is irredundant with respect to P. That is, removing the halfspace corresponding to mapping J (note that this halfspace could also correspond to other mappings) from the description of P will strictly enlarge the feasible set.*

Proof See Appendix B. □

To close this section, we consider the setting where every simplex is 2-dimensional, i.e. $p_1 = \cdots = p_\tau = 2$, but the number of affine functions d can be arbitrary. Note that by contrast, the previous results (Proposition 8, Corollary 2, Proposition 9) held in the setting where $d = 2$ but $p_1, \dots, p_\tau$ were arbitrary. We exploit the symmetry of the transportation problem to immediately obtain the following analogous results, all wrapped up into Corollary 3.

Corollary 3 *Given data $w^1, \dots, w^d \in \mathbb{R}^2$, take $\widetilde{w}^k = w^k_1 - w^k_2$ for all $k \in [\![d]\!]$, and suppose the indices have been sorted so that $\widetilde{w}^1 \leqslant \cdots \leqslant \widetilde{w}^d$. Then*

$$\mathsf{Transport}(x, z; w^1, \dots, w^d) = \min_{K=1}^{d} \left(\widetilde{w}^K x_1 + \sum_{k=1}^{K} w^k_2 z_k + \sum_{k=K+1}^{d} (w^k_1 - \widetilde{w}^K) z_k \right). \tag{21}$$

Moreover, a $K \in [\![d]\!]$ that attains the minimum in the right-hand side of (21) *can be found in $\mathcal{O}(d)$ time.*

Therefore, suppose that D is a product of τ simplices of dimensions $p_1 = \cdots = p_\tau = 2$. For each simplex $i \in [\![\tau]\!]$, take $\widetilde{w}^k_i = w^k_{i,1} - w^k_{i,2}$ for all $k = 1, \dots, d$ and relabel the indices so that $\widetilde{w}^1_i \leqslant \cdots \leqslant \widetilde{w}^d_i$. Then, in the context of (13), *the upper-bound constraints* (13a) *are equivalent to*

$$y \leqslant \sum_{i=1}^{\tau} \left(\widetilde{w}^{K(i)}_i x_{i,1} + \sum_{k=1}^{K(i)} w^k_{i,2} z_k + \sum_{k=K(i)+1}^{d} (w^k_{i,1} - \widetilde{w}^{K(i)}_i) z_k \right) + \sum_{k=1}^{d} b^k z_k. \qquad \forall \text{ mappings } K : [\![\tau]\!] \to [\![d]\!]. \tag{22}$$

Furthermore, none of the constraints in (22) *are redundant.*

Corollary 3 will be particularly useful in Section 5.1.2, where it will allow us to derive sharp formulations for the maximum of d affine functions over a box input domain, in analogy to (20).

5 Applications of our machinery

We are now prepared to return to the concrete goal of this paper: building strong MIP formulations for nonlinearities which are prevalent in modern neural network architectures.

5.1 Practical formulations and cuts for the Max nonlinearity

We turn our attention to the regime where $d > 2$, with the goal of producing practically useful (i.e. explicit, finite descriptions of) strong MIP formulations for the maximum of d affine functions.

5.1.1 A tight big-M formulation

We start by presenting a tightened big-M formulation for the maximum of d affine functions over an arbitrary polytope input domain. We can view the formulation as a relaxation of the system in Proposition 4, where we select d inequalities from each of (10a) and (10b): those corresponding to $\overline{\alpha}, \underline{\alpha} \in \{w^1, \ldots, w^d\}$. This subset yields a valid formulation, and we obviate the need for direct separation. This formulation can also be viewed as an application of Proposition 6.2 of Vielma [66], and is similar to the big-M formulations for generalized disjunctive programs of Trespalacios and Grossmann [65].

Proposition 10 *Take coefficients N such that, for each $\ell, k \in \llbracket d \rrbracket$ with $\ell \neq k$,*

$$N^{\ell,k,+} \geqslant \max_{x^k \in D_{|k}} \{(w^k - w^\ell) \cdot x^k\} \tag{23a}$$

$$N^{\ell,k,-} \leqslant \min_{x^k \in D_{|k}} \{(w^k - w^\ell) \cdot x^k\}, \tag{23b}$$

and $N^{k,k,+} = N^{k,k,-} = 0$ for all $k \in \llbracket d \rrbracket$.

Then a valid MIP formulation for $\operatorname{gr}(\texttt{Max} \circ (f^1, \ldots, f^d); D)$ *is:*

$$y \leqslant w^\ell \cdot x + \sum_{k=1}^{d} (N^{\ell,k,+} + b^k) z_k \quad \forall \ell \in \llbracket d \rrbracket \tag{24a}$$

$$y \geqslant w^\ell \cdot x + \sum_{k=1}^{d} (N^{\ell,k,-} + b^k) z_k \quad \forall \ell \in \llbracket d \rrbracket \tag{24b}$$

$$(x, z) \in D \times \Delta^d \tag{24c}$$

$$z \in \{0, 1\}^d \tag{24d}$$

Proof We enumerate each value z may take subject to (24c–24d). Consider the case where $z = \mathbf{e}^k$; this corresponds to the case where the k-th input function is the maximum, and so $(\hat{x}, \hat{y}, \mathbf{e}^k)$ satisfying (24) implies that $\hat{x} \in D_{|k}$ and that $\hat{y} = f^k(\hat{x}) \geqslant f^\ell(\hat{x})$ for each $\ell \in \llbracket d \rrbracket$. Then for $\ell = k$, (24a) and (24b) produce the desired equation $y = w^\ell \cdot x + b^\ell$. Furthermore, for each $\ell \neq k$, (24a) reduces to $\hat{y} \leqslant w^\ell \cdot x + N^{\ell,k,+} + b^k$. The validity of this inequality follows from

$$\begin{aligned}
\hat{y} &= f^k(\hat{x}) \\
&= f^\ell(\hat{x}) + (f^k(\hat{x}) - f^\ell(\hat{x})) \\
&\leqslant f^\ell(\hat{x}) + \max_{x \in D_{|k}} (f^k(x^k) - f^\ell(x)) \\
&= w^\ell \cdot \hat{x} + \max_{x \in D_{|k}} ((w^k - w^\ell) \cdot x) + b^k \\
&\leqslant w^\ell \cdot \hat{x} + N^{\ell,k,+} + b^k.
\end{aligned}$$

where the first inequality follows as $\hat{x} \in D_{|k}$. Similarly, (24b) for $\ell \neq k$ reduces to $\hat{y} \geqslant w^\ell \cdot x + N^{\ell,k,-} + b^k$, whose validity follows by the same argument above except that we flip the direction of inequalities and exchange max by min and $N^{\ell,k,+}$ by $N^{\ell,k,-}$.

□

There are number of observations to be made about the formulation (24).

First, the coefficients $N^{\ell,k,+}$ and and $N^{\ell,k,-}$ are the "big-M coefficients" that would need to be computed to construct a standard big-M formulation, modulo the constant offset terms b^k.

Second, the tightest possible coefficients in (23) can be computed exactly by solving an LP for each pair of input affine functions $\ell \neq k$. While this might be exceedingly computationally expensive if d is large, it is potentially viable if d is a small fixed constant. For example, the max pooling neuron computes the maximum over a rectangular window in a larger array [31, Section 9.3], and is frequently used in image classification architectures. Typically, max pooling units work with a 2×2 or a 3×3 window, in which case $d = 4$ or $d = 9$, respectively.

Third, if in practice we observe that if the set $D_{|k}$ is empty, then we can infer that the neuron is not irreducible as the k-th input function is never the maximum, and we can safely prune it. In particular, if we attempt to compute the coefficients for z_k and it is proven infeasible, we can prune the k-th function.

Fourth, in the case where $D = [L, U]$ and d is sufficiently large that we cannot tractably compute the tightest coefficients in (23), we can readily produce valid coefficients by relaxing the constraint $\tilde{x} \in D_{|\ell}$ to $\tilde{x} \in [L, U]$. We can then compute the corresponding values by inspecting the values of the input functions at the corners of the domain $[L, U]$:

$$N^{\ell,k,+} = \sum_{i=1}^{\eta} \max\{(w_i^k - w_i^\ell)L_i, (w_i^k - w_i^\ell)U_i)\} \tag{25a}$$

$$N^{\ell,k,-} = \sum_{i=1}^{\eta} \min\{(w_i^k - w_i^\ell)L_i, (w_i^k - w_i^\ell)U_i)\} \tag{25b}$$

for each $\ell \neq k$. All coefficients for formulation (24) can then be computed in time $\mathcal{O}(d^2 \cdot \eta)$. Although potentially suboptimal, they may still lead to a big-M formulation that is substantially stronger than those in the literature. For example, the tightest version of the formulation of Tjeng et al. [64] is equivalent in our framework to selecting for each $\ell \neq k$:

$$\begin{aligned} N^{\ell,k,+} &= b^\ell - b^k + \max_{t \neq \ell} \left(\max_{\tilde{x} \in [L,U]} w^t \cdot \tilde{x} - \min_{\mathring{x} \in [L,U]} w^\ell \cdot \mathring{x} \right) \\ N^{\ell,k,-} &= b^\ell - b^k. \end{aligned}$$

Note in particular that as the inner maximization and minimization are completely decoupled, and that the outer maximization in the definition of $N^{\ell,k,+}$ is completely independent of k.

5.1.2 An efficiently-separable sharp formulation for `Max` *on box domains*

In the case where $D = [L,U]$ is a box, we can construct a sharp formulation with exponentially many constraints that can be efficiently separated. Note the similarity between the coefficients in the constraints below, and the valid big-M coefficients (25) computed by inspecting the corners of the input box domain.

Proposition 11 *A sharp formulation for* $\text{gr}(\texttt{Max} \circ (f^1, \ldots, f^d); [L,U])$ *is*

$$y \leqslant \sum_{i=1}^{\eta} \left(w_i^{I(i)} x_i + \sum_{k=1}^{d} \max\{(w_i^k - w_i^{I(i)})L_i, (w_i^k - w_i^{I(i)})U_i\} z_k \right) + \sum_{k=1}^{d} b^k z_k \quad \forall \textit{ mappings } I : [\![\eta]\!] \to [\![d]\!] \tag{27a}$$

$$y \geqslant w^k \cdot x + b^k \quad \forall k \in [\![d]\!] \tag{27b}$$

$$(x,y,z) \in [L,U] \times \mathbb{R} \times \Delta^d \tag{27c}$$

$$z \in \{0,1\}^d. \tag{27d}$$

Furthermore, none of the constraints in the families (27a) *and* (27b) *are redundant.*

Proof Follows from Corollary 3 after a transformation of variables; see Appendix C. □

Given $\ell \in [\![d]\!]$, we can recover the inequalities (24a) with big-M values in (25) if we set $I(i) = \ell$ for all $i \in [\![\eta]\!]$. Therefore, a practical approach to using this sharp formulation would be to start with the big-M formulation (24), and separating from the remaining inequalities (27a) as-needed.

We emphasize that this formulation is particularly strong when $d = 2$.

Corollary 4 *Formulation* (27) *is ideal when* $d = 2$. *Moreover, the constraints in the families* (27a) *and* (27b) *are facet-defining.*

Proof Idealness follows directly from Corollary 1. Since the constraints (27a) and (27b) are irredundant and the formulation is ideal, they must either be facet-defining or describe an implied equality. Given that the equality $\sum_{k=1}^{d} z_k = 1$ appears in (27c), it suffices to observe that the polyhedron defined by (27) has dimension $\eta + d$, which holds under Assumption 2. □

We next show how to compute a most-violated inequality from the family (27a) efficiently.

Proposition 12 *Consider the family of inequalities* (27a)*. Take some point* $(\hat{x}, \hat{y}, \hat{z}) \in [L, U] \times \mathbb{R} \times \Delta^d$*. If any constraint in the family is violated at the given point, a most-violated constraint can be constructed by selecting* $\hat{I} : [\![\eta]\!] \to [\![d]\!]$ *such that*

$$\hat{I}(i) \in \arg\min_{\ell \in [\![d]\!]} \left(w_i^\ell \hat{x}_i + \sum_{k=1}^{d} \max\{(w_i^k - w_i^\ell) L_i, (w_i^k - w_i^\ell) U_i\} \hat{z}_k \right) \tag{28}$$

for each $i \in [\![\eta]\!]$*. Moreover, if the weights* w_i^k *are sorted on* k *for each* $i \in [\![\eta]\!]$*, this can be done in* $\mathcal{O}(\eta d)$ *time.*

Proof Follows directly from Corollary 3, which says that the minimization problem (28) can be solved in $\mathcal{O}(d)$ time for any $i \in [\![\eta]\!]$. □

Note that naïvely, the minimization problem (28) would take $\mathcal{O}(d^2)$ time, because one has to check every $\ell \in [\![d]\!]$, and then sum over $k \in [\![d]\!]$ for every ℓ. However, if we instead pre-sort the weights $w_i^1, \ldots, w_i^d$ for every $i \in [\![\eta]\!]$ in $\mathcal{O}(\eta d \log d)$ time, we can use Corollary 3 to run efficiently separate via a linear search. We note, however, that this pre-sorting step can potentially be obviated by solving the fractional knapsack problems appearing as a weighted median problem, which can be solved in $\mathcal{O}(d)$ time.

5.2 The ReLU over a box domain

We can now present the results promised in Theorem 2.3. In particular, we derive a non-extended ideal formulation for the ReLU nonlinearity, stated only in terms of the original variables (x, y) and the single additional binary variable z. Put another way, it is the strongest possible tightening that can be applied to the big-M formulation (5), and so matches the strength of the multiple choice formulation without the growth in the number of variables remarked upon in Section 2.2. Notationally, for each $i \in [\![\eta]\!]$ take

$$\breve{L}_i = \begin{cases} L_i & \text{if } w_i \geqslant 0 \\ U_i & \text{if } w_i < 0 \end{cases} \quad \text{and} \quad \breve{U}_i = \begin{cases} U_i & \text{if } w_i \geqslant 0 \\ L_i & \text{if } w_i < 0 \end{cases}.$$

Proposition 13 *Take some affine function $f(x) = w \cdot x + b$ over input domain $D = [L, U]$. The following is an ideal MIP formulation for* $\operatorname{gr}(\texttt{ReLU} \circ f; [L, U])$*:*

$$y \leqslant \sum_{i \in I} w_i(x_i - \breve{L}_i(1-z)) + \left(b + \sum_{i \notin I} w_i \breve{U}_i\right) z \quad \forall I \subseteq [\![\eta]\!] \tag{29a}$$

$$y \geqslant w \cdot x + b \tag{29b}$$

$$(x, y, z) \in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1] \tag{29c}$$

$$z \in \{0, 1\}. \tag{29d}$$

Furthermore, each inequality in (29a) and (29b) is facet-defining.

Proof [5] This result is a special case of Corollary 4, which refers to the formulation in Proposition 11. This includes the result that the inequalities in (29a) and (29b) are facet-defining.

To see why it is a special case, consider the notation of Proposition 11 and let $w_i^1 = w_i$ and $w_i^2 = 0$ for all $i \in [\![\eta]\!]$, $b^1 = b$, $b^2 = 0$, $z_1 = z$, and $z_2 = 1 - z$. Note that the constraint $y \geqslant 0$ from (27b) is implied by the domain restriction to $\mathbb{R}_{\geqslant 0}$ in (29c). Any constraint in (27a) defined by a mapping $I : [\![\eta]\!] \to [\![2]\!]$ can be simplified to

$$\begin{aligned} y &\leqslant \sum_{i=1}^{\eta} \Big(w_i^{I(i)} x_i + \max\{(w_i - w_i^{I(i)})L_i, (w_i - w_i^{I(i)})U_i\} z \\ &\qquad\qquad + \max\{-w_i^{I(i)} L_i, -w_i^{I(i)} U_i\}(1-z) \Big) + bz \\ &= \sum_{i : I(i) = 1} (w_i x_i - \min\{w_i L_i, w_i U_i\}(1-z)) + \sum_{i : I(i) = 2} \max\{w_i L_i, w_i U_i\} z + bz \\ &= \sum_{i : I(i) = 1} w_i(x_i - \breve{L}_i(1-z)) + \left(\sum_{i : I(i) = 2} w_i \breve{U}_i + b \right) z \end{aligned}$$

by the definitions of $\breve{L}_i$ and $\breve{U}_i$. Therefore, the corresponding constraint in (29a) is found by setting $I = \{\, i \in [\![\eta]\!] \mid I(i) = 1 \,\}$.

□

Formulation (29) has a number of constraints exponential in the input dimension η, so it will not be useful directly as a MIP formulation. However, it is straightforward to separate the exponential family (29a) dynamically, as-needed.

Proposition 14 *Take a point $(\hat{x}, \hat{y}, \hat{z}) \in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1]$, along with the set*

$$\hat{I} = \left\{ i \in [\![\eta]\!] \;\middle|\; w_i \hat{x}_i < w_i \left(\breve{L}(1 - \hat{z}) + \breve{U}_i \hat{z} \right) \right\}.$$

[5] Alternatively, a constructive proof of validity and idealness using Fourier–Motzkin elimination is given in the extended abstract of this work [3, Proposition 1].

If

$$\hat{y} > b\hat{z} + \sum_{i \in \hat{I}} w_i \left(\hat{x}_i - \breve{L}(1 - \hat{z}) \right) + \sum_{i \notin \hat{I}} w_i \breve{U}_i \hat{z},$$

then the constraint in (29a) *corresponding to* $\hat{I}$ *is the most violated in the family. Otherwise, no inequality in the family is violated at* $(\hat{x}, \hat{y}, \hat{z})$*.*

Proof Follows as a special case of Proposition 12. □

Furthermore, note that (5b) and (5c) correspond to (29a) with $I = [\![\eta]\!]$ and $I = \emptyset$, respectively. All this suggests an iterative approach to formulating ReLU neurons over box domains: start with the big-M formulation (5), and use Proposition 14 to separate strengthening inequalities from the exponential family (29a) as they are needed.

5.3 The ReLU with one-hot encodings

Although box domains are a natural choice for many applications, it is often the case that some (or all) of the first layer of a neural network will be constrained to be the product of simplices. The *one-hot encoding* is a standard technique used in the machine learning community to preprocess discrete or categorical data to a format more amenable for learning (see, for example, [15, Chapter 2.2]). More formally, if input x is constrained to take categorical values $x \in C = \{c^1, \dots, c^t\}$, the one-hot transformation encodes this as $\tilde{x} \in \{0,1\}^t$, where $\tilde{x}_i = 1$ if and only if $x = c^i$. In other words, the input is constrained such that $\tilde{x} \in \underline{\Delta}^\eta \overset{\text{def}}{=} \Delta^\eta \cap \{0,1\}^\eta$.

It is straightforward to construct a small ideal formulation for $\text{gr}(\texttt{ReLU} \circ f; \underline{\Delta}^\eta)$ as $\{\, (x, \sum_{i=1}^\eta \max\{0, w_i x_i + b\}) \mid x \in \underline{\Delta}^\eta \,\}$. However, it is typically the case that multiple features will be present in the input, meaning that the input domain would consist of the product of (potentially many) simplices. For example, neural networks have proven extremely well-suited for predicting the propensity for a given DNA sequence to bind with a given protein [1,76], where the network input consists of a sequence of n base pairs, each of which can take 4 possible values. In this context, the input domain would be $\prod_{i=1}^n \underline{\Delta}^4$.

In this section, we restate the general results presented in Section 2, specialized for the standard case of the ReLU nonlinearity.

Corollary 5 *Presume that the input domain* $D = \Delta^{p_1} \times \cdots \times \Delta^{p_\tau}$ *is a product of* τ *simplices, and that* $f(x) = \sum_{i=1}^\tau \sum_{j=1}^{p_i} w_{i,j} x_{i,j} + b$ *is an affine function. Presume that, for each* $i \in [\![\tau]\!]$*, the weights are sorted such that* $w_{i,1} \leqslant \cdots \leqslant$

w_{i,p_i}. *Then an ideal formulation for* $\text{gr}(\texttt{ReLU} \circ f; D)$ *is:*

$$y \geqslant w \cdot x + b \tag{30a}$$

$$y \leqslant \sum_{i=1}^{\tau} \left(w_{i,J(i)} z + \sum_{j=J(i)+1}^{p_i} (w_{i,j} - w_{i,J(i)}) x_{i,j} \right) + bz \tag{30b}$$
$$\forall \textit{ mappings } J : [\![\tau]\!] \to \mathbb{Z} \textit{ with } J(i) \in [\![p_i]\!] \ \forall i \in [\![\tau]\!]$$

$$(x, y, z) \in D \times \mathbb{R}_{\geqslant 0} \times \{0, 1\}. \tag{30c}$$

Moreover, a most-violated constraint from the family (30b)*, if one exists, can be identified in* $\mathcal{O}(p_1 + \cdots + p_\tau)$ *time. Finally, none of the constraints from* (30b) *are redundant.*

Proof Follows directly from applying Corollary 2 to the set R_{sharp}. By Corollary 1, this set actually leads to an ideal formulation, because we are taking the maximum of only two functions (with one of them being zero). The statement about non-redundancy follows from Proposition 9. □

5.4 The leaky ReLU over a box domain

A slightly more exotic variant of the ReLU is the *leaky ReLU*, defined as $\texttt{Leaky}(v; \alpha) = \max\{\alpha v, v\}$ for some constant $0 < \alpha < 1$. Instead of fixing any negative input to zero, the leaky ReLU scales it by a (typically quite small) constant α. This alteration has been empirically observed to help avoid the "vanishing gradient" problem during the training of certain deep learning models [50,75]. We can present analogous results for the leaky ReLU as for the ReLU: an ideal MIP formulation with exponentially-many linear inequality constraints which can be efficiently separated.

Proposition 15 *Take some affine function* $f(x) = w \cdot x + b$ *over input domain* $D = [L, U]$. *The following is a valid formulation for* $\text{gr}(\texttt{Leaky} \circ f; [L, U])$*:*

$$y \geqslant f(x) \tag{31a}$$
$$y \geqslant \alpha f(x) \tag{31b}$$
$$y \leqslant f(x) - (1 - \alpha) \cdot M^-(f; [L, U]) \cdot (1 - z) \tag{31c}$$
$$y \leqslant \alpha f(x) - (\alpha - 1) \cdot M^+(f; [L, U]) \cdot z \tag{31d}$$
$$(x, y, z) \in [L, U] \times \mathbb{R} \times [0, 1] \tag{31e}$$
$$z \in \{0, 1\}. \tag{31f}$$

Moreover, an ideal formulation is given by (31)*, along with the constraints*

$$y \leqslant \left(\sum_{i \in I} w_i (x_i - \breve{L}_i (1 - z)) + \left(b + \sum_{i \notin I} w_i \breve{U}_i \right) z \right)$$
$$+ \alpha \left(\sum_{i \notin I} w_i (x_i - \breve{U}_i z) + \left(b + \sum_{i \in I} w_i \breve{L}_i \right) (1 - z) \right) \quad \forall I \subseteq [\![\eta]\!]. \tag{32}$$

Additionally, the most violated inequality from the family (32) *can be separated in* $\mathcal{O}(\eta)$ *time. Finally, each inequality in* (31a–31d) *and* (32) *is facet-defining.*

Proof Follows as a special case of Corollary 4, which refers to the formulation in Proposition 11. □

5.5 The clipped ReLU

We close this section by studying a nonlinearity that falls just outside our framework, but for which we can nonetheless derive similar results. The clipped ReLU nonlinearity is a modification of the standard ReLU unit which caps its output at some constant positive value C:

$$\texttt{Clipped}(v; C) \overset{\text{def}}{=} \min\{C, \max\{0, v\}\}.$$

Clipped ReLUs were introduced in the deep learning community as a tool to avoid oversaturation during the training procedure, particularly for recurrent networks [33]. The `relu6` activation function in TensorFlow is a special case where $C = 6$. Additionally, clipped ReLUs are attractive in our convex relaxation framework, as the cap C allows bound propagation through deep networks to be regulated, which in turn can lead to tighter convex relaxations.

We also observe that the clipped ReLU is very similar to the hard tanh nonlinearity $\texttt{HardTanh}(v) = \min\{1, \max\{-1, v\}\}$, a piecewise linear approximation of the popular tanh activation function. The formulations and cutting planes presented below for the clipped ReLU can be adapted in a straightforward way to the hard tanh nonlinearity, so we omit explicit statements for brevity.

Proposition 16 *Take some affine function* $f(x) = w \cdot x + b$ *over input domain* $D = [L, U]$. *The following is a valid formulation for* $\text{gr}(\texttt{Clipped} \circ f; [L, U])$:

$$Cz_3 \leqslant y \leqslant C(z_2 + z_3) \tag{33a}$$

$$y \leqslant f(x) - M^-(f)z_1 \tag{33b}$$

$$y \geqslant f(x) + \big(C - M^+(f)\big)\, z_3 \tag{33c}$$

$$1 = z_1 + z_2 + z_3 \tag{33d}$$

$$(x, y, z) \in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1]^3 \tag{33e}$$

$$z \in \{0, 1\}^3. \tag{33f}$$

Moreover, the following inequalities are valid for (33):

$$y \leqslant \sum_{i \in I} w_i x_i + \left(b + \sum_{i \notin I} w_i \breve{U}_i\right)(z_2 + z_3) - \left(\sum_{i \in I} w_i \breve{L}_i\right) z_1 \qquad \forall I \subseteq [\![\eta]\!] \tag{34}$$

$$y \geqslant \sum_{i \in I} w_i x_i + \left(b + \sum_{i \notin I} w_i \breve{L}_i\right)(z_1 + z_2) - \left(\sum_{i \in I} w_i \breve{U}_i - C\right) z_3 \quad \forall I \subseteq [\![\eta]\!]. \tag{35}$$

Additionally, the most violated inequality from the either (34) *or* (35) *can be separated in* $\mathcal{O}(\eta)$ *time.*

Finally, take $\phi(I) \overset{\text{def}}{=} \sum_{i \in I} w_i \breve{L}_i + \sum_{i \notin I} w_i \breve{U}_i + b$*. Then* (34) *is facet-defining if and only if* $\phi(I) < C$*, and* (35) *is facet-defining if and only if* $\phi([\![\eta]\!] \backslash I) > 0$*.*

Proof See Appendix D. □

We note in passing that under the facet-defining condition, the inequalities (34) and (35) are of the form (10a) and (10b), respectively, for an appropriate choice of dual variables.

6 Computational experiments

To conclude this work, we perform a preliminary computational study of our approaches for ReLU-based networks. We focus on verifying image classification networks trained on the canonical MNIST digit data set [47]. We train a neural network $f : [0,1]^{28\times 28} \to \mathbb{R}^{10}$, where each of the 10 outputs corresponds to the logits for each of the digits from 0 to 9. Given a training image $\tilde{x} \in [0,1]^{28\times 28}$, our goal is to prove that there does not exist a perturbation of $\tilde{x}$ such that the neural network f produces a wildly different classification result. If $f(\tilde{x})_i = \max_{j=1}^{10} f(\tilde{x})_j$, then image $\tilde{x}$ is placed in class i. Consider an input image with known label i. To evaluate robustness around $\tilde{x}$ with respect to class j, we can solve the following optimization problem for some small constant $\epsilon > 0$:

$$\max_{a : ||a||_\infty \leqslant \epsilon} f(\tilde{x} + a)_j - f(\tilde{x} + a)_i.$$

If the optimal solution (or a valid dual bound thereof) is less than zero, this verifies that our network is robust around $\tilde{x}$ in the sense that we cannot produce a small perturbation that will flip the classification from i to j.

We train two models, each using the same architecture with two convolutional layers with ReLU activation functions, feeding into a dense layer of ReLU neurons, and then a final dense linear layer. TensorFlow pseudocode specifying the two network architectures is included in Figure 4. We generate 100 instances for each network by randomly selecting images $\tilde{x}$ with true label i from the test data, along with a random target adversarial class $j \neq i$. Note that we make no attempts to utilize recent techniques that train the networks to be verifiable [25,71,72,74].

For all experiments, we use the Gurobi v7.5.2 solver, running with a single thread on a machine with 128 GB of RAM and 32 CPUs at 2.30 GHz. We use a time limit of 30 minutes (1800 s) for each run. We perform our experiments using the `tf.opt` package for optimization over trained neural networks; `tf.opt` is under active development at Google, with the intention to open source the project in the future. Below, the *big-*M *+* (29a) method is the big-M formulation (5) paired with separation[6] over the exponential family (29a), and with

[6] We use cut callbacks in Gurobi to inject separated inequalities into the cut loop. While this offers little control over when the separation procedure is run, it allows us to take advantage of Gurobi's sophisticated cut management implementation.

```
input = placeholder(float32, shape=(28,28))
conv1 = conv2d(input, filters=16, kernel_size=4,
               strides=(2,2), activation=relu, use_bias=True)
conv2 = conv2d(input, filters=32, kernel_size=4,
               strides=(2,2), activation=relu, use_bias=True)
flatten = reshape(conv2, [5*5*32])
dense = dense(flatten, 100, activation=relu, use_bias=True)
logits = dense(dense, 10, use_bias=True)
```

Fig. 4: TensorFlow pseudocode specifying the network architecture used.

method	time (s)	optimality gap	win
big-M + (29a)	174.49	0.53%	81
big-M	1233.49	6.03%	0
big-M + no cuts	1800.00	125.6%	0
extended	890.21	1.26%	6

Table 1: Results for network with standard training. Shifted geometric mean for time and optimality gap taken over 100 instances (shift of 10 and 1, respectively). The "win" column is the number of (solved) instances on which the method is the fastest.

Gurobi's cutting plane generation turned off. Similarly, the *big-M* and the *extended* methods are the big-M formulation (5) and the extended formulation (7) respectively, with default Gurobi settings. Finally, the *big-M + no cuts* method turns off Gurobi's cutting plane generation without adding separation over (29a).

6.1 Network with standard training

We start with a model trained with a standard procedure, using the Adam algorithm [43], running for 15 epochs with a learning rate of 10^{-3}. The model attains 97.2% test accuracy. We select a perturbation ball radius of $\epsilon = 0.1$. We report the results in Table 1 and in Figure 5. The big-M + (29a) method solves 7 times faster on average than the big-M formulation. Indeed, for 79 out of 100 instances the big-M method does not prove optimality after 30 minutes, and it is never the fastest choice (the "win" column). Moreover, the big-M + no cuts times out on every instance, implying that using *some* cuts is important. The extended method is roughly 5 times slower than the big-M + (29a) method, but only exceeds the time limit on 19 instances, and so is substantially more reliable than the big-M method for a network of this size.

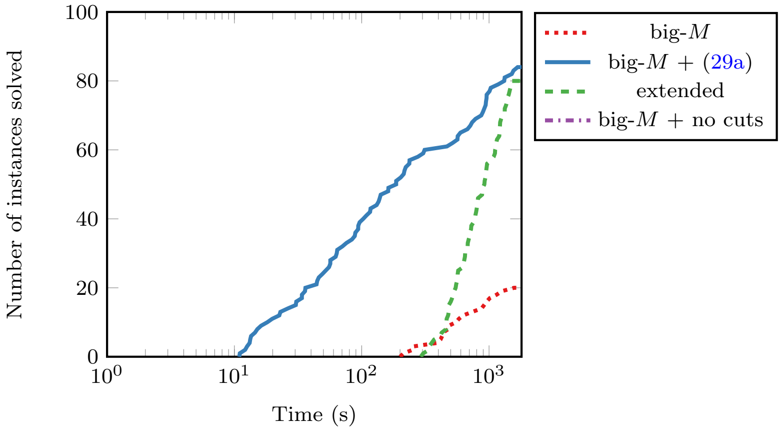


Fig. 5: Number of small network instances solved within a given amount of time. Curves to the upper left are better, with more instances solved in less time.

6.2 ReLU network with L1 regularization

The optimization problems studied in Section 6.1 are surprisingly difficult given the relatively small size of the networks involved. This fact can largely be attributed to the fact that the weights describing the neural network are almost completely dense. In an attempt to remedy this, we train a second model, using the same network architecture from Figure 4, but with L1 regularization added to the training loss as suggested by Xiao et al. [74]. We again set a radius of $\epsilon = 0.1$, and use Adam for training, running for 100 epochs with a learning rate of $5 \cdot 10^{-4}$ and a regularization parameter of 10^{-4}.

We report the corresponding results in Table 2 and Figure 6. While the extended approach does not seem to be substantially affected by the network sparsity, the big-M-based approaches are able to exploit it to solve more instances, more quickly. The big-M approach is able to solve 70 of 100 instances to optimality within the time limit, though the mean solve time is still quite large. In contrast, The big-M + (29a) approach is able to fully exploit the sparsity in the resulting model, solving each instance in the test bed strictly faster than each of the other approaches. Indeed, the approach is able to solve 69 instances in less than 10 seconds, and solves all instances in under 120 seconds.

References

1. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology **33**, 831–838 (2015)

method	time (s)	optimality gap	win
big-M + (29a)	13.65	0%	100
big-M	994.20	14.87%	0
big-M + no cuts	1730.78	80.87%	0
extended	1252.86	37.45%	0

Table 2: Results for network trained with L1 regularization. Shifted geometric mean for time and optimality gap taken over 100 instances (shift of 10 and 1, respectively). The “win” column is the number of (solved) instances on which the method is the fastest.

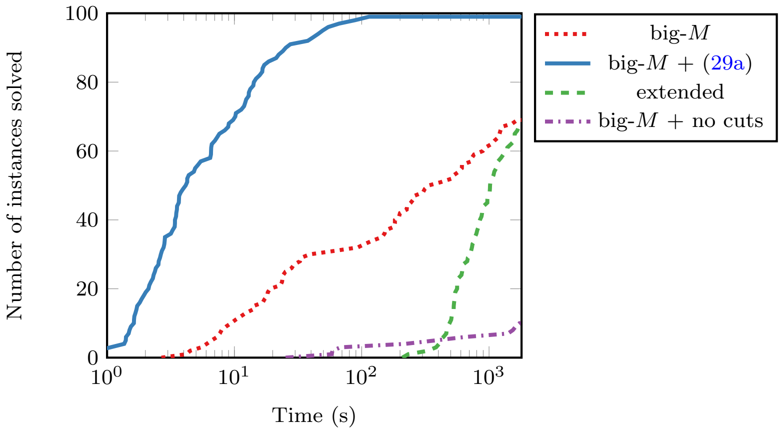


Fig. 6: Number of L1-regularized instances solved within a given amount of time. Curves to the upper left are better, with more instances solved in less time.

2. Amos, B., Xu, L., Kolter, J.Z.: Input convex neural networks. In: D. Precup, Y.W. Teh (eds.) Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 146–155. PMLR, International Convention Centre, Sydney, Australia (2017)
3. Anderson, R., Huchette, J., Tjandraatmadja, C., Vielma, J.P.: Strong mixed-integer programming formulations for trained neural networks. In: A. Lodi, V. Nagarajan (eds.) Proceedings of the 20th Conference on Integer Programming and Combinatorial Optimization, pp. 27–42. Springer International Publishing, Cham (2019). https://arxiv.org/abs/1811.08359
4. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine **34**(6), 26–38 (2017)
5. Atamtürk, A., Gómez, A.: Strong formulations for quadratic optimization with M-matrices and indicator variables. Mathematical Programming (2018)
6. Balas, E.: Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. SIAM Journal on Algorithmic Discrete Methods **6**(3), 466–486 (1985)
7. Balas, E.: Disjunctive programming: Properties of the convex hull of feasible points. Discrete Applied Mathematics **89**, 3–44 (1998)
8. Bartolini, A., Lombardi, M., Milano, M., Benini, L.: Neuron constraints to model complex real-world problems. In: International Conference on the Principles and Practice

of Constraint Programming, pp. 115–129. Springer, Berlin, Heidelberg (2011)
9. Bartolini, A., Lombardi, M., Milano, M., Benini, L.: Optimization and controlled systems: A case study on thermal aware workload dispatching. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, pp. 427–433 (2012)
10. Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A.V., Criminisi, A.: Measuring neural net robustness with constraints. In: Advances in Neural Information Processing Systems, pp. 2613–2621 (2016)
11. Belotti, P., Bonami, P., Fischetti, M., Lodi, A., Monaci, M., Nogales-Gomez, A., Salvagnin, D.: On handling indicator constraints in mixed integer programming. Computational Optimization and Applications **65**(3), 545–566 (2016)
12. Bertsimas, D., Kallus, N.: From predictive to prescriptive analytics. Management Science (2018). https://arxiv.org/abs/1402.5481
13. Bertsimas, D., Tsitsiklis, J.: Introduction to Linear Optimization. Athena Scientific (1997)
14. Biggs, M., Hariss, R.: Optimizing objective functions determined from random forests (2017). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2986630
15. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
16. Bonami, P., Lodi, A., Tramontani, A., Wiese, S.: On mathematical programming with indicator constraints. Mathematical Programming **151**(1), 191–223 (2015)
17. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2559–2566 (2010)
18. Bunel, R., Turkaslan, I., Torr, P.H., Kohli, P., Kumar, M.P.: A unified view of piecewise linear neural network verification. In: Advances in Neural Information Processing Systems (2018)
19. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57 (2017)
20. Cheng, C.H., Nührenberg, G., Ruess, N.: Maximum resilience of artifical neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Springer, Cham (2017)
21. Deng, Y., Liu, J., Sen, S.: Coalescing data and decision sciences for analytics. In: Recent Advances in Optimization and Modeling of Contemporary Problems. INFORMS (2018)
22. Donti, P., Amos, B., Kolter, J.Z.: Task-based end-to-end model learning in stochastic optimization. In: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems 30, pp. 5484–5494. Curran Associates, Inc. (2017)
23. Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., Coppin, B.: Deep reinforcement learning in large discrete action spaces (2015). https://arxiv.org/abs/1512.07679
24. Dutta, S., Jha, S., Sanakaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: NASA Formal Methods Symposium (2018)
25. Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O'Donoghue, B., Uesato, J., Kohli, P.: Training verified learners with learned verifiers (2018). https://arxiv.org/abs/1805.10265
26. Dvijotham, K., Stanforth, R., Gowal, S., Mann, T., Kohli, P.: A dual approach to scalable verification of deep networks. In: Thirty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (2018)
27. Ehlers, R.: Formal verification of piece-wise linear feed-forward neural networks. In: International Symposium on Automated Technology for Verification and Analysis. Springer, Cham (2017)
28. Elmachtoub, A.N., Grigas, P.: Smart "Predict, then Optimize" (2017). https://arxiv.org/abs/1710.08005
29. Fischetti, M., Jo, J.: Deep neural networks and mixed integer linear optimization. Constraints (2018)
30. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style (2015). https://arxiv.org/abs/1508.06576
31. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, vol. 1. MIT Press Cambridge (2016)

32. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: Proceedings of the 30th International Conference on Machine Learning, vol. 28, pp. 1319–1327 (2013)
33. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A.Y.: Deep Speech: Scaling up end-to-end speech recognition (2014). https://arxiv.org/abs/1412.5567
34. den Hertog, D., Postek, K.: Bridging the gap between predictive and prescriptive analytics – new optimization methodology needed (2016). http://www.optimization-online.org/DB_HTML/2016/12/5779.html
35. Hijazi, H., Bonami, P., Cornuéjols, G., Ouorou, A.: Mixed-integer nonlinear programs featuring "on/off" constraints. Computational Optimization and Applications **52**(2), 537–558 (2012)
36. Hijazi, H., Bonami, P., Ouorou, A.: A note on linear on/off constraints (2014). http://www.optimization-online.org/DB_FILE/2014/04/4309.pdf
37. Huber, B., Rambau, J., Santos, F.: The Cayley Trick, lifting subdivisions and the Bohne-Dress theorem of zonotopal tiltings. Journal of the European Mathematical Society **2**(2), 179–198 (2000)
38. Huchette, J.: Advanced mixed-integer programming formulations: Methodology, computation, and application. Ph.D. thesis, Massachusetts Institute of Technology (2018)
39. Jeroslow, R., Lowe, J.: Modelling with integer variables. Mathematical Programming Study **22**, 167–184 (1984)
40. Jeroslow, R.G.: Alternative formulations of mixed integer programs. Annals of Operations Research **12**, 241–276 (1988)
41. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: International Conference on Computer Aided Verification, pp. 97–117 (2017)
42. Khalil, E.B., Gupta, A., Dilkina, B.: Combinatorial attacks on binarized neural networks. In: International Conference on Learning Representations (2019)
43. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). https://arxiv.org/abs/1412.6980
44. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. Springer (2000)
45. Kumar, A., Serra, T., Ramalingam, S.: Equivalent and approximate transformations of deep neural networks (2019). https://arxiv.org/abs/1905.11428
46. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
47. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86, pp. 2278–2324 (1998)
48. Lombardi, M., Gualandi, S.: A lagrangian propagator for artificial neural networks in constraint programming. Constraints **21**(4), 435–462 (2016)
49. Lomuscio, A., Maganti, L.: An approach to reachability analysis for feed-forward ReLU neural networks (2017). https://arxiv.org/abs/1706.07351
50. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML Workshop on Deep Learning for Audio, Speech and Language (2013)
51. Mišić, V.V.: Optimization of tree ensembles (2017). https://arxiv.org/abs/1705.10883
52. Mladenov, M., Boutilier, C., Schuurmans, D., Elidan, G., Meshi, O., Lu, T.: Approximate linear programming for logistic Markov decision processes. In: Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI-17), pp. 2486–2493. Melbourne, Australia (2017)
53. Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going deeper into neural networks (2015). https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html
54. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017). https://distill.pub/2017/feature-visualization
55. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy, pp. 372–387 (2016)

56. Raghunathan, A., Steinhardt, J., Liang, P.: Semidefinite relaxations for certifying robustness to adversarial examples. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, pp. 10,900–10,910. Curran Associates Inc. (2018)
57. Salman, H., Yang, G., Zhang, H., Hsieh, C.J., Zhang, P.: A convex relaxation barrier to tight robustness verification of neural networks (2019). https://arxiv.org/abs/1902.08722
58. Say, B., Wu, G., Zhou, Y.Q., Sanner, S.: Nonlinear hybrid planning with deep net learned transition models and mixed-integer linear programming. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 750–756 (2017)
59. Schweidtmann, A.M., Mitsos, A.: Global deterministic optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications (2018)
60. Serra, T., Ramalingam, S.: Empirical bounds on linear regions of deep rectifier networks (2018). https://arxiv.org/abs/1810.03370
61. Serra, T., Tjandraatmadja, C., Ramalingam, S.: Bounding and counting linear regions of deep neural networks. In: Thirty-fifth International Conference on Machine Learning (2018)
62. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
63. Tawarmalani, M., Sahinidis, N.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software and Applications, vol. 65. Springer Science & Business Media (2002)
64. Tjeng, V., Xiao, K., Tedrake, R.: Verifying neural networks with mixed integer programming. In: International Conference on Learning Representations (2019)
65. Trespalacios, F., Grossmann, I.E.: Improved big-M reformulation for generalized disjunctive programs. Computers and Chemical Engineering **76**, 98–103 (2015)
66. Vielma, J.P.: Mixed integer linear programming formulation techniques. SIAM Review **57**(1), 3–57 (2015)
67. Vielma, J.P.: Embedding formulations and complexity for unions of polyhedra. Management Science **64**(10), 4471–4965 (2018)
68. Vielma, J.P.: Small and strong formulations for unions of convex sets from the Cayley embedding. Mathematical Programming (2018)
69. Vielma, J.P., Nemhauser, G.: Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Mathematical Programming **128**(1-2), 49–72 (2011)
70. Weibel, C.: Minkowski sums of polytopes: Combinatorics and computation. Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2007)
71. Wong, E., Kolter, J.Z.: Provable defenses against adversarial examples via the convex outer adversarial polytope. In: International Conference on Machine Learning (2018)
72. Wong, E., Schmidt, F., Metzen, J.H., Kolter, J.Z.: Scaling provable adversarial defenses. In: 32nd Conference on Neural Information Processing Systems (2018)
73. Wu, G., Say, B., Sanner, S.: Scalable planning with Tensorflow for hybrid nonlinear domains. In: Advances in Neural Information Processing Systems, pp. 6276–6286 (2017)
74. Xiao, K.Y., Tjeng, V., Shafiullah, N.M., Madry, A.: Training for faster adversarial robustness verification via inducing ReLU stability. In: International Conference on Learning Representations (2019)
75. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolution network (2015). https://arxiv.org/abs/1505.00853
76. Zeng, H., Edwards, M.D., Liu, G., Gifford, D.K.: Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics **32**(12), 121–127 (2016)

A Proof of Lemma 1

Lemma 1 *If $d = 2$, then at any values of (x, z) where $\overline{g}(x, z) \geqslant \underline{g}(x, z)$ (i.e. there exists a y such that $(x, y, z) \in R_{\text{cayley}}$), we have*

$$\underline{g}(x, z) = \max\{w^1 \cdot x + b^1, w^2 \cdot x + b^2\}.$$

Proof For convenience, we work with the change of variables $x^k \leftarrow \frac{\tilde{x}^k}{z_k}$ for each $k \in [\![2]\!]$.

Suppose without loss of generality that $x \in D_{|2}$, and consider any feasible solution (x^1, x^2) to the optimization problem for $\underline{g}(x, z)$, which is feasible by the assumption that $(x, z) \in \text{Proj}_{x,z}(R_{\text{cayley}})$. We will show that $(w^1 \cdot x^1 + b^1)z_1 + (w^2 \cdot x^2 + b^2)z_2 \geqslant w^2 \cdot x + b^2$. We assume that $z_2 > 0$, since otherwise $x = x^1 \in D_{|1} \cap D_{|2}$ and the result is immediate.

Since $x = x^1 z_1 + x^2 z_2$ and $z \in \Delta^2$, the line segment joining x^1 to x^2 contains x. Furthermore, since $x^1 \in D_{|1}$ and $x^2 \in D_{|2}$, this line segment also intersects the hyperplane $\left\{ \hat{x} \in \mathbb{R}^\eta \;\middle|\; w^1 \cdot \hat{x} + b^1 = w^2 \cdot \hat{x} + b^2 \right\}$. Let $\hat{x}^1$ denote this point of intersection, and let $\hat{z}^1 \in \Delta^2$ be such that $\hat{x}^1 = x^1 \hat{z}^1_1 + x^2 \hat{z}^1_2$. Since $x \in D_{|2}$, we know that $\hat{x}^1$ is closer to x^1 than x, i.e. $\hat{z}^1_1 \geqslant z_1$. Moreover, take the point $\hat{x}^2$ on this line segment such that

$$x = \hat{x}^1 z_1 + \hat{x}^2 z_2, \tag{36}$$

where $\hat{x}^2 = x^1 \hat{z}^2_1 + x^2 \hat{z}^2_2$ for some $\hat{z}^2 \in \Delta^2$. We have $\hat{z}^2_1 \leqslant z_1$ since $\hat{x}^2$ is further away from x^1 than x. Note that $\hat{x}^1 \in D_{|1} \cap D_{|2}$ while $\hat{x}^2 \in D_{|2}$, and thus $(\hat{x}^1, \hat{x}^2)$ is feasible.

It can be computed that $\hat{z}^2_1 = z_1 \cdot \frac{\hat{z}^1_2}{z_2}$, which implies that $z_1 = z_1(\hat{z}^1_1 + \hat{z}^1_2) = z_1 \hat{z}^1_1 + z_2 \hat{z}^2_1$ and $z_2 = z_2(\hat{z}^2_1 + \hat{z}^2_2) = z_1 \hat{z}^1_2 + z_2 \hat{z}^2_2$. Using these two identities, we obtain

$$\begin{aligned}
&(w^1 \cdot x^1 + b^1)z_1 + (w^2 \cdot x^2 + b^2)z_2 \\
&= (w^1 \cdot x^1 + b^1)(z_1 \hat{z}^1_1 + z_2 \hat{z}^2_1) + (w^2 \cdot x^2 + b^2)(z_1 \hat{z}^1_2 + z_2 \hat{z}^2_2) \\
&= ((w^1 \cdot x^1 + b^1)\hat{z}^1_1 + (w^2 \cdot x^2 + b^2)\hat{z}^1_2)z_1 \\
&\quad + ((w^1 \cdot x^1 + b^1)\hat{z}^2_1 + (w^2 \cdot x^2 + b^2)\hat{z}^2_2)z_2 \\
&= (f(x^1)\hat{z}^1_1 + f(x^2)\hat{z}^1_2)z_1 + (f(x^1)\hat{z}^2_1 + f(x^2)\hat{z}^2_2)z_2,
\end{aligned}$$

where we let $f(\hat{x})$ denote the function $\max\{w^1 \cdot \hat{x} + b^1, w^2 \cdot \hat{x} + b^2\}$, recalling that $x^1 \in D_{|1}$ and $x^2 \in D_{|2}$. Since $f(\hat{x})$ is convex, by Jensen's inequality the preceding expression is at least

$$(f(x^1 \hat{z}^1_1 + x^2 \hat{z}^1_2)))z_1 + (f(x^1 \hat{z}^2_1 + x^2 \hat{z}^2_2))z_2.$$

The preceding expression equals $(w^2 \cdot \hat{x}^1 + b^2)z_1 + (w^2 \cdot \hat{x}^2 + b^2)z_2$ by the definitions of $\hat{x}^1$ and $\hat{x}^2$, and the fact that they both lie in $D_{|2}$. Invoking (36) completes the proof. □

B Proof of Proposition 9

Proposition 9 *Consider the polyhedron P defined as the intersection of all halfspaces corresponding to the inequalities (20). Consider some arbitrary mapping $J : [\![\tau]\!] \to \mathbb{Z}$ with $J(i) \in [\![p_i]\!]$ for each $i \in [\![\tau]\!]$. Then the inequality in (20) for J is irredundant with respect to P. That is, removing the halfspace corresponding to mapping J (note that this halfspace could also correspond to other mappings) from the description of P will strictly enlarge the feasible set.*

Proof Fix a mapping J. Consider the feasible points with $z = 1/2$, and $x_{i,j} = \mathbb{1}[j = J(i)]$ for each i and j. At such points, the constraint corresponding to any mapping $J' \neq J$ in (20) is

$$\begin{aligned} y &\leqslant \sum_{i=1}^{\tau} \left(\frac{\widetilde{w}_{i,J'(i)}}{2} + \sum_{j=J'(i)+1}^{p_i} (\widetilde{w}_{i,j} - \widetilde{w}_{i,J'(i)})\mathbb{1}[j = J(i)] + \sum_{j=1}^{p_i} w^2_{i,j}\mathbb{1}[j = J(i)] \right) + \frac{b^1 + b^2}{2} \\ &= \sum_{i=1}^{\tau} \left(\frac{\widetilde{w}_{i,J'(i)}}{2} + (\widetilde{w}_{i,J(i)} - \widetilde{w}_{i,J'(i)})\mathbb{1}[J'(i) < J(i)] \right) + \sum_{i=1}^{\tau} w^2_{i,J(i)} + \frac{b^1 + b^2}{2}. \end{aligned}$$

For any simplex i, recall that the indices are sorted so that $\widetilde{w}_{i1} \leqslant \cdots \leqslant \widetilde{w}_{ip_i}$. Thus, if $J'(i) \geqslant J(i)$, then the expression inside the outer parentheses equals $\frac{\widetilde{w}_{i,J'(i)}}{2} \geqslant \frac{\widetilde{w}_{i,J(i)}}{2}$. On the other hand, if $J'(i) < J(i)$, then the expression inside the outer parentheses can be re-written as $\frac{\widetilde{w}_{i,J(i)}}{2} + \frac{\widetilde{w}_{i,J(i)} - \widetilde{w}_{i,J'(i)}}{2} \geqslant \frac{\widetilde{w}_{i,J(i)}}{2}$. Therefore, setting $J'(i) = J(i)$ for every simplex i achieves the tightest upper bound in (20), which simplifies to

$$y \leqslant \sum_{i=1}^{\tau} \frac{\widetilde{w}_{i,J(i)}}{2} + \sum_{i=1}^{\tau} w^2_{i,J(i)} + \frac{b^1 + b^2}{2}. \tag{37}$$

Now, suppose that the same upper bound on y is achieved by a mapping J' such that $J'(i) \neq J(i)$ on a simplex i. By the argument above, regardless of whether $J'(i) > J(i)$ or $J'(i) < J(i)$, the expression inside the outer parentheses can equal $\frac{\widetilde{w}_{i,J(i)}}{2}$ only if $\widetilde{w}_{i,J'(i)} = \widetilde{w}_{i,J(i)}$. In this case, inspecting the term inside the summation in (20) for mappings J and J', we observe that regardless of the values of x or z,

$$\widetilde{w}_{i,J'(i)} z + \sum_{j=J'(i)+1}^{p_i} (\widetilde{w}_{i,j} - \widetilde{w}_{i,J'(i)}) x_{i,j} = \widetilde{w}_{i,J(i)} z + \sum_{j=J(i)+1}^{p_i} (\widetilde{w}_{i,j} - \widetilde{w}_{i,J(i)}) x_{i,j}.$$

Therefore, for a mapping J' to achieve the tightest upper bound (37), it must be the case that $\widetilde{w}_{i,J'(i)} = \widetilde{w}_{i,J(i)}$ on every simplex i, which means that J' and J correspond to the same half-space in $D \times \mathbb{R} \times [0, 1]$. This completes the proof. □

C Proof of Proposition 11

Proposition 11 *A sharp formulation for* $\mathrm{gr}(\mathtt{Max} \circ (f^1, \ldots, f^d); [L, U])$ *is*

$$y \leqslant \sum_{i=1}^{\eta} \left(w_i^{I(i)} x_i + \sum_{k=1}^{d} \max\{(w_i^k - w_i^{I(i)})L_i, (w_i^k - w_i^{I(i)})U_i\} z_k \right) + \sum_{k=1}^{d} b^k z_k$$

$$\forall \textit{ mappings } I : [\![\eta]\!] \to [\![d]\!] \tag{27a}$$

$$y \geqslant w^k \cdot x + b^k \quad \forall k \in [\![d]\!] \tag{27b}$$

$$(x, y, z) \in [L, U] \times \mathbb{R} \times \Delta^d. \tag{27c}$$

$$z \in \{0, 1\}^d. \tag{27d}$$

Furthermore, none of the constraints in the families (27a) *and* (27b) *are redundant.*

Proof Our goal is to transform the box domain $[L, U]$ to the domain which is a product of η dimension-2 simplices, so that we can apply our results from Section 4 (specifically, Corollary 3). Consider the domain $D = (\Delta^2)^\eta$ where the x-coordinates are given by $x_{i,1}, x_{i,2} \geqslant 0$ over $i \in [\![\eta]\!]$, with $x_{i,1} + x_{i,2} = 1$ for each simplex i. For each i, let $w^k_{i,1} = w^k_i U_i$, $w^k_{i,2} = w^k_i L_i$, and let $\sigma_i : [\![d]\!] \to [\![d]\!]$ be a permutation such that $w^{\sigma_i(1)}_{i,1} - w^{\sigma_i(1)}_{i,2} \leqslant \cdots \leqslant w^{\sigma_i(d)}_{i,1} - w^{\sigma_i(d)}_{i,2}$.

The set of constraints (22) from Corollary 3 can then be expressed as

$$
\begin{aligned}
y \leqslant \sum_{i=1}^{\tau} \Bigg(&(w_{i,1}^{\sigma_i(I(i))} - w_{i,2}^{\sigma_i(I(i))}) x_{i,1} + \sum_{k=1}^{I(i)} w_{i,2}^{\sigma_i(k)} z_k \\
&+ \sum_{k=I(i)+1}^{d} (w_{i,1}^{\sigma_i(k)} - (w_{i,1}^{\sigma_i(I(i))} - w_{i,2}^{\sigma_i(I(i))})) z_k \Bigg) + \sum_{k=1}^{d} b^k z_k,
\end{aligned}
$$

over the mappings $I : [\![\eta]\!] \to [\![d]\!]$. Substituting in the definitions of $w_{i,1}^k, w_{i,2}^k$ and letting ξ_i denote $(U_i - L_i)x_{i,1} + L_i$ for brevity, we can rewrite these constraints as:

$$
\begin{aligned}
y \leqslant \sum_{i=1}^{\eta} \Bigg(&w_i^{\sigma_i(I(i))}(U_i - L_i)x_{i,1} + \sum_{k=1}^{I(i)} w_i^{\sigma_i(k)} L_i z_k \\
&+ \sum_{k=I(i)+1}^{d} (w_i^{\sigma_i(k)} U_i - w_i^{\sigma_i(I(i))}(U_i - L_i)) z_k \Bigg) + \sum_{k=1}^{d} b^k z_k \\
= \sum_{i=1}^{\eta} \Bigg(&w_i^{\sigma_i(I(i))} \xi_i + \sum_{k=1}^{I(i)} (w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))}) L_i z_k + \sum_{k=I(i)+1}^{d} (w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))}) U_i z_k \Bigg) + \sum_{k=1}^{d} b^k z_k \\
= \sum_{i=1}^{\eta} \Bigg(&w_i^{\sigma_i(I(i))} \xi_i + \sum_{k=1}^{d} \max\{(w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))}) L_i, (w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))}) U_i\} z_k \Bigg) + \sum_{k=1}^{d} b^k z_k \\
= \sum_{i=1}^{\eta} \Bigg(&w_i^{\sigma_i(I(i))} \xi_i + \sum_{k=1}^{d} \max\{(w_i^{k} - w_i^{\sigma_i(I(i))}) L_i, (w_i^{k} - w_i^{\sigma_i(I(i))}) U_i\} z_k \Bigg) + \sum_{k=1}^{d} b^k z_k,
\end{aligned}
\tag{39}
$$

where the first equality holds because $\xi_i - (U_i - L_i)x_{i,1} = L_i$ for each i and $\sum_k z_k = 1$, the second equality holds because $(w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))})L_i \geqslant (w_i^{\sigma_i(k)} - w_i^{\sigma_i(I(i))})U_i$ if $k \leqslant I(i)$ (and a reverse argument can be made if $k > I(i)$), and the third equality holds because each $\sigma_i : [\![d]\!] \to [\![d]\!]$ is a bijection.

Now, since we are taking the constraints (39) over all mappings $I : [\![\eta]\!] \to [\![d]\!]$, it is equivalent to replace $\sigma_i(I(i))$ with $I(i)$ in (39). Therefore, Corollary 3 says that the following is a sharp formulation for the function $\max_{k=1}^{d}\{\sum_{i=1}^{\eta}(w_{i,1}^k x_{i,1} + w_{i,2}^k x_{i,2}) + b^k\}$ over $(\Delta^2)^\eta$:

$$
\begin{aligned}
y \leqslant \sum_{i=1}^{\eta} \Bigg(w_i^{I(i)} \xi_i + \sum_{k=1}^{d} \max\{(w_i^k - w_i^{I(i)}) L_i, (w_i^k - w_i^{I(i)}) U_i\} z_k \Bigg) + \sum_{k=1}^{d} b^k z_k \quad &\forall I : [\![\eta]\!] \to [\![d]\!] \\
y \geqslant \sum_{i=1}^{\eta} (w_{i,1}^k x_{i,1} + w_{i,2}^k x_{i,2}) + b^k \quad &\forall k \in [\![d]\!] \\
(x, y, z) \in (\Delta^2)^\eta \times \mathbb{R} \times \Delta^d&.
\end{aligned}
$$

Note that $w_{i,1}^k x_{i,1} + w_{i,2}^k x_{i,2}$ can be re-written as $w_i^k(U_i x_{i,1} + L_i(1 - x_{i,1})) = w_i^k \xi_i$, where we have used the identity $x_{i,2} = 1 - x_{i,1}$ and the fact that ξ_i is shorthand for $(U_i - L_i)x_{i,1} + L_i$.

Now, consider the sharp formulation above with the x-variables replaced by the ξ-variables, where each ξ_i is allowed to range over $[L_i, U_i]$ (since each $x_{i,1}$ was allowed to range over $[0, 1]$). This transformation of variables is a bijection, and hence the new formulation over $(\xi, y, z) \in [L, U] \times \mathbb{R} \times \Delta^d$ is also sharp, yielding the desired result.

The irredundancy of (27a) also follows from Corollary 3. For the irredundancy of (27b), fix k and take a point $(\hat{x}, \hat{y}, \hat{z})$ where $f^k(\hat{x}) > f^\ell(\hat{x})$ for all $\ell \neq k$, which exists by Assumption 2. Thus, since the inequalities (27b) are the only ones bounding y from below, the point $(\hat{x}, \hat{y} - \epsilon, \hat{z})$ for some $\epsilon > 0$ is satisfied by all constraints except (27b) corresponding to k, and therefore it is not redundant. □

D Proof of Proposition 16

We separately prove each of the three parts of Proposition 16.

Proposition 16 (Part 1) *Take some affine function $f(x) = w \cdot x + b$ over input domain $D = [L, U]$. The following is a valid formulation for* $\mathrm{gr}(\texttt{Clipped} \circ f; [L, U])$*:*

$$\begin{aligned}
Cz_3 \leqslant y \leqslant C(z_2 + z_3) && \text{(33a)}\\
y \leqslant f(x) - M^-(f)z_1 && \text{(33b)}\\
y \geqslant f(x) + \left(C - M^+(f)\right) z_3 && \text{(33c)}\\
1 = z_1 + z_2 + z_3 && \text{(33d)}\\
(x, y, z) \in [L, U] \times \mathbb{R}_{\geqslant 0} \times [0, 1]^3 && \text{(33e)}\\
z \in \{0, 1\}^3. && \text{(33f)}
\end{aligned}$$

Proof If $z = (1, 0, 0)$, then the (33a-33c,33e) reduces to

$$P^1 = \left\{ (x, y) \in [L, U] \times \mathbb{R}_{\geqslant 0} \;\middle|\; \begin{array}{c} y = 0 \\ f(x) \geqslant M^-(f) \\ y \geqslant f(x) \end{array} \right\}.$$

If $z = (0, 1, 0)$, then the (33a-33c,33e) reduces to

$$P^2 = \left\{ (x, y) \in [L, U] \times \mathbb{R}_{\geqslant 0} \;\middle|\; \begin{array}{c} 0 \leqslant y \leqslant C \\ y \leqslant f(x) \\ y \geqslant f(x) \end{array} \right\}.$$

If $z = (0, 0, 1)$, then the (33a-33c,33e) reduces to

$$P^3 = \left\{ (x, y) \in [L, U] \times \mathbb{R}_{\geqslant 0} \;\middle|\; \begin{array}{c} y = C \\ y \leqslant f(x) \\ f(x) \leqslant M^+(f) \end{array} \right\}.$$

This exhausts all feasible values for z with respect to (33d,33f). Furthermore, $\mathrm{gr}(\texttt{Clipped} \circ f; [L, U]) = P^1 \cup P^2 \cup P^3$, giving the result. □

Proposition 16 (Part 2) *Moreover, the following inequalities are valid for (33):*

$$y \leqslant \sum_{i \in I} w_i x_i + \left(b + \sum_{i \notin I} w_i \breve{U}_i \right) (z_2 + z_3) - \left(\sum_{i \in I} w_i \breve{L}_i \right) z_1 \qquad \forall I \subseteq [\![\eta]\!] \tag{34}$$

$$y \geqslant \sum_{i \in I} w_i x_i + \left(b + \sum_{i \notin I} w_i \breve{L}_i \right) (z_1 + z_2) - \left(\sum_{i \in I} w_i \breve{U}_i - C \right) z_3 \quad \forall I \subseteq [\![\eta]\!]. \tag{35}$$

Additionally, the most violated inequality from the either (34) or (35) can be separated in $\mathcal{O}(\eta)$ time.

Proof Fix some subset $I \subseteq [\![\eta]\!]$.

Validity of (34). Follows by case analysis: when $z = (1, 0, 0)$, the inequality reduces to

$$y \leqslant \sum_{i \in I} w_i (x_i - \breve{L}_i),$$

whose validity follows as $w_i x_i \geqslant w_i \breve{L}_i$ for all i. If $z = (0, 1, 0)$ or $z = (0, 0, 1)$, then $y \leqslant w \cdot x + b$ is a valid inequality, and (34) reduces to

$$y \leqslant \sum_{i \in I} w_i x_i + \sum_{i \notin I} w_i \breve{U}_i + b,$$

whose validity follows from the bounds $w_i x_i \leqslant w_i \breve{U}_i$ for all i.

Validity of (35). Consider the case when $z = (1,0,0)$ or $z = (0,1,0)$, in which case the inequality $y \geqslant w \cdot x + b$ is valid, and the inequality (35) reduces to

$$y \geqslant \sum_{i \in I} w_i x_i + \sum_{i \notin I} w_i \breve{L}_i + b,$$

whose validity then follows from the bounds $w_i x_i \geqslant w_i \breve{L}_i$ for all i. If $z = (0,0,1)$, then $y = C$ is valid, and the inequality (35) reduces to

$$y \geqslant C + \sum_{i \in I} w_i (x_i - \breve{U}_i),$$

whose validity follows from the bounds $w_i x_i \leqslant w_i \breve{U}_i$ for all i.

Finally, observe that separation over both (34) and (35) can be performed by a single pass through the input components $i \in [\![\eta]\!]$, as minimizing the right-hand side of either inequality can be done in a completely separable manner. □

Proposition 16 (Part 3) *Finally, take $\phi(I) \overset{\text{def}}{=} \sum_{i \in I} w_i \breve{L}_i + \sum_{i \notin I} w_i \breve{U}_i + b$. Then (34) is facet-defining if and only if $\phi(I) < C$, and (35) is facet-defining if and only if $\phi([\![\eta]\!] \backslash I) > 0$.*

Proof "Only-if" direction. To show the only-if direction for (34), we observe that when $\phi(I) \geqslant C$, we can express (34) as a conic combination of other constraints in the following way:

$$\begin{array}{rcll} y \leqslant C(z_2 + z_3) & \times & 1 & \\ w_i \breve{L}_i \leqslant w_i x_i & \times & 1 & \forall i \in I \\ 0 \leqslant z_2 & \times & \phi(I) - C & \\ 0 \leqslant z_3 & \times & \phi(I) - C, & \end{array}$$

and then simplifying using the equation $z_1 + z_2 + z_3 = 1$.

To show the only-if direction for (35), we observe that when $\phi([\![\eta]\!] \backslash I) \leqslant 0$, we can express (35) as a conic combination of the other constraints in the following way:

$$\begin{array}{rcll} y \geqslant C z_3 & \times & 1 & \\ w_i \breve{U}_i \geqslant w_i x_i & \times & 1 & \forall i \in I \\ z_1 \geqslant 0 & \times & -\phi([\![\eta]\!] \backslash I) & \\ z_2 \geqslant 0 & \times & -\phi([\![\eta]\!] \backslash I), & \end{array}$$

and then simplifying using the equation $z_1 + z_2 + z_3 = 1$.

"If" direction. To show that each inequality in the family (34) with $\phi(I) < C$ (resp. (35) with $\phi([\![\eta]\!] \backslash I) > 0$) is facet-defining under the strict activity assumption, presume w.l.o.g. that $w \geqslant 0$; if this is not the case, in the argument below replace L and U with $\breve{L}$ and $\breve{U}$, respectively, and take care to either add or subtract ϵ perturbations appropriately to maintain feasibility. Take the points $p^1 = (x, y, z) = (L, 0, \mathbf{e}^1)$ and $p^3 = (x, y, z) = (U, C, \mathbf{e}^3)$, which are both feasible for (33) under our assumptions, and satisfy any inequality (34) or (35) at equality. Take some sufficiently small $\epsilon > 0$, and define the points $p^{1,i} = (L + \epsilon \mathbf{e}^i, 0, \mathbf{e}^1)$ for each $i \notin I$. By strict activity, such an ϵ exists such that each point is feasible for (33), and satisfies the inequalities in (34) and (35) corresponding to I at equality.

Take some $\mathring{x} \in [L, U]$ where: $0 < w \cdot \mathring{x} + b < C$, $\mathring{x}_i = U_i$ (resp. $\mathring{x}_i = L_i$) for each $i \notin I$, and $L_i < \mathring{x}_i < U_i$ for each $i \in I$. For example, in the case of (34), the point $\mathring{x}$ exists as it can be found as a convex combination between the point U (where $f(U) > C$ by strict activity and $w \geqslant 0$) and the point $\tilde{x}$ given by $\tilde{x}_i = L_i$ for $i \in I$ and $\tilde{x}_i = U_i$ for $i \notin I$, for which $f(\tilde{x}) = \phi(I) < C$. The analogous is true for (35) with $\phi([\![\eta]\!] \backslash I) > 0$. Take the points

$p^2 = (\mathring{x}, f(\mathring{x}), \mathbf{e}^2)$ and $p^{2,k} = (\mathring{x} + \delta \mathbf{e}^k, f(\mathring{x}) + w_k \delta, \mathbf{e}^2)$ for each $k \in I$. For sufficiently small δ, each point is feasible for (33), and satisfies the inequalities in (34) and (35) corresponding to I at equality.

To finish, we must show that the $\eta + 3$ points constructed thus far are affinely independent. Presume w.l.o.g. that $I = \llbracket \kappa \rrbracket$ for some $\kappa \in \llbracket \eta \rrbracket$.

$$\begin{pmatrix} p^2 - p^1 \\ p^3 - p^1 \\ p^{2,1} - p^1 \\ \vdots \\ p^{2,\kappa} - p^1 \\ p^{1,\kappa+1} - p^1 \\ \vdots \\ p^{1,\eta} - p^1 \end{pmatrix} = \begin{pmatrix} \mathring{x} - L & f(\mathring{x}) - f(L) & \mathbf{e}^2 - \mathbf{e}^1 \\ \mathring{x} - U & f(\mathring{x}) - f(U) & \mathbf{e}^2 - \mathbf{e}^3 \\ \mathring{x} - L + \delta \mathbf{e}^1 & f(\mathring{x}) - f(L) + w_1 \delta & \mathbf{e}^2 - \mathbf{e}^1 \\ \vdots & \vdots & \vdots \\ \mathring{x} - L + \delta \mathbf{e}^\kappa & f(\mathring{x}) - f(L) + w_\kappa \delta & \mathbf{e}^2 - \mathbf{e}^1 \\ \epsilon \mathbf{e}^{\kappa+1} & w_{\kappa+1} \epsilon & 0 \\ \vdots & \vdots & \vdots \\ \epsilon \mathbf{e}^\eta & w_\eta \epsilon & 0 \end{pmatrix}.$$

Now subtract the $p^2 - p^1$ row from rows $p^{2,k} - p^1$ to yield

$$\begin{pmatrix} \mathring{x} - L & f(\mathring{x}) - f(L) & \mathbf{e}^2 - \mathbf{e}^1 \\ \mathring{x} - U & f(\mathring{x}) - f(U) & \mathbf{e}^2 - \mathbf{e}^3 \\ \delta \mathbf{e}^1 & w_1 \delta & 0 \\ \vdots & \vdots & \vdots \\ \delta \mathbf{e}^\kappa & w_\kappa \delta & 0 \\ \epsilon \mathbf{e}^{\kappa+1} & w_{\kappa+1} \epsilon & 0 \\ \vdots & \vdots & \vdots \\ \epsilon \mathbf{e}^\eta & w_\eta \epsilon & 0 \end{pmatrix}.$$

If we permute the last three columns (corresponding to the z variables) to the first three columns, we observe that the resulting matrix is upper triangular with a nonzero diagonal, and so has full row rank. Therefore, the starting matrix also has full row rank, as we only applied elementary row operations, and therefore the $\eta + 3$ points are affinely independent, giving the result. □