A Mathematical Background

A.1 Oriented Hyperplane Arrangements

Any $\bar{\mathbf{x}} \in \mathbb{R}^{d+1}$ defines a tripartite division of $\mathbb{R}^{d+1}$ given by

$$H^{\alpha} = \{\mathbf{w} \in \mathbb{R}^{d+1} \mid \operatorname{sign}(\mathbf{w}^T \bar{\mathbf{x}}) = \alpha\}, \tag{4}$$

where $\alpha \in \{-1, 0, +1\}$. The set H^0 is a hyperplane while H^+, H^- are the positive and negative open half-spaces, respectively. We call their closures $\bar{H}^{\pm} = H^0 \cup H^{\pm}$ the positive/negative closed half-spaces. For consistency, we say $\bar{H}^0 = H^0$.

We are provided with a finite set of vectors $\{\bar{\mathbf{x}}_i\}_{i=1}^N \subseteq \mathbb{R}^{d+1}$. We interpret these vectors as the training examples in section 3, but for now consider them to be arbitrary vectors. Let H_i^{α} denote the subsets given by (4) for the i-th example. The union of the hyperplanes $\bigcup_{i=1}^N H_i^0$ separates chambers of $\mathbb{R}^{d+1}$ into a finite number of disjoint cells. Each cell can be described as the intersection of open half-spaces and thus can be indexed by a vector of sign patterns reflecting whether the positive or negative half-space is used. The structure induced by these hyperplanes along with their orientation information creates what is known as an oriented hyperplane arrangement (Richter-Gebert & Ziegler, 2017).

A.2 Polyhedral Complexes

We can also describe this arrangement through the notion of a polyhedral complex (Ziegler, 2012). We first provide some definitions that will be useful. A polyhedral set is any set that is the intersection of a finite number of closed half-spaces. A polytope is a bounded polyhedral set. Any hyperplane intersecting a polyhedral set will either divide it into two polyhedral sets or only intersect it on its boundary. In the latter case, we call such a hyperplane a supporting hyperplane. A face of a polyhedral set is defined as its intersection with a supporting hyperplane. The dimension of a face is the dimension of its affine span. If the dimension of a face is k, we call it a k-face. We call 0-faces vertices and 1-faces edges. By convention, the empty set is a face of any polyhedral set. The 1-skeleton of a polyhedral set is the graph formed by its vertices and edges.

A polyhedral complex $\mathcal{K}$ is a finite set of polyhedral sets satisfying

1. If $P \in \mathcal{K}$ and F is a face of P, then $F \in \mathcal{K}$.
2. If $P_1, P_2 \in \mathcal{K}$, then their intersection $P_1 \cap P_2$ is a face of both P_1, P_2.

We call a codimension 0 member of $\mathcal{K}$ a chamber of the polyhedral complex. The support of a polyhedral complex is the union of its polyhedral sets. If a polyhedral complex's support equals the entire space, then we call it a polyhedral decomposition of the space. The "is a face of" relation induces a poset structure on the members of a polyhedral complex, which we call its face poset. This can be extended further to a meet-semilattice with the meet operation being given by set intersection. We call this the face semilattice.

Going back to the case of an oriented hyperplane arrangement, let $\mathbf{a} \in \{-1, 0, +1\}^N$ be some sign pattern. Now let us write

$$R^{\mathbf{a}} = \bigcap_{i=1}^{N} \bar{H}_i^{a_i}. \tag{5}$$

Since our hyperplanes are all linear (i.e. non-affine), we always have $\mathbf{0} \in R^{\mathbf{a}}$. If $R^{\mathbf{a}} = \{\mathbf{0}\}$, we say that $R^{\mathbf{a}}$ is null. Define $\mathcal{R}$ to be the set of all $R^{\mathbf{a}}$. Then $\mathcal{R}$ is a polyhedral complex, which we prove in appendix B.1.

Every chamber of $\mathcal{R}$ corresponds to a non-null $R^{\mathbf{a}}$ with $\mathbf{a} \in \{-1, +1\}^N$. If the hyperplanes are general positions, this correspondence is one-to-one. The face semilattice of $\mathcal{R}$ provides information about how its chambers are arranged in space. For example, let $M \in \mathcal{R}$ be the meet of two chambers. If $M \neq \{\mathbf{0}\}$, then those chambers are neighbors. Then $\dim M \in \{1, \ldots, d\}$ and the sign patterns of the chambers differ by at least $d + 1 - \dim M$ sign flips, with equality always holding the hyperplanes are in general positions.

A.3 Dual Zonotopes

It turns out that we can describe the incidence structure of the polyhedral complex $\mathcal{R}$ nicely with a single polytope called its dual zonotope $\mathcal{Z}$ (Ziegler, 2012). A zonotope is any polytope that can be expressed as the Minkowski sum of a finite set of line segments called its generators (McMullen, 1971). In the case of the dual zonotope of $\mathcal{R}$, these generators are the line segments $\{[\mathbf{0}, \bar{\mathbf{x}}_i]\}_{i=1}^N$. We can thus write

$$\mathcal{Z} = \left\{ \sum_{i=1}^{N} \lambda_i \bar{\mathbf{x}}_i \mid \lambda_i \in [0, 1] \right\}. \tag{6}$$

When the generators are in general positions, each k-face of $\mathcal{Z}$ is a k-dimension parallelepiped. Nontrivial linear dependencies between generators, however, lead to k-faces that are the union of multiple k-dimension parallelepipeds lying in the same k-dimension affine subspace.

The duality between $\mathcal{Z}$ and $\mathcal{R}$ allows us to associate members of $\mathcal{R}$ with faces of $\mathcal{Z}$. Each k-face of $\mathcal{Z}$ corresponds to a codimension k member of $\mathcal{R}$. Notably, the vertices of $\mathcal{Z}$, denoted by $\operatorname{vert}(\mathcal{Z})$, correspond to the chambers of $\mathcal{R}$. Relationships between members of $\mathcal{R}$ carry over to faces of $\mathcal{Z}$. For example, two neighboring chambers of $\mathcal{R}$ whose sign patterns differ by a single flipped sign will correspond to two vertices connected by an edge in $\mathcal{Z}$.

We now describe how to make this correspondence explicit. Let $\mathbf{v} \in \operatorname{vert}(\mathcal{Z})$ be a vertex. It can be shown that $\mathbf{v}$ has a unique representation as $\sum_{i=1}^N \lambda_i \bar{\mathbf{x}}_i$ with every $\lambda_i \in \{0, 1\}$. We call the vector $(\lambda_1, \ldots, \lambda_N)$ the barcode of the vertex. We will often treat a vertex interchangeably with its barcode in this paper with difference being clear by context. Let $\mathbf{a}$ be the sign pattern of the chamber corresponding to $\mathbf{v}$. Then $a_i = -1$ if $\lambda_i = 0$ and $a_i = +1$ if $\lambda_i = 1$ for $i = 1, \ldots, N$.

Now suppose two vertices $\mathbf{v}_1, \mathbf{v}_2 \in \operatorname{vert}(\mathcal{Z})$ are connected by an edge, and that the hyperplanes of $\mathcal{R}$ are in general positions. Then there exists a single i^* such that, WLOG, $\mathbf{v}_2 = \mathbf{v}_1 + \bar{\mathbf{x}}_{i^*}$. The sign pattern of the member of $\mathcal{R}$ corresponding to the edge can then be found by finding the sign pattern for $\mathbf{v}_1$ and changing its i^*-th entry to be 0.

A.3.1 Cartesian Power of Zonotopes

Let us consider an m-ary Cartesian power of a zonotope $\mathcal{Z}^m = \prod_{i=1}^m \mathcal{Z}$. We can see that $\mathcal{Z}^m$ is also a zonotope and is generated by line-segments from the origin to members of $\bigcup_{i=1}^m \bigcup_{j=1}^N \{\mathbf{e}_i \bar{\mathbf{x}}_j^T\}$, where $\mathbf{e}_i \in \mathbb{R}^m$ is the i-th standard coordinate vector. Each k-face of $\mathcal{Z}^m$ is the Cartesian product of a set of $\{k_1, \ldots, k_m\}$-faces of $\mathcal{Z}$ where $k = k_1 + \cdots + k_m$. Notably, each vertex of $\mathcal{Z}^m$ corresponds to a product of m vertices of $\mathcal{Z}$. Edges of $\mathcal{Z}^m$ correspond to the product of a single edge of $\mathcal{Z}$ with $m - 1$ vertices.

B Proofs for Appendix A

B.1 Proof that $\mathcal{R}$ is a Polyhedral Complex

Recall that a polyhedral complex $\mathcal{K}$ is a finite set of polyhedral sets satisfying

1. If $P \in \mathcal{K}$ and F is a face of P, then $F \in \mathcal{K}$.
2. If $P_1, P_2 \in \mathcal{K}$, then their intersection $P_1 \cap P_2$ is a face of both P_1, P_2.

Recall that we have defined $\mathcal{R}$ as

$$\mathcal{R} = \left\{ R^{\mathbf{a}} \subseteq \mathbb{R}^{d+1} \mid \mathbf{a} \in \{-1, 0, +1\}^N \right\}, \tag{7}$$

where $R^{\mathbf{a}}$ is given by (5). Note that generally $|\mathcal{R}| \leq 3^N$ since multiple $R^{\mathbf{a}}$ can equal $\{\mathbf{0}\}$.

It is easy to see that every $R^{\mathbf{a}} \in \mathcal{R}$ is a polyhedral set since it can be can defined as the intersection of finitely many closed half-spaces. When $a_i = 0$, its corresponding hyperplane in (5) is equivalent to the intersection of its positive and negative closed half-spaces.

We now prove the first condition for $\mathcal{R}$ being a polyhedral complex. Suppose $R^{\mathbf{a}} \in \mathcal{R}$ and suppose F is a face of $R^{\mathbf{a}}$. Hence F is the intersection of $R^{\mathbf{a}}$ with a supporting hyperplane. It is straightforward to see that any face of $R^{\mathbf{a}}$ can be represented by a $R^{\mathbf{b}}$ where $b_i = a_i$ for all $i \in [N] \setminus I$ and $b_i = 0, a_i = \pm 1$ for $i \in I \subseteq [N]$. Hence the face $F = R^{\mathbf{b}} \in \mathcal{R}$.

We now prove the second condition for $\mathcal{R}$ being a polyhedral complex. Let $R^{\mathbf{a}}, R^{\mathbf{b}} \in \mathcal{R}$. From (5), we see that

$$R^{\mathbf{a}} \cap R^{\mathbf{b}} = \bigcap_{i=1}^{N} \bar{H}_i^{a_i} \cap \bar{H}_i^{b_i}. \tag{8}$$

We see that $H_i^{a_i} \cap \bar{H}_i^{b_i} = H_i^{\pm 1}$ if $H_i^{a_i} = H_i^{b_i} = H_i^{\pm 1}$, and that $H_i^{a_i} \cap \bar{H}_i^{b_i} = H_i^0$ otherwise. It is easy to see that this intersection can be represented as the intersection of a supporting hyperplane with either $R^{\mathbf{a}}$ or $R^{\mathbf{b}}$. Hence their intersection is a mutual face.

C Proof of Theorem 4.1

Suppose we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ in general position. Given some $\epsilon > 0$, let $\mathcal{D}_\epsilon = \{(\mathbf{x}_i', y_i)\}_{i=1}^N$ be any perturbation of $\mathcal{D}$ such that $\|\mathbf{x}_i - \mathbf{x}_i'\|_2 \leq \epsilon$.

Let $\bar{X} \in \mathbb{R}^{N \times (d+1)}$ denote the data matrix in homogeneous coordinates for $\mathcal{D}$. Let $\bar{X}'$ denote the corresponding data matrix for $\mathcal{D}_\epsilon$. We see that $\|\bar{X} - \bar{X}'\|_F \leq \sqrt{N}\epsilon$. Hence some matrix $P \in \mathbb{R}^{N \times (d+1)}$ exists such that $\bar{X}' = \bar{X} + P$ and $\|P\|_F \leq \sqrt{N}\epsilon$.

We can interpret the zonotope definition (6) as saying that a zonotope is the image of a hypercube under the matrix formed by its generators. Hence if $\mathcal{Z}$ is the zonotope of the original dataset, we may write

$$\mathcal{Z} = \left\{ \bar{X}^T \mathbf{u} \in \mathbb{R}^{d+1} \mid \mathbf{u} \in [0,1]^N \right\}. \tag{9}$$

Let $\mathcal{Z}'$ denote the corresponding zonotope for the perturbed dataset.

We now wish to show that the vertices of $\mathcal{Z}$ are in a one-to-one correspondence with the vertices $\mathcal{Z}'$ for sufficiently small ϵ with their sets of vertex barcodes coinciding. Let $\mathbf{b} \in \{0,1\}^N$ be some binary vector such that $\mathbf{p} = \bar{X}^T \mathbf{b}$ is a vertex of $\mathcal{Z}$. Then we know that some affine hyperplane $H \subseteq \mathbb{R}^{d+1}$ exists such that $H \cap \mathcal{Z} = \{\mathbf{p}\}$.

Let $\mathbf{q} = \bar{X}^T \mathbf{c}$, where $\mathbf{c} \in \{0,1\}^N$, be the image of an arbitrary vertex of the hypercube such that $\mathbf{q} \neq \mathbf{p}$. Note that $\mathbf{q}$ is not necessarily a vertex of $\mathcal{Z}$. We thus find some $\delta > 0$ such that the distance from $\mathbf{q}$ to the hyperplane H is greater than δ for every such $\mathbf{q}$. Furthermore, all such $\mathbf{q}$ will lie in exactly one of the half-spaces formed by the hyperplane.

Let $H' \subseteq \mathbb{R}^{d+1}$ be the affine hyperplane formed by shifting H by $P^T \mathbf{b}$. If we let $\mathbf{p}' = \bar{X}'^T \mathbf{b}$, it is straightforward to see that $\mathbf{p}' \in H'$. Hence H' intersects the perturbed zonotope $\mathcal{Z}'$.

Let $\mathbf{q}' = \bar{X}'^T \mathbf{c}$. Note that we can write $\mathbf{p}' = \mathbf{p} + P^T \mathbf{b}$ and $\mathbf{q}' = \mathbf{q} + P^T \mathbf{c}$. Let us now bound $\|P^T \mathbf{b}\|_2$. We see that $\|\mathbf{b}\|_2 \leq \sqrt{N}$. Using known relations between matrix norms, we see that $\|P^T\|_2 \leq \|P^T\|_F \leq \sqrt{N}\epsilon$. Hence $\|P^T \mathbf{b}\|_2 \leq N\epsilon$. By the exact same logic, we see that $\|P^T \mathbf{c}\|_2 \leq N\epsilon$.

Now suppose that we choose $\epsilon < \frac{\delta}{2N}$. By the triangle inequality, we can see that $\mathbf{q}'$ can move a distance at most $2N\epsilon < \delta$ relative to the hyperplane H'. Since the distance from $\mathbf{q}$ to H was greater than δ, we see that every such $\mathbf{q}'$ must lie on the same side of the hyperplane H'. Hence $H' \cap \mathcal{Z}' = \{\mathbf{p}'\}$, which implies that $\mathbf{p}'$ is a vertex of $\mathcal{Z}'$. Hence every vertex barcode of $\mathcal{Z}$ is a vertex barcode of $\mathcal{Z}'$. As $\mathcal{D}_\epsilon$ will also be in general position for small enough ϵ, we can swap the roles of $\mathcal{Z}$ and $\mathcal{Z}'$ in our proof to see that every vertex barcode of $\mathcal{Z}'$ is a vertex barcode of $\mathcal{Z}$. It thus follows that there is a one-to-one correspondence between vertex barcodes of $\mathcal{Z}$ and $\mathcal{Z}'$.

By the relationship between zonotope vertices and activation regions shown in section 3, we have thus proved that the set of convex optimization problems for $\mathcal{D}$ are a slightly perturbed version of the convex optimization problems for $\mathcal{D}_\epsilon$.

As any subset of a set of vectors in general position is also in general position, we can see that the solution of each convex optimization problem is continuous with respect to perturbations of the data (Agrawal et al., 2019). The global minimum of the loss is given by the minimum over the set of per-activation-region local minima. Hence the global minimum is continuous with respect to the training dataset as it is the composition of two continuous functions.

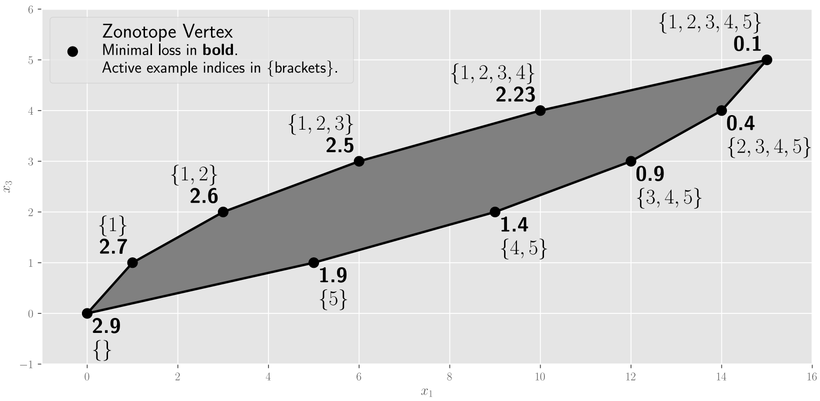

Figure 4: The zonotope $\mathcal{Z}$ associated to the dataset introduced in appendix D.1.

D Examples of Discontinuities at Datasets not in General Position

This section provides examples of the two sources of discontinuities of the minimal loss of a dataset when it is not in general position.

D.1 Convex Problem Associated to a Vertex being Discontinuous

Here we provide an example of a dataset whose optimal loss is not continuous with respect to the dataset. For this dataset, the optimal vertex is the same and exists in both the original and perturbed zonotopes. This means that its associated convex optimization problem is discontinuous with respect to the dataset.

Let us consider the dataset $\mathcal{D} \subseteq \mathbb{R}^2 \times \mathbb{R}$ given by

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 1 & 2 & 2.5 & 4 & 5 \end{bmatrix}.$$

Note that $\mathcal{X}$ lies entirely within the $x_2 = 0$ hyperplane and thus is not in general position.

Now consider the problem of optimizing a single affine ReLU over $\mathcal{D}$ with respect to the L1 loss. We assume a linear second layer and take the ReLU's corresponding second layer weight to be 1. The zonotope $\mathcal{Z}$ associated to this optimization problem is presented in fig. 4.[1] Each vertex has been labeled with its minimal loss in bold and with its set of active example indices. The vertex with the smallest loss of 0.1 is active on all of the examples.

Now consider what happens when we perform the following perturbation on the dataset

$$X_\epsilon = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & \epsilon & 0 & 0 \end{bmatrix},$$

where $\epsilon > 0$ is arbitrarily small. Let $\mathcal{D}_\epsilon = (X_\epsilon, Y)$. If we set the parameters $(\mathbf{w}, b)$ of our ReLU to $\mathbf{w} = \begin{bmatrix} 1 & -\frac{1}{2\epsilon} \end{bmatrix}^T$ and $b = 0$, we see that we fit $\mathcal{D}_\epsilon$ exactly and thus obtain zero L1 loss. These parameters belong to the same vertex as the global minimum of the unperturbed dataset. Hence we conclude that the convex optimization problem associated to this vertex is discontinuous with respect to the dataset.

D.2 Global Loss in New Vertex of Perturbed Zonotope

Here we provide an example of a dataset whose optimal loss is not continuous with respect to the dataset. For this dataset, the optimal vertex in the perturbed zonotope does not exist in the original zonotope.

---

[1] Technically this is a slice of the zonotope along the $x_2 = 0$ plane. The full zonotope is equal to the cylinder $\mathcal{Z} + \mathbb{R}\mathbf{e}_2$.

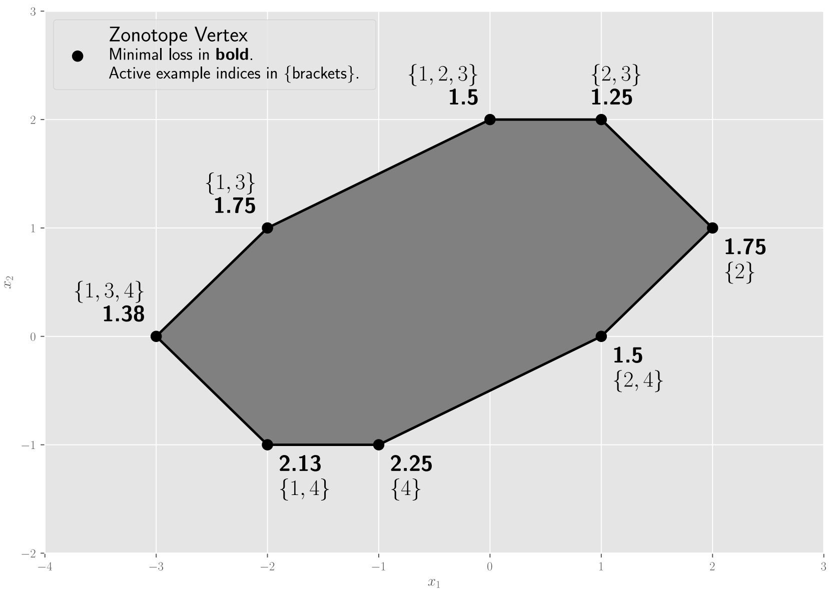

Figure 5: The zonotope $\mathcal{Z}$ associated to the dataset introduced in appendix D.2.

Let us consider the dataset $\mathcal{D} \subseteq \mathbb{R}^3 \times \mathbb{R}$ given by

$$X = \begin{bmatrix} -1 & 2 & -1 & -1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}.$$

Consider the problem of optimizing a single linear ReLU over $\mathcal{D}$ with an L1 loss. Assume that the second layer is linear with the ReLU's output weight set to 1. Note that $\mathcal{X}$ lies entirely within the $x_3 = 0$ plane and thus is not in general position. The zonotope $\mathcal{Z}$ associated to this optimization problem is presented in fig. 5.[2] Each vertex has been labeled with its minimal loss in bold and with its set of active example indices. The vertex with the smallest loss of 1.25 has the examples $\{\mathbf{x}_2, \mathbf{x}_3\}$ active.

Now consider what happens when we perform the following perturbation on the dataset

$$X_\epsilon = \begin{bmatrix} -1 & 2 & -1 & -1 \\ 0 & 1 & 1 & -1 \\ 0 & \epsilon & 0 & 0 \end{bmatrix} \tag{10}$$

where $\epsilon > 0$ is arbitrarily small. Let $\mathcal{D}_\epsilon = (X_\epsilon, Y)$. The global minimum of the loss for $\mathcal{D}_\epsilon$ occurs at the parameter value $\mathbf{w} = \begin{bmatrix} -\frac{3}{2} & \frac{1}{2} & \frac{11}{2\epsilon} \end{bmatrix}^T$. The loss value at this point is 0.625, and these parameters are associated to the zonotope vertex with all examples active. As evident from fig. 5, such a vertex does not exist in the unperturbed zonotope.

E NP-Hardness

Goel et al. (2020) prove the NP-hardness of optimizing a single ReLU by reducing solving an instance of the NP-hard set cover problem to optimizing a single ReLU over a train dataset. In the set cover problem, we are given a collection $\mathcal{T} = \{T_1, \ldots, T_M\}$ of subsets of a given set U. Given some $t \in \mathbb{N}$, the goal is determine whether a subcollection $\mathcal{S} \subseteq \mathcal{T}$ exists such that $U = \bigcup_{S \in \mathcal{S}} S$ and $|\mathcal{S}| \leq t$.

---

[2] Technically this is a slice of the zonotope along the $x_3 = 0$ plane. The full zonotope is equal to the cylinder $\mathcal{Z} + \mathbb{R}\mathbf{e}_3$.

E.1 Reduction to ReLU Optimization

Goel et al. (2020) use a single ReLU without a bias as their model, so we may write our network as

$$f_{\mathbf{w}}(\mathbf{x}) = \phi(\mathbf{w}^T\mathbf{x}). \tag{11}$$

The input dimension of their model is $d = M + 2$. Of these dimensions, M correspond to members of $\mathcal{T}$ and two are used as "constraint coordinates". We use $\mathbf{e}_{T_i}$ to denote a unit coordinate vector corresponding to T_i, and $\mathbf{e}_\gamma$ and $\mathbf{e}_1$ as the unit coordinate vectors for the constraint coordinates.

Set $\gamma = 0.01/M^2$. Overall, they create $N = |U| + M + 2$ labeled training examples. For the constraint coordinates, they create the examples

$$(\mathbf{x}_\gamma, y_\gamma) = (\mathbf{e}_\gamma, \gamma) \tag{12}$$

and

$$(\mathbf{x}_1, y_1) = (\mathbf{e}_1, 1). \tag{13}$$

For each $T_i \in \mathcal{T}$, they create the example

$$(\mathbf{x}_{T_i}, y_{T_i}) = (\mathbf{e}_\gamma + \mathbf{e}_{T_i}, \gamma). \tag{14}$$

For each $u \in U$, they create the example

$$(\mathbf{x}_u, y_u) = (\mathbf{e}_1 + \sum_{T_i \ni u} \mathbf{e}_{T_i}, 0). \tag{15}$$

Let $\mathcal{D}$ denote the entire labeled dataset, and let $\mathcal{X}$ denote the just the examples without labels. Note that both $\mathcal{D}$ and $\mathcal{X}$ will generally be multisets since if $u, u' \in U$ belong to exactly the same set of subsets in $\mathcal{T}$, then $\mathbf{x}_u = \mathbf{x}_{u'}$.

Using mean squared error, the training loss of the network can be written as

$$L(\mathbf{w}) = \frac{1}{N} \sum_{(\mathbf{x},y)\in\mathcal{D}} (f_{\mathbf{w}}(\mathbf{x}) - y)^2. \tag{16}$$

Goel et al. (2020) show that if the union of t or fewer members of $\mathcal{T}$ equals U, then the global minimum of this loss will be less than or equal to $t\gamma^2/N$. The weights $\mathbf{w} \in \mathbb{R}^d$ corresponding to this optima will have $w_\gamma = \gamma$, $w_1 = 1$, $w_{T_i} = -1$ for all $T_i \in \mathcal{S}$, and all other parameters set to zero.

The activation pattern for this optima will have $\mathbf{x}_1$ and $\mathbf{x}_\gamma$ being active while $\mathbf{x}_u$ is inactive for every $u \in U$. Any $\mathbf{x}_{T_i}$ will be inactive if $T_i \in \mathcal{S}$ and active otherwise.

E.2 Discontinuous Response to Perturbation

Here, we demonstrate how to create a perturbed dataset $\mathcal{D}'_\epsilon$ so that the the global minimum of loss will always be at most γ^2/N regardless of the solution to the set cover problem. This will hold true even as the scale of the perturbation $\epsilon > 0$ approaches 0.

In this perturbation, we pick an arbitrary $T \in \mathcal{T}$ and set $\mathbf{x}'_u = \mathbf{x}_u + \epsilon\mathbf{e}_T$ for the examples corresponding to all $u \in U$. All other examples are left unchanged.

Clearly, $\|\mathbf{x} - \mathbf{x}'\|_2 \leq \epsilon$ for all original-perturbed example pairs. To find a $\mathbf{w}$ with a loss value of γ^2/N, set $w_\gamma = \gamma$, $w_1 = 1$, $w_T \leq -\epsilon^{-1}$, and set all other parameters to 0. In this case, the model's predictions are correct for every training example except $(\mathbf{x}_T, y_T)$. In this case, the model predicts 0 while the label is γ, so the total loss is γ^2/N.

Note that the L2 norm of the parameters at the global minimum for the perturbed dataset approaches infinity as the size of the perturbation ϵ approaches 0.

The activation pattern at this optima will be the same as the activation pattern at the unperturbed optima except that $\mathbf{x}_T$ will be inactive while the rest of the $\mathbf{x}_{T_i}$ will be active.

We note that this activation pattern is achievable on the unperturbed dataset. However, it requires setting w_1 to a small positive value, w_γ to a relatively large positive value, all w_{T_i} for $T_i \neq T$ to moderate negative values, and w_T to a large enough negative value. For example, setting $w_1 = \gamma$, $w_\gamma = 2$, $w_{T_i} = -1$ for $T_i \neq T$, and $w_T = -3$ works. Hence as discussed in section 4.1.1, this discontinuity corresponds to a discontinuity in the constrained convex optimization problem (3) associated to a vertex rather than the new optimum occurring at a vertex not present in the original zonotope.

E.3 Reduction to Dataset in General Position (Proof of Theorem 4.2)

It is possible to perturb some of the examples in appendix E.1 to get a dataset in general position that is still the reduction of the subset-sum problem.

Theorem E.1. *Let $\delta_1, \delta_2 \in \mathbb{R}$ be constants satisfying $0 < \delta_1 < \delta_2 < \frac{1}{2d}$. For each $u \in U$, replace $\mathbf{x}_u$ in the dataset $\mathcal{D}$ with let $\mathbf{x}'_u = \mathbf{x}_u - \boldsymbol{\eta}_u$, where $\boldsymbol{\eta}_u \in \mathbb{R}^d$ is noise sampled IID from the uniform distribution on $[\delta_1, \delta_2]^d$. Denote this updated dataset as $\mathcal{D}'$ and let $\mathcal{X}'$ denote its examples without labels. Then $\mathcal{X}'$ is in general linear position with probability 1, and the global minimum of*

$$L'(\mathbf{w}) = \frac{1}{N} \sum_{(\mathbf{x},y)\in\mathcal{D}} (f_{\mathbf{w}}(\mathbf{x}) - y)^2 \tag{17}$$

is less than or equal to $t\gamma^2/N$ if and only if a set cover of $\mathcal{T}$ exists containing t sets.

The rest of this section is devoted to the proof of theorem E.1. We first prove that $\mathcal{X}'$ is indeed in general position. We then prove both directions of the if and only if statement.

E.3.1 General Position of Perturbed Dataset

Lemma E.2. *The examples of the perturbed dataset $\mathcal{X}'$ are in general linear position.*

Proof. We examine linear rather than affine dependencies between examples since the ReLU that we are using has no bias. Let us partition the training examples as

$$\mathcal{X}' = \mathcal{X}'_U \cup \mathcal{X}_{\mathcal{T}} \cup \{\mathbf{x}_1, \mathbf{x}_\gamma\}. \tag{18}$$

where $\mathcal{X}_{\mathcal{T}}$ corresponds the examples (14) and $\mathcal{X}'_U$ to corresponds to the examples in (15) with perturbations as in theorem E.1.

From their definitions, it is clear that the set of N vectors $\mathcal{X}_{\mathcal{T}} \cup \{\mathbf{x}_1, \mathbf{x}_\gamma\}$ are linearly independent and thus in general position. Since the $\{\boldsymbol{\eta}_u\}_{u\in U}$ are sampled IID from the uniform distribution on $[\delta_1, \delta_2]^d$, the introduction of the examples $\mathcal{X}'_U$ almost surely introduces no new nontrivial linear dependencies. ☐

E.3.2 Set Cover of Required Size Exists

Now suppose that a set cover $\mathcal{S} \subseteq \mathcal{T}$ of size t exists. Choose parameters $\mathbf{w} \in \mathbb{R}^d$ with coordinates $w_1 = 1$, $w_\gamma = \gamma$, $w_{T_i} = -2$ if $T_i \in \mathcal{S}$, and $w_{T_i} = 0$ otherwise.

Lemma E.3. *The loss on the perturbed dataset is equal to $t\gamma^2/N$ when the parameters are set to* $\mathbf{w}$.

Proof. We start by looking at the examples that are unchanged from the original dataset $\mathcal{D}$ in our perturbed version $\mathcal{D}'$. We see that $f_{\mathbf{w}}(\mathbf{x}_\gamma) = w_\gamma = \gamma = y_\gamma$ and $f_{\mathbf{w}}(\mathbf{x}_1) = w_1 = 1 = y_1$, so these two examples have a loss of zero. When $T_i \in \mathcal{S}$, we have $f_{\mathbf{w}}(\mathbf{x}_{T_i}) = \phi(w_\gamma + w_{T_i}) = \phi(\gamma - 2) = 0$ since $\gamma < 2$, so these examples incur a loss of $y_{T_i}^2/N = \gamma^2/N$. When $T_i \notin \mathcal{S}$, we have $f_{\mathbf{w}}(\mathbf{x}_{T_i}) = \phi(w_\gamma + w_{T_i}) = \phi(\gamma) = \gamma = y_{T_i}$, so these examples have a loss of zero. Overall these examples contribute a total of $t\gamma^2/N$ to the loss.

Now consider the examples $\mathbf{x}'_u \in \mathcal{X}'_U$. By definition, $\mathbf{x}'_u = \mathbf{x}_u - \boldsymbol{\eta}_u = \mathbf{e}_1 + \sum_{T_i \ni u} \mathbf{e}_{T_i} - \boldsymbol{\eta}_u$. The preactivation for such an example is

$$\begin{aligned}\mathbf{w}^T\mathbf{x}'_u &= w_1 + \sum_{T_i \ni u} w_{T_i} - \mathbf{w}^T\boldsymbol{\eta}_u \\ &= 1 - \sum_{u\in T_i\in\mathcal{S}} w_{T_i}2 - \mathbf{w}^T\boldsymbol{\eta}_u.\end{aligned}$$

Since $\mathcal{S}$ is a set cover of U, we know at least one $T_i \in \mathcal{S}$ exists such that $u \in T_i$. Hence $1 - \sum_{u\in T_i\in\mathcal{S}} w_{T_i}2 \le -1$. Recall that the entries of $\boldsymbol{\eta}_u$ are all positive and less than δ_2, and recall that all entries of $\mathbf{w}$ are greater than or equal to -2. Thus, $-\mathbf{w}^T\boldsymbol{\eta}_u \le 2d\delta_2$. Because we defined δ_2 such that $\delta_2 < \frac{1}{2d}$, we see that $2d\delta < 1$. Hence $-\mathbf{w}^T\boldsymbol{\eta}_u < 1$, so we see that the preactivation is less than $-1 + 1 = 0$. Thus applying a ReLU activation to this preactivation will output a 0. Thus $f_{\mathbf{w}}(\mathbf{x}'_u) = 0 = y'_u$ for all $u \in U$. Hence the examples from $\mathcal{X}'_U$ do not contribute to the loss, and the loss at $\mathbf{w}$ is $t\gamma^2/N$. ☐

E.3.3 Set Cover of Required Size Does Not Exist

Throughout this section, let $\mathbf{w} \in \mathbb{R}^d$ be parameter values as defined in the previous section. We first prove the following lemma.

Lemma E.4. *If the minimal loss over the original dataset $\mathcal{D}$ is less than or equal to $t\gamma^2/N$, then the minimal loss over the perturbed dataset $\mathcal{D}'$ is less than or equal to $t\gamma^2/N$.*

Proof. From the proof of Theorem 8 in Goel et al. (2020), we know that the minimal loss over the original dataset is less than or equal to $t\gamma^2/N$ only if a set cover of size t exists. This lemma then follows directly from lemma E.3. ☐

We now have the following.

Lemma E.5. *If the minimal loss over the perturbed dataset $\mathcal{D}'$ is greater than $t\gamma^2/N$, then no set cover of U exists that is comprised of t or fewer sets from $\mathcal{T}$.*

Proof. The contraposition of lemma E.4 states that if the the minimal loss over the perturbed dataset $\mathcal{D}'$ is greater than $t\gamma^2/N$, then the minimal loss over the original dataset $\mathcal{D}$ is greater than $t\gamma^2/N$. From the proof of Theorem 8 in Goel et al. (2020), this implies that no set cover of U exists that consists of t or fewer sets from $\mathcal{T}$. ☐

F Proof for Upper Bound on Required Overparameterization

In this section, we provide a proof of theorem 4.3. We begin with the following lemma.

Lemma F.1. *Let $A, B \subseteq \mathbb{R}^d$ be finite subsets such that $|B| = d + 1$ and the d-th coordinate of any $\mathbf{a} \in A$ is strictly less than the d-th coordinate of any $\mathbf{b} \in B$. Furthermore, assume that their union $A \cup B$ is in general position. For any $\mathbf{b} \in B$, let $y_{\mathbf{b}} \in \mathbb{R}$ be its label. Then there exist parameters $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{d+1}$ satisfying*

$$\begin{aligned} \mathbf{w}_1^T\bar{\mathbf{a}} &\le 0, & \mathbf{w}_1^T\bar{\mathbf{b}} &\ge 0, \\ \mathbf{w}_2^T\bar{\mathbf{a}} &\le 0, & \mathbf{w}_2^T\bar{\mathbf{b}} &\ge 0, \end{aligned}$$

for all $\mathbf{a} \in A$ and $\mathbf{b} \in B$ such that the function

$$f(\mathbf{x}) = \phi(\mathbf{w}_1^T\bar{\mathbf{x}}) - \phi(\mathbf{w}_2^T\bar{\mathbf{x}}) \tag{19}$$

satisfies $f(\mathbf{a}) = 0$ for all $\mathbf{a} \in A$ and $f(\mathbf{b}) = y_{\mathbf{b}}$ for all $\mathbf{b} \in B$.

Proof. Let $\alpha \in \mathbb{R}$ be some value that is strictly greater than the d-th coordinate of any $\mathbf{a} \in A$ and strictly less than the d-th coordinate of $\mathbf{b} \in B$. Such an α is guaranteed to exist based on the assumptions of the lemma. Now let $\mathbf{u} = \mathbf{e}_d - \alpha\mathbf{e}_{d+1} \in \mathbb{R}^{d+1}$. It is clear that $\mathbf{u}^T\bar{\mathbf{a}} < 0$ for all $\mathbf{a} \in A$ and $\mathbf{u}^T\bar{\mathbf{b}} > 0$ for all $\mathbf{b} \in B$. We can thus multiply $\mathbf{u}$ by a positive scalar to get a vector $\tilde{\mathbf{u}} \in \mathbb{R}^{d+1}$ such that $\tilde{\mathbf{u}}^T\bar{\mathbf{a}} < -1$ for all $\mathbf{a} \in A$ and $\tilde{\mathbf{u}}^T\bar{\mathbf{b}} > 1$ for all $\mathbf{b} \in B$.

Let us now look at the unconstrained problem of finding a $\mathbf{w} \in \mathbb{R}^{d+1}$ such that $\mathbf{w}^T\bar{\mathbf{b}} = y_{\mathbf{b}}$ for all $\mathbf{b} \in B$. As B contains $d + 1$ examples in general positions, such a $\mathbf{w}$ will always exist and can be found via standard linear regression.

Define

$$\beta_A = \max_{\mathbf{a} \in A} \phi(\mathbf{w}^T\bar{\mathbf{a}}) \tag{20}$$

and

$$\beta_B = \max_{\mathbf{b} \in B} \phi(-\mathbf{w}^T\bar{\mathbf{b}}). \tag{21}$$

Let $\beta = \max\{\beta_A, \beta_B\}$. Then setting $\mathbf{w}_1 = \mathbf{w} + \beta\tilde{\mathbf{u}}$ and $\mathbf{w}_2 = \beta\tilde{\mathbf{u}}$ satisfies the conditions of the lemma. ☐

Algorithm 3 Modified Greedy Local Search (mGLS) Heuristic

```
Input: data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, output weights $\mathbf{v} \in \mathbb{R}^{m+1}$, max steps $T \in \mathbb{N}$
$A_0 \in \mathrm{vert}(\mathcal{Z}^m)$ {Random initial zonotope vertex.}
for $t \in \{0, \ldots, T\}$ do
  $A_{t+1} \leftarrow A_t$
  $W^* \leftarrow$ solution of (3) for $A_t$
  if any of the constraints in (3) are equalities then
    $A_f \leftarrow A_t$ with all those constraints flipped
    $N_f \leftarrow$ subset of neighbors of $A_t$ differing only on one of those constraints
    $\mathcal{N} \leftarrow (\{A_f\}, N_f, \mathrm{neighbors}(A_t) \setminus N_f)$
  else
    $\mathcal{N} \leftarrow (\mathrm{neighbors}(A_t))$
  end if
  for $G \in \mathcal{N}$ do
    for $A' \in G$ do
      if $\mathcal{L}^*(A'; \mathcal{D}) < \mathcal{L}^*(A_{t+1}; \mathcal{D})$ then
        $A_{t+1} \leftarrow A'$
        continue main loop
      end if
    end for
  end for
  if $A_{t+1} = A_t$ then
    return $A_t$
  end if
end for
return $A_T$
```

We are now ready for the proof. Let $\mathcal{D}$ and $\mathcal{D}_k$ for $k = 1, \ldots, \lceil \frac{N}{d+1} \rceil$ be defined as in section 4.2.1. Let us do a proof by induction on k. Suppose that f is a ReLU network with $2\lceil \frac{N}{d+1} \rceil - 2$ hidden units fitting the labels in $\bigcup_{k'=1}^{k-1} \mathcal{D}_{k'}$ exactly.

Let us now relabel the entire dataset by subtracting the predictions of f from the labels to get

$$\mathcal{D}' = \{(\mathbf{x}, y - f(\mathbf{x})) \mid (\mathbf{x}, y) \in \mathcal{D}\} . \tag{22}$$

Define $\mathcal{D}'_k$ accordingly. Clearly, we have the labels being all zero for all examples in $\mathcal{D}' \setminus \mathcal{D}'_k$. The labels for examples in $\mathcal{D}'_k$ will generally be non-zero.

We can use lemma F.1 to find a unit layer network g such that $g(\mathbf{x}) = 0 = y$ for all $(\mathbf{x}, y) \in \mathcal{D}' \setminus \mathcal{D}'_k$ and $g(\mathbf{x}) = y$ for all $(\mathbf{x}, y) \in \mathcal{D}'_k$. Hence g fits $\mathcal{D}'$ exactly. From this it is clear that $f + g$ fits the original dataset $\mathcal{D}$ exactly. We can find a ReLU network with $2\lceil \frac{N}{d+1} \rceil$ hidden units representing $f + g$ by having its last two units be the units of g and the remaining units be the units of f.

G Modified Greedy Local Search

This section provides more information on the additional heuristics used in the mGLS algorithm introduced in section 4.3.1. The purpose of these modifications is to reduce the typical number of convex problems that we have to solve in a run of the algorithm.

The major difference is that as we iterate over neighboring zonotope vertices, we move to any vertex with a lower loss than the current vertex. This is in contrast to algorithm 2, which evaluates the loss at every neighbor and moves to the one with the lowest loss. Especially near the start of the optimization procedure, we find that this greatly reduces the number of vertices that we need to solve convex programs for. The order in which we iterate over the neighbors is mostly random with the caveat discussed below.

We also make use of geometric information coming the optimal parameter values given the current vertex to preferentially try some subsets of neighboring vertices first. If they lie at the boundary of the current activation region, then it stands to reason that activation regions on the other side of that boundary are more likely to have better solutions. Solutions lying on the boundary of an activation region have a subset of preactivations that are exactly zero. Equivalently, a subset of the inequalities in (3) become equalities at the solution. In such cases, we first try the vertex that has all of those constraints flipped. Note that this vertex is not usually a neighbor of the current current in the 1-skeleton of the zonotope and might not even be feasible. If feasible and k constraints are flipped, then that vertex and the current vertex belong to the same k-face of the zonotope. We then try the neighbors of the current vertex that correspond to flipping one of those constraints. Afterwards, we try the remaining neighbors.

H Experimental Details

H.1 Synthetic Data

H.1.1 Synthetic Dataset Generation

Here we present the details of the generation of the synthetic datasets used in the experiments in this paper.

We start out with the dimension of the input d and the number of units m_{gen} in the shallow ReLU network used to generate the labels. We use this to calculate the number of examples $N = (d+1)m_{\text{gen}}$. We then generate the examples $\{\mathbf{x}_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$ by sampling them i.i.d. from a standard normal distribution.

We use a randomly generated ReLU network to label these examples. We can express this network as

$$g(\mathbf{x}) = \mathbf{v}_{\text{gen}}^T \phi(W_{\text{gen}} \bar{\mathbf{x}}) + c_{\text{gen}}. \tag{23}$$

We generate the parameters $\mathbf{v}_{\text{gen}} \in \mathbb{R}^{m_{\text{gen}}}$, $W_{\text{gen}} \in \mathbb{R}^{m_{\text{gen}} \times (d+1)}$, and $c_{\text{gen}} \in \mathbb{R}$ by via sampling from standard normal distributions. The label for the i-th example can then be expressed as $y_i = g(\mathbf{x}_i)$.

H.1.2 Training Details

All experiments on the synthetic datasets used the mean squared error (MSE) loss. The network architecture for these experiments takes the form of

$$f(\mathbf{x}) = \mathbf{v}^T \phi(W \bar{\mathbf{x}}) + c, \tag{24}$$

where $\mathbf{v} \in \mathbb{R}^m$, $W \in \mathbb{R}^{m \times (d+1)}$, and $c \in \mathbb{R}$. We use $m \in \mathbb{N}$ to denote the number of units in the network that we train on the dataset.

Gradient Descent All experiments involving gradient descent on synthetic datasets in this paper used batch gradient descent with a learning rate of 1e-3 for 400,000 steps. All parameters, including the second layer weights $\mathbf{v}, c$, were trained. We used the parameter initialization scheme from Glorot & Bengio (2010).

Random Vertex A random vertex was selected by sampling the first layer weights $W \in \mathbb{R}^{m \times (d+1)}$ from a standard Gaussian and taking its corresponding activation pattern over the dataset. We randomly initialize $\mathbf{v}$ with values chosen uniformly from the set $\{-1, 1\}$. Optimizing over all of the parameters of a shallow ReLU network within a single activation region is non-convex. However, the problems of training W, c given a fixed $\mathbf{v}$ and training $\mathbf{v}, c$ given a fixed W are convex. The former is a slight variant of (3) while the latter is simple linear regression over fixed features. We thus iterate between solving these two problems until we converge to fixed loss value.

This process is guaranteed to converge to a local minima of the loss (Xu & Yin, 2013). However, it is possible that our process of optimizing the second layer weights here is suboptimal and does not reach the global optimum within the activation region.

GLS Heuristic Here we fix $\mathbf{v} \in \mathbb{R}^m$ to have $m/2$ entries set to -1 and $m/2$ entries set to +1. We slightly modify the convex problem (3) to include optimizing the $c \in \mathbb{R}$ in addition to the first layer weights $W \in \mathbb{R}^{m \times (d+1)}$. We choose the starting vertex at random by sampling the first layer weights W from a standard Gaussian and taking its corresponding activation pattern over the dataset. We set $T = 1024$ as the maximum number of steps.

Table A1: Results of all synthetic data experiments performed in this paper. The large numbers are the median final MSE over 16 runs for GLS heuristic and over 8 runs for gradient descent and random vertex. The subscript numbers provide the standard deviation over the runs. Some cells are empty as the particular combination of $d, m_{\text{gen}}, m,$ and optimization method was not needed for our set of comparisons.

d	m_{GEN}	m	GRADIENT DESCENT	RANDOM VERTEX	GLS HEURISTIC
4	2	2	3.82E-10$_{\text{3.2E-04}}$	1.25E-01$_{\text{1.7E-01}}$	8.27E-01$_{\text{1.4E00}}$
4	2	3	3.43E-11$_{\text{7.6E-10}}$	3.85E-05$_{\text{1.2E-02}}$	–
4	2	4	6.01E-12$_{\text{3.7E-09}}$	4.10E-08$_{\text{8.1E-07}}$	1.58E-01$_{\text{5.3E-01}}$
4	2	8	–	8.81E-09$_{\text{1.4E-08}}$	8.29E-13$_{\text{1.7E-02}}$
4	2	16	–	1.13E-08$_{\text{7.3E-08}}$	4.11E-17$_{\text{1.2E-16}}$
4	4	4	8.30E-03$_{\text{1.8E-01}}$	1.98E-03$_{\text{6.6E-02}}$	8.11E-01$_{\text{8.5E-01}}$
4	4	6	5.04E-12$_{\text{5.7E-10}}$	3.13E-04$_{\text{7.7E-04}}$	–
4	4	8	6.54E-12$_{\text{9.7E-10}}$	4.70E-05$_{\text{5.3E-05}}$	8.47E-09$_{\text{7.8E-02}}$
4	4	16	–	5.08E-07$_{\text{2.8E-05}}$	1.68E-12$_{\text{1.6E-11}}$
4	4	32	–	2.71E-07$_{\text{5.1E-06}}$	1.97E-15$_{\text{4.7E-15}}$
4	8	8	8.53E-03$_{\text{5.3E-03}}$	8.47E-03$_{\text{8.3E-03}}$	4.68E-01$_{\text{4.0E-01}}$
4	8	12	2.52E-11$_{\text{4.3E-10}}$	7.59E-04$_{\text{9.8E-04}}$	–
4	8	16	2.21E-11$_{\text{2.6E-09}}$	6.07E-04$_{\text{6.9E-04}}$	9.96E-03$_{\text{3.1E-02}}$
4	8	32	–	1.70E-05$_{\text{7.9E-05}}$	6.68E-12$_{\text{2.9E-10}}$
4	8	64	–	6.26E-06$_{\text{4.8E-05}}$	3.49E-14$_{\text{2.8E-14}}$
8	4	4	2.54E-03$_{\text{1.2E-01}}$	1.44E-02$_{\text{4.4E-02}}$	3.74E00$_{\text{2.0E00}}$
8	4	6	6.11E-11$_{\text{1.4E-02}}$	1.11E-03$_{\text{2.6E-02}}$	–
8	4	8	7.69E-12$_{\text{8.2E-02}}$	6.96E-08$_{\text{4.2E-05}}$	4.94E-01$_{\text{4.7E-01}}$
8	4	16	–	2.24E-08$_{\text{2.6E-08}}$	5.33E-12$_{\text{4.1E-08}}$
8	4	32	–	6.25E-09$_{\text{2.5E-09}}$	2.40E-13$_{\text{1.1E-12}}$
8	8	8	7.47E-03$_{\text{1.2E-02}}$	2.45E-02$_{\text{9.9E-03}}$	3.67E00$_{\text{2.7E00}}$
8	8	12	7.30E-12$_{\text{1.2E-10}}$	1.69E-04$_{\text{3.1E-04}}$	–
8	8	16	5.83E-13$_{\text{7.0E-12}}$	2.07E-06$_{\text{9.5E-06}}$	1.77E00$_{\text{8.9E-01}}$
8	8	32	–	8.01E-08$_{\text{1.1E-07}}$	3.09E-12$_{\text{6.5E-11}}$
8	8	64	–	2.43E-08$_{\text{6.6E-09}}$	3.15E-13$_{\text{4.1E-13}}$
8	16	16	–	2.72E-02$_{\text{3.3E-02}}$	7.07E00$_{\text{2.7E00}}$
8	16	24	–	1.61E-03$_{\text{1.8E-03}}$	–
8	16	32	–	8.57E-05$_{\text{2.1E-04}}$	1.47E00$_{\text{6.3E-01}}$
8	16	64	–	5.62E-07$_{\text{4.8E-07}}$	2.35E-02$_{\text{3.9E-02}}$
8	16	128	–	1.03E-07$_{\text{7.6E-08}}$	3.16E-13$_{\text{4.7E-12}}$
16	8	8	–	2.79E-02$_{\text{4.5E-02}}$	1.85E01$_{\text{6.4E00}}$
16	8	12	–	9.05E-07$_{\text{2.2E-05}}$	–
16	8	16	–	1.10E-07$_{\text{2.0E-08}}$	8.76E00$_{\text{2.7E00}}$
16	8	32	–	2.86E-08$_{\text{6.4E-09}}$	1.74E00$_{\text{8.2E-01}}$
16	8	64	–	9.53E-09$_{\text{2.2E-09}}$	2.82E-13$_{\text{7.9E-11}}$
16	16	16	–	4.99E-02$_{\text{1.6E-02}}$	2.81E01$_{\text{6.4E00}}$
16	16	24	–	4.70E-05$_{\text{1.1E-04}}$	–
16	16	32	–	4.80E-07$_{\text{2.1E-06}}$	1.63E01$_{\text{3.0E00}}$
16	16	64	–	8.13E-08$_{\text{1.9E-08}}$	4.03E00$_{\text{1.0E00}}$
16	16	128	–	2.75E-08$_{\text{3.0E-09}}$	3.42E-13$_{\text{3.8E-12}}$

H.1.3 Full Results

We experimented with a range of d, m_{gen}, and m values. Present our full results in table A1. The scores represent the median of 16 runs for random vertex scores and the median of 8 runs for the rest.

H.2 Toy Versions of Real-World Datasets

H.2.1 Dataset Creation

Our datasets were created from the MNIST (LeCun et al., 2010) and Fashion MNIST (Xiao et al., 2017) datasets. Both datasets are 10-way multiclass classification datasets; however, our mGLS algorithm only works for ReLU networks with scalar output. Hence we have to create binary classification tasks from these datasets.

We did this by restricting each dataset to two classes and having the task to correctly differentiate between only those two classes. For MNIST, we chose the 4 and the 9 classes. For Fashion MNIST, we chose the pullover and the coat classes. These classes were chosen for the interclass similarity of their examples, which increases the difficulty of the task.

To reduce the dimensionality of the data, we performed principle components analysis (PCA) using the `scikit-learn` Python package (Pedregosa et al., 2011) on all of the training examples in each dataset belonging to their respective two chosen classes. When then used the first $d \in \{8, 16\}$ whitened components for our dataset. We then took the first $N \in \{350, 700\}$ examples in the training split as our training dataset. We always chose the same examples across experiments to reduce variance.

H.2.2 Training Details

Since these datasets were binary classification tasks, we used the sigmoid cross entropy loss function $\ell(\hat{y}, y) = -y\hat{y} + \sigma(\hat{y})$, where σ is the logistic sigmoid function. The network architecture for these experiments takes the form of

$$f(\mathbf{x}) = \mathbf{v}^T \phi(W\bar{\mathbf{x}}) + c, \tag{25}$$

where $\mathbf{v} \in \mathbb{R}^m$, $W \in \mathbb{R}^{m\times(d+1)}$, and $c \in \mathbb{R}$. We use $m \in \mathbb{N}$ to denote the number of units in the network that we train on the dataset. In all experiments in this section, set $\mathbf{v}$ to a vector containing half ones and half negative ones and froze it throughout training. The rest of the variables were optimized during training. Note that this is different than what we did for the synthetic datasets.

Gradient Descent We trained for one million steps with a learning rate of 1e-3 using batch gradient descent. We used the parameter initialization scheme from Glorot & Bengio (2010).

Random Vertex A random vertex was selected by sampling the first layer weights $W \in \mathbb{R}^{m\times(d+1)}$ from a standard Gaussian and taking its corresponding activation pattern over the dataset. We then solved its corresponding convex program (3) using the ECOS (Domahidi et al., 2013) solver in the `cvxpy` Python package (Diamond & Boyd, 2016). We also optimized the bias $c \in \mathbb{R}$ in the final layer as well as the first layer parameters parameters in the convex program.

mGLS Heuristic We chose the initial vertex by sampling the first layer weights $W \in \mathbb{R}^{m\times(d+1)}$ from a standard Gaussian and taking its corresponding activation pattern over the dataset. Like for the random vertex experiment, we also optimized the second layer bias $c \in \mathbb{R}$ in the convex program. We used the mGLS algorithm presented in appendix G to perform the optimization. We set $T = 2048$ as the maximum number of steps.

H.2.3 Full Results

We experimented with a range of d, N, m values on both MNIST 5/9 and Fashion MNIST coat/pullover. We present our full results comparing gradient descent to the random vertex method in table A2 and comparing gradient descent to mGLS in table A3. Random vertex results are the median of 16 runs while the results for the other two methods are the median of 8 runs.

Table A2: Results of all experiments comparing gradient descent to random vertex optimization in this paper. The subscripts provide standard deviation across runs.

DATASET				GRADIENT DESCENT		RANDOM VERTEX	
	d	N	m	LOSS	ACC (%)	LOSS	ACC (%)
MNIST	8	350	4	1.16E-01$_{1.1E-02}$	95.6$_{0.80}$	6.04E-01$_{6.7E-02}$	67.4$_{8.53}$
	8	350	8	5.37E-02$_{7.9E-03}$	98.9$_{0.45}$	5.27E-01$_{1.1E-01}$	72.9$_{8.30}$
	8	350	16	2.09E-02$_{2.6E-03}$	99.5$_{0.16}$	3.21E-01$_{7.3E-02}$	85.3$_{4.87}$
	8	350	32	8.38E-03$_{5.1E-04}$	100.0$_{0.00}$	2.30E-01$_{5.0E-02}$	91.4$_{2.36}$
	8	350	64	4.21E-03$_{1.5E-04}$	100.0$_{0.00}$	1.71E-01$_{4.9E-02}$	94.1$_{1.86}$
	8	700	4	1.62E-01$_{1.9E-03}$	93.2$_{0.73}$	6.50E-01$_{8.7E-02}$	63.4$_{9.13}$
	8	700	8	1.24E-01$_{6.7E-03}$	95.0$_{0.56}$	4.94E-01$_{8.6E-02}$	75.9$_{7.11}$
	8	700	16	7.90E-02$_{6.1E-03}$	97.6$_{0.49}$	3.95E-01$_{6.4E-02}$	82.1$_{4.34}$
	8	700	32	3.97E-02$_{1.5E-03}$	99.4$_{0.05}$	2.79E-01$_{5.7E-02}$	88.6$_{3.02}$
	8	700	64	1.94E-02$_{6.5E-04}$	100.0$_{0.04}$	2.14E-01$_{2.2E-02}$	91.5$_{1.06}$
	16	350	4	1.78E-02$_{2.2E-03}$	99.5$_{0.18}$	5.95E-01$_{7.8E-02}$	68.7$_{7.10}$
	16	350	8	5.75E-03$_{7.4E-04}$	100.0$_{0.00}$	4.89E-01$_{1.2E-01}$	75.4$_{9.28}$
	16	350	16	2.49E-03$_{2.8E-04}$	100.0$_{0.00}$	4.14E-01$_{1.1E-01}$	80.0$_{7.91}$
	16	350	32	1.12E-03$_{5.0E-05}$	100.0$_{0.00}$	1.39E-01$_{9.1E-02}$	95.0$_{3.95}$
	16	350	64	4.87E-04$_{2.2E-05}$	100.0$_{0.00}$	4.81E-02$_{3.9E-02}$	99.1$_{1.29}$
	16	700	4	3.51E-02$_{4.4E-03}$	99.3$_{0.18}$	6.61E-01$_{5.6E-02}$	57.7$_{7.61}$
	16	700	8	1.16E-02$_{1.3E-03}$	99.9$_{0.16}$	5.34E-01$_{9.0E-02}$	72.8$_{7.68}$
	16	700	16	4.61E-03$_{2.9E-04}$	100.0$_{0.00}$	4.37E-01$_{8.7E-02}$	78.9$_{6.01}$
	16	700	32	2.06E-03$_{1.1E-04}$	100.0$_{0.00}$	2.36E-01$_{6.3E-02}$	90.1$_{3.11}$
	16	700	64	1.00E-03$_{4.0E-05}$	100.0$_{0.00}$	1.20E-01$_{3.3E-02}$	95.6$_{1.44}$
FASHION MNIST	8	350	4	2.98E-01$_{1.0E-02}$	88.4$_{0.67}$	5.97E-01$_{7.2E-02}$	68.4$_{8.60}$
	8	350	8	2.24E-01$_{1.3E-02}$	91.4$_{0.71}$	5.58E-01$_{5.5E-02}$	72.9$_{6.28}$
	8	350	16	1.39E-01$_{9.8E-03}$	95.9$_{0.90}$	4.96E-01$_{5.7E-02}$	78.6$_{3.39}$
	8	350	32	6.71E-02$_{7.4E-03}$	98.8$_{0.43}$	3.78E-01$_{2.8E-02}$	85.1$_{1.68}$
	8	350	64	3.21E-02$_{2.3E-03}$	100.0$_{0.19}$	2.91E-01$_{3.1E-02}$	88.1$_{1.34}$
	8	700	4	3.52E-01$_{3.8E-03}$	84.8$_{0.53}$	6.62E-01$_{5.1E-02}$	60.5$_{7.12}$
	8	700	8	3.15E-01$_{6.4E-03}$	86.6$_{0.63}$	5.99E-01$_{4.4E-02}$	69.3$_{4.66}$
	8	700	16	2.48E-01$_{1.0E-02}$	90.5$_{0.76}$	5.17E-01$_{4.2E-02}$	76.1$_{3.20}$
	8	700	32	1.65E-01$_{5.0E-03}$	94.3$_{0.50}$	4.31E-01$_{3.0E-02}$	80.7$_{1.78}$
	8	700	64	8.70E-02$_{6.0E-03}$	98.3$_{0.37}$	3.66E-01$_{1.7E-02}$	84.4$_{1.27}$
	16	350	4	1.93E-01$_{1.6E-02}$	92.4$_{1.25}$	6.24E-01$_{6.2E-02}$	64.4$_{9.07}$
	16	350	8	8.33E-02$_{1.3E-02}$	98.0$_{0.70}$	5.46E-01$_{6.0E-02}$	72.7$_{5.67}$
	16	350	16	3.71E-02$_{3.7E-03}$	99.3$_{0.14}$	4.27E-01$_{5.2E-02}$	80.1$_{3.29}$
	16	350	32	1.43E-02$_{3.3E-03}$	100.0$_{0.20}$	3.00E-01$_{4.4E-02}$	87.3$_{2.31}$
	16	350	64	6.72E-03$_{4.3E-04}$	100.0$_{0.00}$	1.65E-01$_{1.0E-01}$	94.3$_{4.22}$
	16	700	4	2.84E-01$_{1.1E-02}$	88.1$_{0.48}$	6.30E-01$_{4.6E-02}$	64.8$_{5.66}$
	16	700	8	1.95E-01$_{8.5E-03}$	92.0$_{0.31}$	5.95E-01$_{4.9E-02}$	68.4$_{4.37}$
	16	700	16	1.13E-01$_{7.5E-03}$	96.5$_{0.58}$	5.19E-01$_{4.6E-02}$	75.7$_{3.27}$
	16	700	32	4.53E-02$_{3.4E-03}$	99.4$_{0.25}$	4.26E-01$_{2.4E-02}$	81.4$_{1.45}$
	16	700	64	2.05E-02$_{1.5E-03}$	100.0$_{0.06}$	3.43E-01$_{1.5E-02}$	85.6$_{1.14}$

Table A3: Results of all experiments comparing gradient descent to mGLS in this paper. The subscripts provide standard deviation across runs.

DATASET	m	GRADIENT DESCENT		MGLS HEURISTIC	
		LOSS	ACC (%)	LOSS	ACC (%)
MNIST	4	$1.16\text{E-}01_{1.1\text{E-}02}$	$95.6_{0.80}$	$1.09\text{E-}01_{3.7\text{E-}02}$	$95.6_{1.76}$
	8	$5.37\text{E-}02_{7.9\text{E-}03}$	$98.9_{0.45}$	$1.95\text{E-}03_{1.5\text{E-}02}$	$100.0_{0.58}$
	16	$2.09\text{E-}02_{2.6\text{E-}03}$	$99.5_{0.16}$	$8.24\text{E-}04_{1.2\text{E-}03}$	$100.0_{0.00}$
	32	$8.38\text{E-}03_{5.1\text{E-}04}$	$100.0_{0.00}$	$3.73\text{E-}02_{8.5\text{E-}03}$	$99.0_{0.61}$
FASHION MNIST	4	$2.98\text{E-}01_{1.0\text{E-}02}$	$88.4_{0.67}$	$2.88\text{E-}01_{1.8\text{E-}02}$	$88.3_{0.64}$
	8	$2.24\text{E-}01_{1.3\text{E-}02}$	$91.4_{0.71}$	$1.72\text{E-}01_{4.5\text{E-}02}$	$93.7_{1.99}$
	16	$1.39\text{E-}01_{9.8\text{E-}03}$	$95.9_{0.90}$	$2.91\text{E-}03_{1.5\text{E-}02}$	$100.0_{0.47}$
	32	$6.71\text{E-}02_{7.4\text{E-}03}$	$98.8_{0.43}$	$2.33\text{E-}02_{6.1\text{E-}02}$	$99.4_{2.45}$