The Convex Landscape of Neural Networks: Characterizing Global Optima and Stationary Points via Lasso Models

Tolga Ergen and Mert Pilanci

Abstract

Due to the non-convex nature of training Deep Neural Network (DNN) models, their effectiveness relies on the use of non-convex optimization heuristics. Traditional methods for training DNNs often require costly empirical methods to produce successful models and do not have a clear theoretical foundation. In this study, we examine the use of convex optimization theory and sparse recovery models to refine the training process of neural networks and provide a better interpretation of their optimal weights. We focus on training two-layer neural networks with piecewise linear activations and demonstrate that they can be formulated as a finite-dimensional convex program. These programs include a regularization term that promotes sparsity, which constitutes a variant of group Lasso. We first utilize semi-infinite programming theory to prove strong duality for finite width neural networks and then we express these architectures equivalently as high dimensional convex sparse recovery models. Remarkably, the worst-case complexity to solve the convex program is polynomial in the number of samples and number of neurons when the rank of the data matrix is bounded, which is the case in convolutional networks. To extend our method to training data of arbitrary rank, we develop a novel polynomial-time approximation scheme based on zonotope subsampling that comes with a guaranteed approximation ratio. We also show that all the stationary points of the nonconvex training objective can be characterized as the global optimum of a subsampled convex program. Our convex models can be trained using standard convex solvers without resorting to heuristics or extensive hyper-parameter tuning unlike non-convex methods. Due to the convexity, optimizer hyperparameters such as initialization, batch sizes, and step size schedules have no effect on the final model. Through extensive numerical experiments, we show that convex models can outperform traditional non-convex methods and are not sensitive to optimizer hyperparameters. The code for our experiments is available at https://github.com/pilancilab/convex_nn.

Index Terms

Neural networks, convex optimization, polynomial-time complexity, piecewise linear activations, stationary points, implicit regularization

I Introduction

Convex optimization has been a topic of interest due to several desirable properties that make it attractive for use in machine learning models. First and foremost, convex optimization problems in standard form are computationally tractable and typically admit a unique global optimum. In addition, standard convex optimization problems can be solved efficiently using well-established numerical solvers. This is in contrast to non-convex optimization problems, which can have multiple local minima and require heuristics to obtain satisfactory solutions.

The distinction between convex and non-convex optimization is of great practical importance for machine learning problems. In non-convex optimization, the choice of optimization method and its internal parameters such as initialization, mini-batching, and step sizes have a significant effect on the quality of the learned model. This is in sharp contrast to convex optimization, for which these hyperparameters have no effect, and solutions are often unique and are determined by the data and the model, as opposed to being a function of the training trajectory and hyperparameters as in the case of neural networks. Moreover, convex optimization solutions can be obtained in a robust, reproducible, and transparent manner.

T. Ergen is with LG AI Research, Ann Arbor, MI, 48105 USA e-mail: tergen@lgresearch.ai

M. Pilanci is with the Electrical Engineering Department, Stanford University, Stanford, CA, 94305 USA.

A preliminary version of part of this work was published at ICML 2020 with the title “Neural Networks are Convex Regularizers: Exact Polynomial-time Convex Optimization Formulations for Two-layer Networks”.

I-A Related work

Existing works on convex neural networks [6, 9, 34] consider neural networks of infinite width to enable convexification over a set of measures. Hence, these results do not apply to finite width neural networks that are used in practice. [9] proved that infinite width neural network training problems can be cast as a convex optimization problem with infinitely many variables. They also introduced an incremental algorithm that inserts a hidden neuron at a time by solving a maximization problem to obtain a linear classifier at each step. However, even though the algorithm may be used to achieve a global minimum for small datasets, it does not scale to high dimensional cases. In addition, [6] investigated infinite width convex neural network training, however, did not provide a computationally tractable algorithm. In particular, strategies based on Frank-Wolfe [6] require solving an intractable problem in order to train only a single neuron and do not optimize finite width networks. Similarly, [34] proved that in the infinite width limit, neural networks can be approximated as infinite dimensional convex learning models with an appropriate reparameterization.

Sparse recovery models have become an essential tool in a wide variety of disciplines such as signal processing and statistics and forms the foundation of compressed sensing [14, 24, 73]. The key idea behind these models is to leverage the sparsity inherent in many data sources to enable more efficient and accurate processing. A notable technique used in sparse recovery is the l1-norm minimization, also known as the Lasso, which encourages sparsity in the solution vector [15, 16]. Further developments, such as group ℓ_1-norm minimization, extend this idea to incorporate structured sparsity, where groups of variables are either jointly included or excluded from the model [78]. These sparse recovery models provide an elegant framework for finding meaningful and parsimonious representations of complex data, thereby allowing more effective analysis and interpretation.

In contrast to existing work on convexifying neural networks, in this paper, we introduce a novel approach to derive exact finite dimensional convex program representations for finite width networks. Our characterization parallels sparse recovery models studied in the compressed sensing literature. The principal innovation lies in our analysis of hyperplane arrangements, an area of study originating from Cover's work on linear classifiers (Cover, 1965). Our results are applicable to any piecewise linear activations such as the Rectified Linear Unit (ReLU).

I-B Our contributions

A preliminary work on convex formulations of ReLU networks appeared in [56]. Our contributions over this work and other previous studies can be summarized as follows:

- We introduce a convex analytic framework to describe the training of two-layer neural networks with piecewise linear activations (including ReLU, leaky ReLU, and absolute value activation) as equivalent finite dimensional convex programs that perform sparse recovery. We prove the polynomial-time trainability of these architectures by standard convex optimization solvers when the data matrix has bounded rank, as is the case of Convolutional Neural Networks (CNNs).
- In Theorem II.2, we prove that all of the stationary points of nonconvex neural networks correspond to the global optimum of a subsampled convex program. Therefore, we characterize all critical points of the nonconvex training problem which may be found via local heuristics as a global minimum of our subsampled convex program.
- We introduce a simple randomized algorithm to generate the hyperplane arrangements which are required to solve the convex program. We prove a theoretical bound on the number of required samples (Theorem III.2) by relating it to sampling vertices of zonotopes. This approach significantly simplifies convex neural networks, since prior work assumed that the arrangement patterns are computed through enumeration.
- One potential limitation of solving our convex program exactly is the exponential worst-case complexity when applied to data with unbounded rank, as is often the case in Fully Connected (FC) neural networks. In order to address this, we introduce an approximation algorithm (Theorem III.1) and prove strong polynomial-time approximation guarantees with respect to the global optimum. Combined with Theorem III.2, this enables a highly practical and simple method with strong guarantees.
- We show that the optimal solution of the convex program is typically extremely sparse due to a small number of effective hyperplane arrangements in practical applications. We propose a novel hyperplane arrangement sampling technique utilizing convolutions and achieve substantial performance improvements in standard benchmarks.

TABLE I: List of the neural network architectures that we study in this paper and the corresponding non-convex and convex training objectives.

	Model	Non-convex Objective	Convex Objective	Result
FC scalar output NN	$f_\theta(\mathbf{X}) = \phi\big(\mathbf{X}\mathbf{W}^{(1)}\big)\mathbf{w}^{(2)}$	(3)	(7)	Theorem II.1
Nonlinear CNN	$f_\theta(\mathbf{X}) = \frac{1}{K}\sum_{k,j} \phi(\mathbf{X}_k \mathbf{w}_j^{(1)}) w_j^{(2)}$	(13)	(7)	Section IV-A
Linear CNN	$f_{\theta,c}(\{\mathbf{X}_k\}) = \sum_{k,j} \mathbf{X}_k \mathbf{w}_j^{(1)} w_{jk}^{(2)}$	(15)	(18)	Section IV-B
Circular linear CNN	$f_{\theta,c}(\mathbf{X}) = \sum_j \mathbf{X}\mathbf{W}_j^{(1)} \mathbf{w}_j^{(2)}$	(19)	(20)	Section IV-C
FC vector output NN	$f_\theta(\mathbf{X}) = \phi\big(\mathbf{X}\mathbf{W}^{(1)}\big)\mathbf{W}^{(2)}$	(27)	(29)	Theorem VIII.1

- Proposed convex models reveal novel interpretations of neural network models through diverse convex regularization mechanisms. The regularizers range from group ℓ_p-norm to nuclear norm depending on the network architecture such as the connection structure and the number of outputs.
- Our derivations are extended to various neural network architectures including convolutional networks, piecewise linear activation functions, vector outputs, arbitrary convex losses, and ℓ_p-norm regularizers. We study vector output networks and derive exact convex programs for different regularizers.
- We extend the analysis to several practically relevant variants of the NN training problem. In particular, we examine networks with bias terms, ℓ_p-norm regularization of weights, and the interpolation regime.

I-C Notation

We use uppercase and lowercase bold letters to denote matrices and vectors, respectively, throughout the paper. We use subscripts to index entries (columns) of vectors (matrices). We use $\mathbf{I}_k$ for the identity matrix of size $k \times k$. We denote the set of integers from 1 to n as $[n]$. Moreover, $\|\cdot\|_F$ and $\|\cdot\|_*$ are Frobenius and nuclear norms and $\mathcal{B}_p := \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\|_p \leq 1\}$ is the unit ℓ_p ball. We also use $\mathbb{1}[x \geq 0]$ as an element-wise 0-1 valued indicator function. Furthermore, we use $\sigma_{max}(\cdot)$ to represent the maximum singular value of its argument. Finally, $\mathbf{D}(\cdot)$ (or $\mathbf{D}$) denotes a diagonal matrix. We use $\mathrm{Conv}(S)$ to denote the convex hull of a subset $S \subseteq \mathbb{R}^d$.

I-D Preliminaries

We consider a two-layer neural network architecture $f_\theta(\mathbf{X}) : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times C}$ with m hidden neurons and C outputs as follows

$$f_\theta(\mathbf{X}) := \phi(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}\,, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is a data matrix containing n training samples in $\mathbb{R}^d$, $\mathbf{W}^{(1)} \in \mathbb{R}^{d\times m}$ and $\mathbf{W}^{(2)} \in \mathbb{R}^{m\times C}$ are the hidden and output layer weights respectively. Here, $\phi(\cdot)$ is the non-linear activation function. We consider positive homogeneous activations of degree one, i.e., $\phi(tx) = t\phi(x)$, $\forall t \in \mathbb{R}_+$ such as ReLU, leaky ReLU, and absolute value. In addition, we denote all trainable parameters by $\theta := \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\}$ and the corresponding parameter space $\Theta := \{\theta : \mathbf{W}^{(1)} \in \mathbb{R}^{d\times m},\ \mathbf{W}^{(2)} \in \mathbb{R}^{m\times C}\}$.

Due to the nondifferentiability of the piecewise linear activations, we also review the definition of the Clarke subdifferential [19] of a given function f. Let $D \subset \mathbb{R}^d$ be the set of points at which f is differentiable. We assume that D has (Lebesgue) measure 1, meaning that f is differentiable *almost everywhere*. The Clarke subdifferential of f at $\mathbf{x}$ is then defined as

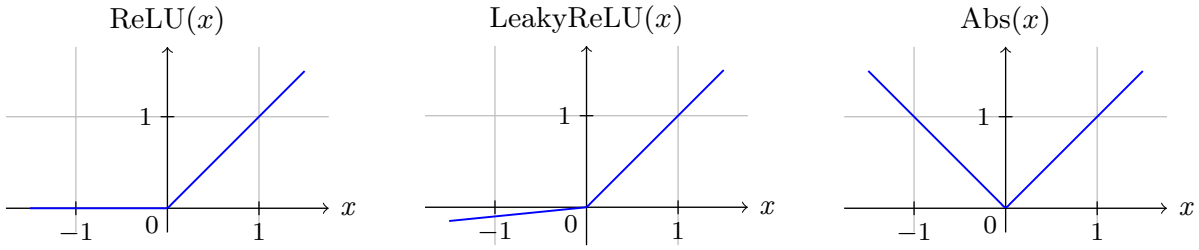
$$\partial_C f(\mathbf{x}) = \mathrm{Conv}\left\{ \lim_{k\to\infty} \nabla f(\mathbf{x}_k) \mid \lim_{k\to\infty} \mathbf{x}_k \to \mathbf{x},\ \mathbf{x}_k \in D \right\}.$$

Then, we say that $\mathbf{x} \in \mathbb{R}^d$ is Clarke stationary with respect to f if $\mathbf{0} \in \partial_C f(\mathbf{x})$.

Given a matrix of labels $\mathbf{Y} \in \mathbb{R}^{n\times C}$, the regularized training problem for the network in (1) is given by

$$\min_{\theta\in\Theta} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{Y}) + \beta\mathcal{R}(\theta)\,,$$

where $\mathcal{L}(\cdot,\cdot)$ is an arbitrary convex loss function, $\mathcal{R}(\cdot)$ is a regularization term, and $\beta > 0$ is the corresponding regularization parameter. We focus on the standard supervised regression/classification framework with conventional



(a) $\kappa = 0$ (b) $\kappa = 0.1$ (c) $\kappa = -1$

Fig. 1: Examples of piecewise linear activations ϕ satisfying the definition in (2).

squared ℓ_2-norm, i.e., weight decay, regularization denoted as $\mathcal{R}(\theta) = \frac{1}{2}(\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{W}^{(2)}\|_F^2)$. We consider the following family of piecewise linear activation functions

$$\phi(x) := \begin{cases} x & \text{if } x \geq 0 \\ \kappa x & \text{if } x < 0 \end{cases} \tag{2}$$

for some fixed scalar $\kappa < 0.5$. We note that the definition above includes a set of commonly used activation functions including ReLU, Leaky ReLU, and absolute value (see Figure 1). With these definitions, we have the following training problem

$$p^* := \min_{\theta \in \Theta} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{Y}) + \frac{\beta}{2}(\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{W}^{(2)}\|_F^2)\,. \tag{3}$$

Notice that the objective function above is highly non-convex due to the nested minimization of first and second layer weights and the composition of the nonlinearity ϕ with the loss function $\mathcal{L}$.

One of our key contributions is in developing an alternative parameterization of the same neural network and the corresponding training objective that enables significantly more efficient optimization.

To illustrate the challenges involved in optimizing the original formulation, we will examine the combinatorial nature of the original parameterization and why straightforward attempts to convexify the objective function fail. Consider rewriting the non-convex optimization problem (3) with ReLU activations and a scalar output, i.e., $\kappa = 0$ and $C = 1$, via enumerating all activation patterns of all ReLU neurons as

$$\min_{\substack{\theta \in \Theta \\ \mathbf{d}_j \in \mathcal{H}_{d_j}}} \mathcal{L}\left(\sum_{j=1}^{m} \left[\mathbf{d}_j \odot (\mathbf{X}\mathbf{w}_j^{(1)})\right] w_j^{(2)}, \mathbf{y}\right) + \frac{\beta}{2}(\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{w}^{(2)}\|_2^2)\,, \tag{4}$$

where $\odot$ denotes element-wise multiplication, and $\mathcal{H}_{d_j} := \{\mathbf{d}_j \in \{0,1\}^n : (2\mathbf{d}_j - \mathbf{1}) \odot (\mathbf{X}\mathbf{w}_j^{(1)}) \geq 0\}$ is a discrete parameterization of the piecewise parameterization of the set of ReLU activation patterns. A brute-force search would involve enumerating all possible combinations of m different length-n binary vectors, $\{\mathbf{d}_j\}_{j=1}^m$, which takes exponential time in the number of neurons m and the number of samples n. In fact, the best known algorithm for directly optimizing the training objective (3) with ReLU activations is a brute-force search over all possible piecewise linear regions of ReLU activations of m neurons and sign patterns for the output layer, which has complexity $\mathcal{O}(2^m n^{dm})$ (see Theorem 4.1 in [2]). In fact, known algorithms for approximately learning m hidden neuron ReLU networks have complexity $\mathcal{O}(2^{\sqrt{m}})$ (see Theorem 5 of [37]) due to similar combinatorial hardness with respect to the number of neurons. Since the number of hidden neurons in practical networks is typically in the order of hundreds or thousands, existing methods are computationally intractable even in small feature dimensions, e.g., $d = 2$.

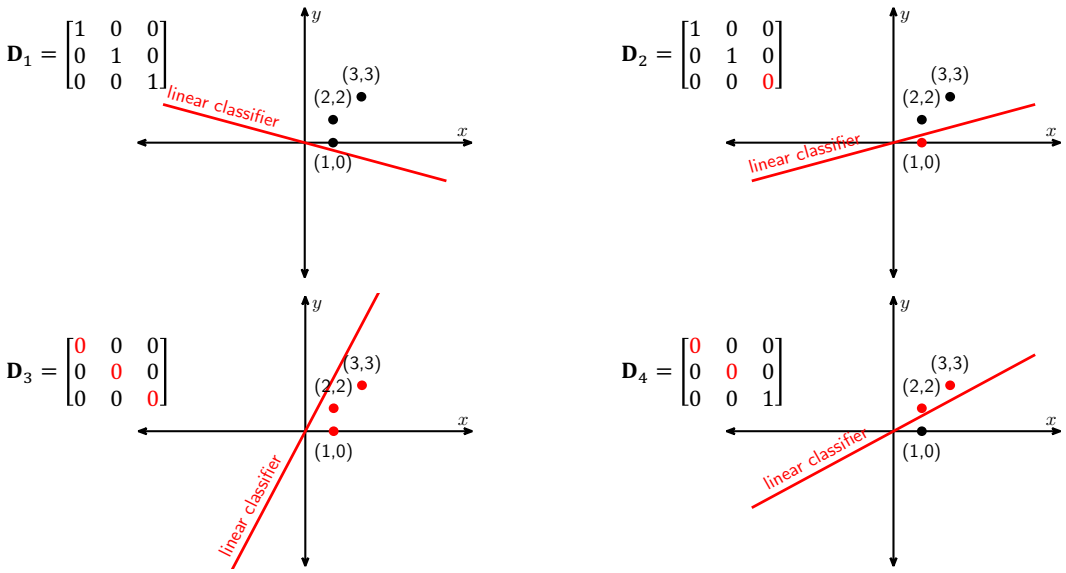


Fig. 2: Two dimensional illustration of all possible hyperplane arrangements that determine the diagonal matrices $\{\mathbf{D}_i\}_{i=1}^P$ for a toy dataset with dimensions $n = 3$, $d = 2$. In this example, we consider the ReLU activation, i.e., $\phi(x) = \max\{x, 0\}$. Note that the hyperplanes pass through the origin, as there is no bias term included in the neurons.

II AN EQUIVALENT CONVEX PROGRAM FOR TWO-LAYER NEURAL NETWORKS

We first introduce the notion of hyperplane arrangements of the data matrix $\mathbf{X}$ and then introduce an exact convex program as an alternative to the non-convex problem (3). Next, we note that piecewise linear activations can be equivalently represented via linear inequality constraints when their activation patterns are fixed since

$$\phi(\mathbf{X}\mathbf{w}^{(1)}) = \mathbf{D}\mathbf{X}\mathbf{w}^{(1)} \iff (2\mathbf{D} - \mathbf{I}_n)\mathbf{X}\mathbf{w}^{(1)} \geq 0,$$

where $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a fixed diagonal matrix of activation patterns defined as

$$\mathbf{D}_{ii} := \begin{cases} 1 & \text{if } \mathbf{x}_i^T \mathbf{w}^{(1)} \geq 0 \\ \kappa & \text{otherwise} \end{cases}. \tag{5}$$

It can be seen that each activation pattern corresponds to a hyperplane arrangement of the data matrix $\mathbf{X}$. We now enumerate all such distinct diagonal matrices that can be obtained for all possible $\mathbf{w}^{(1)} \in \mathbb{R}^d$, and denote them as $\mathbf{D}_1, ..., \mathbf{D}_P$ (see Figure 2 for the visualization of a two-dimensional case). Here, P denotes the number of regions in a partition of $\mathbb{R}^d$ by hyperplanes passing through the origin, and are perpendicular to the rows of $\mathbf{X}$. It is well known that

$$P \leq 2 \sum_{k=0}^{r-1} \binom{n-1}{k} \leq 2r \left(\frac{e(n-1)}{r} \right)^r \tag{6}$$

for $r \leq n$, where $r := \text{rank}(\mathbf{X})$ [22, 70] (see Appendix L). Thus, for a given data matrix of bounded rank, the number of hyperplane arrangements P is upper-bounded by an expression that is polynomial in both n and d. In Section IV, we show that convolutional networks used in practice have data rank bounded by a small constant that is equal to the spatial length of a filter.

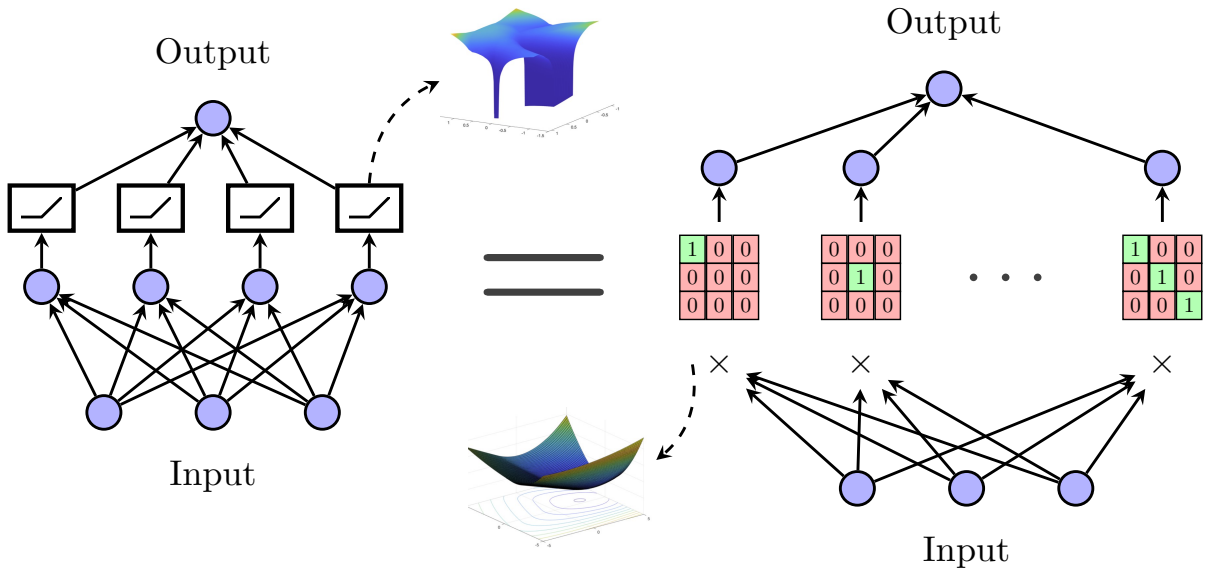


Fig. 3: An illustration of the equivalence between the non-convex ReLU network (**left**) and equivalent convex model (7) (**right**) along with their corresponding training losses.

A crucial observation is that the number P corresponds to the number of distinct activation patterns generated by **only a single neuron** on the training data. This number is polynomial in n when the rank of the data matrix is bounded by a constant. On the other hand, optimizing the non-convex formulation (3) or (4) requires searching over all activation patterns of m neurons jointly, which results in **computational complexity exponential in** m as discussed in Section I-D.

With this observation, we next introduce an exact polynomial-time solvable convex program that solves (3) optimally by using dP variables.

Theorem II.1. *Given a scalar output network, i.e., $C = 1$, consider the convex program*

$$p_{\text{cvx}} = \min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \mathcal{L}(\mathcal{A}(\mathbf{X})\mathbf{w}, \mathbf{y}) + \beta \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2 \tag{7}$$

where $\mathbf{w} := [\mathbf{w}_1^T, \ldots, \mathbf{w}_{2P}^T]^T \in \mathbb{R}^{2dP}$, and let $\mathbf{w}^*$ be the minimum-norm optimal solution. We have $p^* = p_{\text{cvx}}$ using the equivalent formulation in Lemma II.1 when $m \geq m^* := \sum_{i=1}^{2P} \mathbb{1}[\mathbf{w}_i^* \neq 0]$, i.e., when the number of neurons exceeds the critical threshold m^*. Here, $\mathcal{A}(\mathbf{X}) \in \mathbb{R}^{n \times 2dP}$ and the constraint set of the convex program denoted as $\mathcal{C}(\mathbf{X})$ are defined as

$$\mathcal{C}(\mathbf{X}) := \left\{ \mathbf{w} \in \mathbb{R}^{2dP} \ : \ (2\mathbf{D}_i - \mathbf{I}_n)\mathbf{X} \begin{bmatrix} \mathbf{w}_i & \mathbf{w}_{i+P} \end{bmatrix} \geq 0, \forall i \in [P] \right\}$$
$$\mathcal{A}(\mathbf{X}) := \begin{bmatrix} \mathbf{D}_1\mathbf{X} & \ldots & \mathbf{D}_P\mathbf{X} & -\mathbf{D}_1\mathbf{X} & \ldots & -\mathbf{D}_P\mathbf{X} \end{bmatrix}.$$

Theorem II.1 shows that a standard two-layer neural network with piecewise linear activations can be described as a convex mixture of locally linear models $\{\mathbf{D}_i\mathbf{X}\mathbf{w}_i\}_{i=1}^P$ and $\{-\mathbf{D}_i\mathbf{X}\mathbf{w}_{i+P}\}_{i=1}^P$, where the fixed diagonal matrices $\{\mathbf{D}_i\}_{i=1}^P$ control the data samples interacting with the local model as fixed gates. Therefore, optimal two-layer networks can be viewed as sparse convex mixtures of locally linear functions, where sparsity is enforced via the group Lasso regularization. We also note that there is a mapping between the optimal solutions of the non-convex and convex training objectives, which is presented in Proposition II.1.

Next, we prove that Clarke stationary points correspond to the global optimum of a subsampled convex program studied in the previous section. This result explains the neural network models found by first order optimization methods such as (Stochastic) Gradient Descent, which converge to a neighborhood of a stationary point.

Theorem II.2. *Suppose that θ is a Clarke stationary point of the nonconvex training objective in (3). Then, θ corresponds to a global optimum of the convex program where a subset of variables are set to zero, resulting in a subsampled form (see Section III-A for the definition of the subsampled convex program) in (7) with $\tilde{P} = m$ arrangement patterns.*

Theorem II.2 implies that any local minimum of the nonconvex training objective in (3) can be characterized as a global minimum of a subsampled form of the convex program in (7), for which the sampling procedure is in Section III-A. Therefore, we can characterize all stationary points of the nonconvex training objective in (3) by sampling the arrangement patterns for the convex optimization problem in (7).

Importantly, the proposed convex program trains two-layer neural networks *optimally*. In contrast, local search heuristics such as backpropagation may converge to suboptimal solutions, which is illustrated with numerical examples in Section IX. To the best of our knowledge, our results provide the first polynomial-time algorithm to train optimal neural networks when the data rank (or feature dimension) is fixed.

In the light of the results above, a weight decay, i.e., squared ℓ_2 norm, regularized two-layer neural network with piecewise linear activations is a high-dimensional feature selection method that seeks sparsity. More specifically, training the non-convex model can be considered as transforming the data to the higher dimensional feature matrix $\mathcal{A}(\mathbf{X})$, and then seeking a parsimonious convex model through the group Lasso regularization. The optimal model is very concise due to the group sparsity induced by the sum of Euclidean norms. This fact, however, is not obvious from the non-convex formulations of these neural network models.

The following result shows that one can construct a classical two-layer network as in (1) from the solution of the convex program (7).

Proposition II.1. *An optimal solution to the non-convex problem in (3), i.e., denoted as $\{\mathbf{w}_j^{(1)^*}, w_j^{(2)^*}\}_{j=1}^{m^*}$, can be constructed from the optimal solution to the convex program as follows*

$$(\mathbf{w}_{j_i}^{(1)^*}, w_{j_i}^{(2)^*}) = \begin{cases} \left(\frac{\mathbf{w}_i^*}{\sqrt{\|\mathbf{w}_i^*\|_2}}, \sqrt{\|\mathbf{w}_i^*\|_2} \right) & \text{if } \mathbf{w}_i^* \neq 0 \text{ and } i \leq P \\ \left(\frac{\mathbf{w}_i^*}{\sqrt{\|\mathbf{w}_i^*\|_2}}, -\sqrt{\|\mathbf{w}_i^*\|_2} \right) & \text{if } \mathbf{w}_i^* \neq 0 \text{ and } i > P \end{cases},$$

where $\{\mathbf{w}_i^*\}_{i=1}^{2P}$ are the optimal solutions to (7), and $j_i \in [|\mathcal{J}|]$ given the definitions $\mathcal{J} := \{i : \|\mathbf{w}_i\| > 0\}$.

Remark II.1. *Theorem II.1 shows that two-layer networks with m hidden neurons and the activation ϕ can be globally optimized via the second order cone program (7) with $2dP$ variables and $2nP$ linear inequalities where $P \leq 2r\left(\frac{e(n-1)}{r}\right)^r$, and $r = \text{rank}(\mathbf{X})$. The computational complexity is at most $\mathcal{O}\left(d^3 r^3 \left(\frac{n}{r}\right)^{3r}\right)$ using standard interior-point solvers. For fixed rank r (or dimension d), the complexity is polynomial in n and m, which is an exponential improvement over the state of the art [2, 10]. However, for fixed n and $\text{rank}(\mathbf{X}) = d$, the complexity is exponential in d, which can not be improved unless $\mathrm{P} = \mathrm{NP}$ even for $m = 2$ [11]. Note that the convex program and the non-convex problem differ in terms of the hardness of the optimization problem they present. While the non-convex problem has fewer decision variables and can be locally optimized with a computational complexity of $\mathcal{O}(k(md + mC))$ via SGD, where k is the number of iterations, it does not have a known systematic method for solving it (apart from the one presented in this work) or verifying the optimality of a given solution. In contrast, the convex program has a larger number of decision variables, but it can be solved to global optimality and the optimality of any candidate solution can be checked. Thus, the convex program provides a trade-off between the difficulties of high-dimensionality and non-convexity.*

Remark II.2. *Popular non-convex heuristics such as gradient descent and variants applied to the non-convex problem (3) can be viewed as local active set solvers for the convex program (7). In this active set strategy, only a small subset of variables are maintained in the current solution, which corresponds to a small subset of hyperplane arrangements, i.e., column blocks of $\hat{\mathbf{X}}$. The variables in the active set solver enter and exit the active set as the ReLU activation patterns change.*

The proofs of the theorems and other claims (including Theorem II.1) can be found in the Supplementary Material.

To gain a better understanding of the convex program (7) and the resulting convex ReLU neural network model, we will next consider a toy example. This example will provide insight into the underlying mechanisms and allow for a more intuitive interpretation of the equivalent non-convex neural network model.

Example II.1. *Let us consider ReLU activations and the training data matrix*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 0 \end{bmatrix}.$$

Even though there exist $2^3 = 8$ *distinct binary sequences of length* 3*, the number of hyperplane arrangements in this case is* 4 *as shown in Figure 2. Considering an arbitrary label vector* $\mathbf{y} \in \mathbb{R}^3$ *and the squared loss, we formulate the convex program in (7) as follows*

$$\begin{aligned} &\min_{\{\mathbf{w}_i\}_{i=1}^6} \frac{1}{2} \left\| f_{\mathbf{w}}(\mathbf{X}) - \mathbf{y} \right\|_2^2 + \beta \sum_{i=1}^6 \left\| \mathbf{w}_i \right\|_2 \\ &\text{s.t. } \mathbf{x}_i^T [\mathbf{w}_1 \ \mathbf{w}_4] \geq 0, \ i = 1, 2, 3 \\ &\mathbf{x}_i^T [\mathbf{w}_2 \ \mathbf{w}_5] \geq 0, \ i = 1, 2, \quad \mathbf{x}_3^T [\mathbf{w}_2 \ \mathbf{w}_5] \leq 0 \\ &\mathbf{x}_i^T [\mathbf{w}_3 \ \mathbf{w}_6] \leq 0, \ i = 1, 2, \quad \mathbf{x}_3^T [\mathbf{w}_3 \ \mathbf{w}_6] \geq 0, \end{aligned}$$

where

$$\begin{aligned} f_{\mathbf{w}}(\mathbf{X}) &:= \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{x}_3^T \end{bmatrix} (\mathbf{w}_1 - \mathbf{w}_4) + \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathbf{0}^T \end{bmatrix} (\mathbf{w}_2 - \mathbf{w}_5) + \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \mathbf{x}_3^T \end{bmatrix} (\mathbf{w}_3 - \mathbf{w}_6) \\ &= \mathbf{D}_1 \mathbf{X} (\mathbf{w}_1 - \mathbf{w}_4) + \mathbf{D}_2 \mathbf{X} (\mathbf{w}_2 - \mathbf{w}_5) + \mathbf{D}_3 \mathbf{X} (\mathbf{w}_3 - \mathbf{w}_6), \end{aligned}$$

provided that

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \mathbf{D}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Interestingly, we obtain a convex programming description of the neural network model that is interpretable: we are looking for a group sparse model to explain the response $\mathbf{y}$ via a convex mixture of linear models. To give an example, the linear term $\mathbf{w}_2 - \mathbf{w}_5$ is responsible for predicting on the subset $\{\mathbf{x}_1, \mathbf{x}_2\}$ of the dataset, and the linear term $\mathbf{w}_3 - \mathbf{w}_6$ is responsible for predicting on the subset $\{\mathbf{x}_3\}$ of the dataset, etc. Due to the regularization term $\sum_{i=1}^6 \left\| \mathbf{w}_i \right\|_2$, only a few of these linear terms will be non-zero at the optimum, which shows a strong bias towards simple solutions among all piecewise linear models. Here, we may ignore the arrangement that corresponds to all-zeros since it does not contribute to the objective. It is important to note that the above objective is equivalent to the non-convex neural network training problem. Although the non-convex training process given in (3) is hard to interpret, the equivalent convex optimization formulation shows the structure of the optimal neural network through a fully transparent convex model.

Remark II.3. *In general, we expect the number of hyperplane arrangements* P *to be small when the data matrix is of small rank as in Figure 2. In Section III, we show that a similar result applies to near low-rank matrices which are frequently encountered in practice: a relatively small number of arrangement patterns is sufficient to approximate the global optimum up to a small relative error.*

II-A Networks with a bias term in hidden neurons

We now modify the neural network architecture in (1) as

$$f_\theta(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{1}\mathbf{b}^T)\mathbf{W}^{(2)},$$

where $\mathbf{b} \in \mathbb{R}^m$ denotes the trainable bias vector. For this architecture, the training problem is the same with (3) except $\theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}\}$.

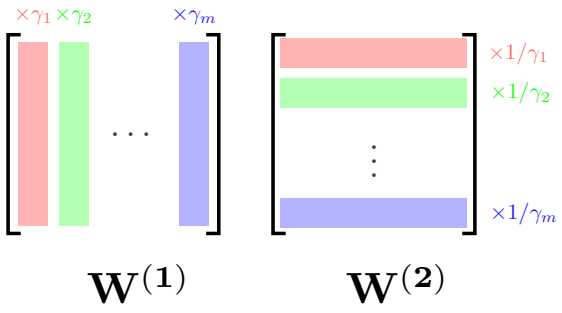


Fig. 4: An illustration of the scaling technique in Lemma II.1. This is instrumental in obtaining the ℓ_1-norm penalized problem (8), which leads to a strong dual formulation.

Corollary II.1. *As a result of Theorem II.1, the non-convex training problem with bias term can be cast as a finite dimensional convex program as follows*

$$\min_{\theta_c \in \mathcal{C}(\mathbf{X})} \mathcal{L}(f_{\theta_c}(\mathbf{X}), \mathbf{y}) + \beta \sum_{i=1}^{2P} \left\| [\mathbf{w}_i; b_i] \right\|_2,$$

where $\theta_c := \{\mathbf{w}, \mathbf{b}\}$. *Moreover,* $f_{\theta_c}(\mathbf{X})$ *and* $\mathcal{C}(\mathbf{X})$ *are defined as*

$$f_{\theta_c}(\mathbf{X}) = \sum_{i=1}^{P} \mathbf{D}_i((\mathbf{X}\mathbf{w}_i + \mathbf{1}b_i) - (\mathbf{X}\mathbf{w}_{i+P} + \mathbf{1}b_{i+P}))$$
$$\mathcal{C}(\mathbf{X}) := \left\{ \mathbf{w} \in \mathbb{R}^{2dP}, \mathbf{b} \in \mathbb{R}^{2P} : (2\mathbf{D}_i - \mathbf{I}_n) \left(\mathbf{X} \begin{bmatrix} \mathbf{w}_i & \mathbf{w}_{i+P} \end{bmatrix} + \mathbf{1} \begin{bmatrix} b_i & b_{i+P} \end{bmatrix} \right) \geq 0, \forall i \in [P] \right\}.$$

Remark II.4. *We note that including bias may improve the expressive power of the neural network (1) in small feature dimensions. This operation corresponds to augmenting a column of all-ones to the data matrix, which implies a slight increase in the number of hyperplane arrangements. The rank of the data matrix increases from r to at most $r+1$, therefore, the new upperbound on the number of arrangements (6) is obtained by simply replacing r with $r+1$. As an example, in Figure 2, $(\mathbf{w}^{(1)}, b) = ([1;1], -5)$ can separate $\mathbf{x}_1$ and $\mathbf{x}_2$, therefore we obtain an additional hyperplane arrangement and a corresponding variable vector in the convex program.*

II-B Convex duality of two-layer neural networks

In this section, we provide a high-level overview of the mathematical proof technique that we use to obtain the convex program. It is worth noting that these results are of independent interest, as they are not solely motivated by global optimization of neural networks. We start with convex duality for (3), which is essential for deriving the convex program in Theorem II.1.

Since the piecewise linear activation ϕ is a positive homogeneous function of degree one, we can apply a rescaling (see Figure 4) to equivalently state the problem in (3) as an ℓ_1-norm minimization problem.

Lemma II.1. *The problem in (3) can be equivalently formulated as the following ℓ_1-norm minimization problem*

$$p^* := \min_{\theta \in \Theta_s} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \beta \|\mathbf{w}^{(2)}\|_1 \,, \tag{8}$$

where $\Theta_s := \{\theta \in \Theta : \|\mathbf{w}_j^{(1)}\|_2 \leq 1, \forall j \in [m]\}$.

We note that is important to obtain an ℓ_1 regularized form of the non-convex problem in order to obtain strong duality. It can be easily verified that a straightforward application of Lagrange duality applied to the original weight-decay regularized objective does not lead to a strong dual.

Algorithm 1 `Polynomial-time convex neural network training algorithm`

1: Set the desired rank k based on the bound in (11)
2: Compute the rank-k approximation of the data matrix: $\hat{\mathbf{X}}_k$
3: Set the number of arrangements to be sampled via Theorem III.2: $\tilde{P}$
4: Sample hyperplane arrangements from $\hat{\mathbf{X}}_k$: $\{\mathbf{D}_i^k\}_{i=1}^{\tilde{P}}$
5: Solve the convex training problem in (7) using the original data $\mathbf{X}$ and rank-k arrangements $\{\mathbf{D}_i^k\}_{i=1}^{\tilde{P}}$

We now use Lemma II.1 to obtain the convex dual form of (3). We first take the dual of (8) with respect to $\mathbf{w}^{(2)}$ and then change the order of min-max to obtain the following dual problem

$$p^* \geq d^* := \max_{\mathbf{v}} \min_{\mathbf{w}_j^{(1)} \in \Theta_s} -\mathcal{L}^*(\mathbf{v}) \tag{9}$$
$$\text{s.t. } \left|\mathbf{v}^T \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big)\right| \leq \beta,\ \forall j \in [m],$$

where $\mathcal{L}^*$ is the Fenchel conjugate function defined as [13]

$$\mathcal{L}^*(\mathbf{v}) := \max_{\mathbf{z}}\ \mathbf{z}^T\mathbf{v} - \mathcal{L}(\mathbf{z}, \mathbf{y})\,.$$

Since $\min_x \max_y f(x,y) \geq \max_y \min_x f(x,y)$, (9) is a lower bound for (8). However, at this point it is not clear whether the lower-bound is tight, i.e., $p^* = d^*$.

Using the dual characterization in (9), we first find a set of hidden layer weights via the optimality conditions and active constraints of (9). We then prove strong duality, i.e., $p^* = d^*$, to verify the optimality of the hidden layer weight found via the dual problem. A complete proof of this result can be found in Appendix C.

III Scalable optimization of the neural network convex program

Notice that the worst-case computational complexity to solve the convex program in Theorem II.1 is exponential in the feature dimension d for full-rank training data as detailed in Remark II.1. Therefore, globally optimizing the training objective (7) may not be feasible for large d.

To avoid the complexity of enumerating exponentially many hyperplane arrangements and to effectively scale to high-dimensional datasets, we consider a low-rank approximation of the data to approximate the arrangements and subsequently obtain an approximation of (3). We denote the rank-k approximation of $\mathbf{X}$ as $\hat{\mathbf{X}}_k$ such that $\|\mathbf{X} - \hat{\mathbf{X}}_k\|_2 \leq \sigma_{k+1}$, where σ_{k+1} is the $(k+1)^{th}$ largest singular value of $\mathbf{X}$. Then, we have the following result.

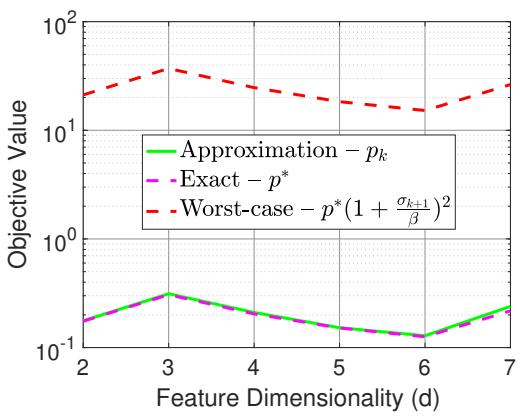
Theorem III.1. *Consider the following variant of the convex program (7) with rank-k approximated hyperplane arrangements*

$$\mathbf{w}^{(k)} \in \operatorname*{argmin}_{\mathbf{w} \in \mathcal{C}(\hat{\mathbf{X}}_k)} \mathcal{L}\left(\sum_{i=1}^{\hat{P}} \mathbf{D}_i^k \mathbf{X}(\mathbf{w}_i - \mathbf{w}_{i+\hat{P}}), \mathbf{y}\right) + \beta \sum_{i=1}^{2\hat{P}} \|\mathbf{w}_i\|_2, \tag{10}$$

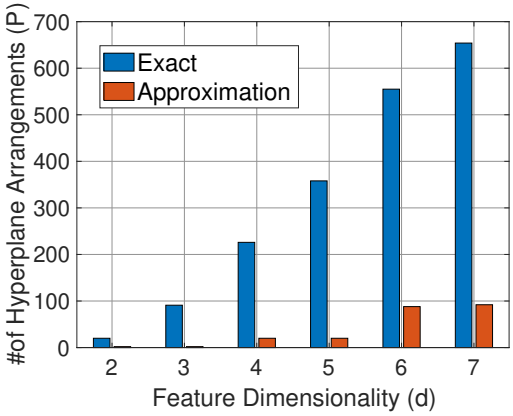
where $\{\mathbf{D}_i^k\}_{i=1}^{\hat{P}}$ denotes the set of hyperplane arrangements generated by the rank-k approximation $\hat{\mathbf{X}}_k$. Let us define $p_{\text{cvx}-k}$ as the value of the non-convex objective (3) evaluated at any minimizer $\mathbf{w}^{(k)}$ defined above. Then, given an L-Lipschitz convex loss $\mathcal{L}(\cdot, \mathbf{y})$ and an R-Lipschitz activation function $\phi(\cdot)$, we have the following approximation guarantee

$$p^* \leq p_{\text{cvx}-k} \leq p^* \left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2. \tag{11}$$

Remark III.1. *Theorem II.1 and Theorem III.1 imply that for a given rank-r data matrix $\mathbf{X}$, the regularized training problem in (3) can be approximately solved via convex optimization solvers to achieve an approximation with objective value $p^* \left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2$ in $\mathcal{O}\Big(d^3 k^3 \big(\frac{n}{k}\big)^{3k}\Big)$ time complexity, where p^* is the optimal value and $k \leq r$. Therefore, even for full rank data matrices for which the worst-case complexity of solving (7) is exponential in d, this method approximately solves the convex program in (7) in polynomial-time with strong guarantees.*



(a) Objective value



(b) Number of hyperplane arrangements (P)

Fig. 5: Verification of the approximation guarantees in Theorem III.1. Here, we train a two-layer ReLU network using the convex program in Theorem II.1 with ℓ_2 loss on a synthetic dataset with $n = 10$, $\beta = 0.1$, and the low-rank approximation $k = \lfloor \frac{d}{2} \rfloor$. To obtain a rank-deficient model, we first generate a random data matrix using a multivariate Gaussian distribution with $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}_d$ and then explicitly set $\sigma_{k+1} = \ldots = \sigma_d = 1$.

As an illustration of Theorem III.1, consider a ReLU network training problem with ℓ_2 loss. The approximation ratio becomes $(1+\frac{\sigma_{k+1}}{\beta})^2$, which is typically close to 1 due to fast decaying singular values of training data matrices encountered in practice. In Figure 5, we present a numerical example on i.i.d. Gaussian synthetic data matrices and the low-rank approximation strategy. Figure 5a shows that[1] the low-rank approximation of the objective p_k is closer to p^* than the worst-case upper-bound predicted by Theorem III.1. However, in Figure 5b, we observe that the low-rank approximation provides a significant reduction in the number of hyperplane arrangements, and therefore in the complexity of solving the convex program.

III-A Efficient sampling of hyperplane arrangements with guarantees

The convex program in (10) can be globally optimized with a polynomial-time complexity, however, it is not obvious how to generate the full set of hyperplane arrangement matrices $\{\mathbf{D}_i\}_{i=1}^P$ in practice. Although there exist algorithms to construct these arrangements, e.g., [26], they can become computationally challenging in high dimensions.In this section, we first show how to efficiently sample these hyperplane arrangements for the convex programs (7) and (10), and then provide probabilistic approximation guarantees for the subsampled convex program where only a subset of arrangements are used.

The convex program (7) can be approximated by sampling a set of diagonal matrices $\{\mathbf{D}_i\}_{i=1}^{\tilde{P}}$. We refer to this lower dimensional problem as the *subsampled convex program*. For example, we can generate vectors from the standard multivariate Gaussian, or some other distribution, as $\mathbf{w}^{(1)} \sim N(\mathbf{0}, \mathbf{I}_d)$ i.i.d. $\tilde{P}$ times, and then construct diagonal matrices via (5) to solve the reduced convex problem. This is essentially a type of random coordinate descent strategy applied to the convex objective (7).

We next show that hyperplane arrangement matrices $\{\mathbf{D}_i\}_{i=1}^P$ have a one-to-one correspondence to the vertices of a zonotope whose generators are the training data samples. We define the data zonotope $\mathcal{Z}(\mathbf{X})$, which is a low-dimensional linear projection of a hypercube as follows.

$$\mathcal{Z}(\mathbf{X}) := \{\mathbf{X}^T\mathbf{u} : \mathbf{u} \in [0,1]^n\} = \text{Conv}\left\{\sum_{i=1}^n \mathbf{x}_i u_i \ : u_i \in \{0,1\},\ \forall i \in [n]\right\},$$

[1]We provide the details of this experiment in A.

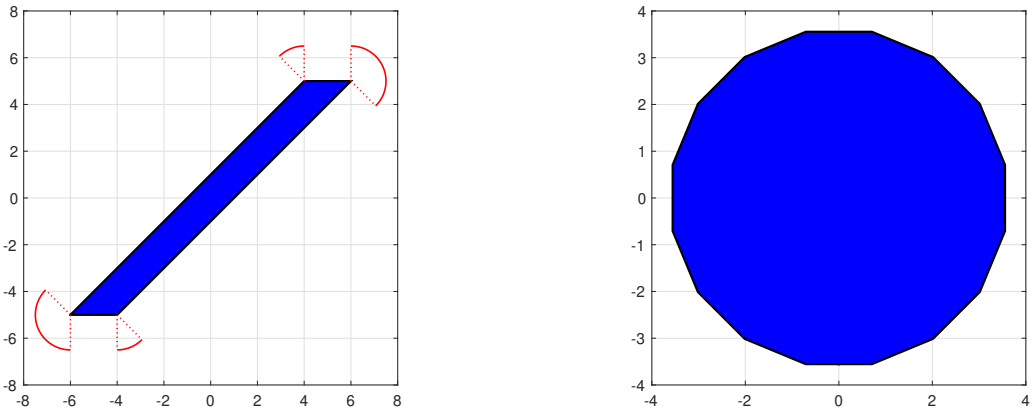


Fig. 6: **Left:** An illustration of the zonotope $\mathcal{Z}(\mathbf{X})$ for the data in Example II.1 and vertex solid angles θ_i, where $\mathbf{X} = [2\ 2; 3\ 3; 1\ 0]$. Note that here the minimum angle is $\theta_{min} = \pi/4$. **Right:** An illustration of a zonotope $\mathcal{Z}(\mathbf{X})$ generated uniformly on the unit sphere $\mathbf{X} \in \mathbb{R}^{16\times 2}$. It can be verified that all the angles are equal, i.e., $\theta_i = \frac{2\pi}{n} = \frac{\pi}{8}\ \forall i$.

where Conv denotes the convex hull operation. Then, we observe that the extreme points of the zonotope defined above are linked to the hyperplane arrangements that appear in our convex program (7) since

$$\mathbf{u}^* := \underset{\mathbf{u}\in[0,1]^n}{\operatorname{argmax}}\, \mathbf{v}^T\mathbf{X}^T\mathbf{u} \implies u_i^* = \mathbb{1}[\mathbf{x}_i^T\mathbf{v} \geq 0],\ \forall i\ :\ \mathbf{x}_i^T\mathbf{v} \neq 0, \tag{12}$$

and we may pick $u_i^* \in [0,1]$ whenever $\mathbf{x}_i^T\mathbf{v} = 0$ for any $i \in [n]$. Therefore, for every direction $\mathbf{v} \in \mathbb{R}^d$, there is an extreme point of $\mathcal{Z}(\mathbf{X})$ of the form $\mathbf{e} = \mathbf{X}^T\mathbb{1}[\mathbf{X}\mathbf{v} \geq 0] = \sum \mathbf{x}_i \mathbb{1}[\mathbf{x}_i^T\mathbf{v} \geq 0]$. In particular, an extreme point $\mathbf{e}$ is optimal when $-\mathbf{v}$ is in the normal cone of the zonotope at $\mathbf{e}$, i.e., $\mathbf{v} \in N_{\mathcal{Z}(\mathbf{X})}(\mathbf{e}) := \left\{\mathbf{w} : \mathbf{w}^T(\mathbf{x}-\mathbf{e}) \leq 0\ \forall \mathbf{x} \in \mathcal{Z}(\mathbf{X})\right\}$ due to convex optimality conditions for the problem (12). An alternative representation of the normal cone is given by $N_{\mathcal{Z}(\mathbf{X})}(\mathbf{e}) := \left\{\mathbf{w} : \operatorname{sign}(\mathbf{X}\mathbf{w}) = \operatorname{sign}(\mathbf{X}\mathbf{v})\right\}$, where $\mathbf{e} = \mathbf{X}^T\mathbb{1}[\mathbf{X}\mathbf{v} \geq 0]$. The number of extreme points of $\mathcal{Z}(\mathbf{X})$ is equal to the number of hyperplane arrangements of $\mathbf{X}$, which we denote by P. We refer the reader to [40] for further details on zonotopes and hyperplane arrangements.

Next, we will now prove that the randomly sampling hyperplane arraignments approach in fact solves the exact convex program in (7) with high probability provided that $\tilde{P}$ exceeds a certain threshold. We define the solid angle of a convex cone $\mathcal{C}(\mathbf{X})$ (see e.g., [7]) as the probability that a randomly drawn standard multivariate Gaussian lies inside $\mathcal{C}(\mathbf{X})$, i.e.,

$$\Omega(\mathcal{C}(\mathbf{X})) := \mathbb{P}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{w})}\left[\mathbf{w} \in \mathcal{C}(\mathbf{X})\right] = \frac{1}{(2\pi)^{d/2}} \int_{\mathcal{C}(\mathbf{X})} e^{-\frac{1}{2}\|\mathbf{x}\|_2^2} d\mathbf{x}\,.$$

Suppose that the normal cones $N_{\mathcal{Z}(\mathbf{X})}(\mathbf{e})$ of $\mathcal{Z}(\mathbf{X})$ at its extreme points have solid angles given by $\{\theta_i\}_{i=1}^P$. More precisely, we define

$$\theta_i := \Omega(N_{\mathcal{Z}(\mathbf{X})}(\mathbf{e}_i)),\ \forall i \in [P]\,,$$

where $\mathbf{e}_i = \mathbf{X}^T\mathbb{1}[\mathbf{X}\mathbf{v}_i \geq 0],\ \forall i \in [P]$ are P distinct extreme points generated by directions $\{\mathbf{v}_i\}_{i=1}^P$. Note that all extreme points have strictly positive solid angle since otherwise those points may be removed from the set of extreme points while maintaining the same convex hull. We also include a two-dimensional illustration of the zonotope $\mathcal{Z}(\mathbf{X})$ for the data in Example II.1 and the corresponding solid angles of normal cones $\{\theta_i\}_{i=1}^P$ at extreme points in Figure 6.

The next result shows that all arrangement patterns will be sampled under the assumption that the minimum solid angle of the data zonotope is bounded by a positive constant.

Theorem III.2. *Let P be the number of hyperplane arrangements for the training data matrix $\mathbf{X}$. Then, in order to sample all P arrangements with probability $1-\epsilon$, it is sufficient to let the number of random samples $\tilde{P}$ satisfy $\tilde{P} \geq \bar{\theta}^{-1} P \log(P/\epsilon)$, where $\bar{\theta} := P \min_{i \in [P]:\theta_i > 0} \theta_i$ is the minimum solid angle of the normal cones of the zonotope $\mathcal{Z}(\mathbf{X})$ multiplied by the number of vertices P.*

Remark III.2. *We note that the multiplicative factor P is introduced in the definition of $\bar{\theta}$ in order to remove its inverse dependence on the number of vertices P. To give concrete examples, the zonotope $\mathcal{Z}(\mathbf{X})$ seen in the right panel of Figure 6 is a regular n-gon which has $\theta_i = \frac{2\pi}{n}\, \forall i \in [n]$ and therefore $\bar{\theta} = n\frac{2\pi}{n} = 2\pi$ since $P = n$. Similarly, the zonotope $\mathcal{Z}(\mathbf{X})$ in the left panel of Figure 6 has $\bar{\theta} = 4\frac{\pi}{4} = \pi$ since $P = 4$.*

Along with the low-rank approximation in Theorem III.1 reducing the number of arrangements to $P = \mathcal{O}((n/k)^k)$ for any target rank k, the efficient sampling approach in Theorem III.2 proves that we can solve the convex program in (7) with a polynomial-time complexity in all problem parameters. The pseudocode for this training approach is presented in Algorithm 1.

IV Convolutional neural networks

Here, we introduce extensions of our approach to CNNs. Two-layer convolutional networks with m hidden neurons (filters) of dimension d and fully connected output layer weights can be described by patch matrices $\mathbf{X}_k \in \mathbb{R}^{n \times d}$, $k = 1, ..., K$. With this notation, $\mathbf{X}_k \mathbf{w}_j^{(1)}$ represents the k^{th} spatial dimension of the convolution with the filter $\mathbf{w}_j^{(1)}$ across the dataset.

IV-A Standard convolutional networks

We first analyze convolutional neural networks with global average pooling, which is a commonly used technique for reducing the dimensionality of the feature maps in a convolutional neural network. Global average pooling involves taking the average of all the values in each spatial feature map. Using the notation above, the output of a CNN with global average pooling is given by

$$\frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \phi(\mathbf{X}_k \mathbf{w}_j^{(1)}) w_j^{(2)}.$$

For this architecture, we consider the following training problem

$$\min_{\theta \in \Theta} \mathcal{L}\left(\frac{1}{K} \sum_{k=1}^{K} f_\theta(\mathbf{X}_k), \mathbf{y} \right) + \frac{\beta}{2} (\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{w}^{(2)}\|_2^2)\,, \tag{13}$$

where $f_\theta(\mathbf{X}_k) = \sum_{j=1}^{m} \phi(\mathbf{X}_k \mathbf{w}_j^{(1)}) w_j^{(2)}$. We first define an augmented data matrix by concatenating the patch matrices as $\hat{\mathbf{X}} := \begin{bmatrix} \mathbf{X}_1^T & \mathbf{X}_2^T & \dots & \mathbf{X}_K^T \end{bmatrix}^T \in \mathbb{R}^{nK \times d}$. Then, (13) can be equivalently written as

$$\min_{\theta \in \Theta} \tilde{\mathcal{L}}(f_\theta(\hat{\mathbf{X}}), \mathbf{y}) + \frac{\beta}{2} (\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{w}^{(2)}\|_2^2)\,, \tag{14}$$

where we define the loss function as

$$\tilde{\mathcal{L}}(f_\theta(\hat{\mathbf{X}}), \mathbf{y}) := \mathcal{L}\left(\frac{1}{K} \sum_{k=1}^{K} f_\theta(\mathbf{X}_k), \mathbf{y} \right).$$

This shows that the convolutional network training problem with global average pooling in (13) can be cast as a standard fully connected network training problem as in (14) using the training data $\hat{\mathbf{X}}$ and modified convex loss function $\tilde{\mathcal{L}}$. Therefore, the convex program (7) solves the above problem exactly in $\mathcal{O}\left(d^3 r^3 \left(\frac{nK}{r}\right)^{3r}\right)$ complexity, where d is the number of variables in a single filter and r is the rank of $\hat{\mathbf{X}}$. It holds that $r \leq d$ since $\hat{\mathbf{X}} \in \mathbb{R}^{nK \times d}$. Note that typical CNNs employ m filters of constant size, e.g., $3 \times 3 \times m$ (d=9) in the first layer [45]. As a result of this small feature dimension (or filter size), our result implies that globally optimizing a CNN architecture can be done in a polynomial-time, i.e., polynomial in all dimensions when the filter size d is a constant.

IV-B Linear convolutional network training as a Semi-Definite Program (SDP)

We now analyze CNNs linear activations $\phi(x) = x$ trained via the following optimization problem

$$\min_{\theta \in \Theta} \mathcal{L}(f_{\theta,c}(\{\mathbf{X}_k\}_{k=1}^K), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^{m} (\|\mathbf{w}_j^{(1)}\|_2^2 + \|\mathbf{w}_j^{(2)}\|_2^2), \tag{15}$$

where

$$f_{\theta,c}(\{\mathbf{X}_k\}_{k=1}^K) = \sum_{k=1}^{K} \sum_{j=1}^{m} \mathbf{X}_k \mathbf{w}_j^{(1)} w_{jk}^{(2)}.$$

The corresponding dual problem is given by

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \sqrt{\sum_{k=1}^{K} \left(\mathbf{v}^T \mathbf{X}_k \mathbf{w}^{(1)}\right)^2} \leq \beta. \tag{16}$$

By similar arguments to those used in the proof of Theorem II.1, strong duality holds. Furthermore, the maximizers of the constraint are the maximal eigenvectors of $\sum_k \mathbf{X}_k^T \mathbf{v}\mathbf{v}^T \mathbf{X}_k$, which are optimal neurons (filters). Thus, we can express (16) as the following SDP

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \sigma_{\max}\left([\mathbf{X}_1^T \mathbf{v} \ldots \mathbf{X}_K^T \mathbf{v}]\right) \leq \beta. \tag{17}$$

The dual of the above SDP is a nuclear norm penalized convex optimization problem (see M-B)

$$\min_{\mathbf{z}_k \in \mathbb{R}^d} \mathcal{L}(\hat{f}_{\theta,c}(\{\mathbf{X}_k\}_{k=1}^K), \mathbf{y}) + \beta \left\| [\mathbf{z}_1, \ldots, \mathbf{z}_K] \right\|_*, \tag{18}$$

where

$$\hat{f}_{\theta,c}(\{\mathbf{X}_k\}_{k=1}^K) = \sum_{k=1}^{K} \mathbf{X}_k \mathbf{z}_k$$

and $\left\| [\mathbf{z}_1, \ldots, \mathbf{z}_K] \right\|_* = \|\mathbf{Z}\|_* := \sum_i \sigma_i(\mathbf{Z})$ is the nuclear norm, i.e, sum of singular values, of $\mathbf{Z}$. In convex optimization, the nuclear norm is often used as a convex surrogate for the rank of a matrix, with the rank being a non-convex function. [35, 59].

IV-C Linear circular convolutional networks

Now, suppose that the patches are padded with zeros and extracted with stride one, and we have full-size filters that can be represented by circular convolution. Then the circular convolution version of (15) can be written as

$$\min_{\theta \in \Theta} \mathcal{L}(f_{\theta,c}(\mathbf{X}), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^{m} \left(\|\mathbf{w}_j^{(1)}\|_2^2 + \|\mathbf{w}_j^{(2)}\|_2^2\right), \tag{19}$$

where

$$f_{\theta,c}(\mathbf{X}) = \sum_{j=1}^{m} \mathbf{X} \mathbf{W}_j^{(1)} \mathbf{w}_j^{(2)},$$

and $\mathbf{W}_j^{(1)} \in \mathbb{R}^{d \times d}$ is a circulant matrix generated by a circular shift modulo d using the elements $\mathbf{w}_j^{(1)} \in \mathbb{R}^h$. Then, the SDP in (17) reduces to (see Appendix M-C)

$$\min_{\mathbf{z} \in \mathbb{C}^d} \mathcal{L}(\hat{f}_{\theta,c}(\hat{\mathbf{X}}), \mathbf{y}) + \beta \|\mathbf{z}\|_1, \tag{20}$$

where $\hat{\mathbf{X}} = \mathbf{X}\mathbf{F}$, $\mathbf{F} \in \mathbb{C}^{d \times d}$ is the Discrete Fourier Transform (DFT) matrix, and $\hat{f}_{\theta,c}(\hat{\mathbf{X}}) = \hat{\mathbf{X}}\mathbf{z}$. We note that certain linear CNNs trained via gradient descent exhibit similar spectral regularization properties [42].

V ℓ_p-NORM REGULARIZATION OF HIDDEN WEIGHTS

In this section, we reconsider two-layer neural network training problems with an alternative ℓ_p^2 regularization on the hidden neurons, which is a generalization of the setting in (3). Hence, we have the following optimization problem

$$p^* = \min_{\theta \in \Theta} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^{m} (\|\mathbf{w}_j^{(1)}\|_p^2 + |w_j^{(2)}|^2) \,. \tag{21}$$

After applying the scaling in Lemma II.1, we equivalently write (21) as

$$p^* = \min_{\theta \in \Theta_s} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \beta \|\mathbf{w}^{(2)}\|_1 \,, \tag{22}$$

where $\Theta_s := \{\theta \in \Theta : \|\mathbf{w}_j^{(1)}\|_p \leq 1, \forall j \in [m]\}$. Therefore, we have the following equivalent convex program for ℓ_p^2 regularized networks.

Corollary V.1. *As a result of Theorem II.1, the non-convex training problem in (21) can be cast as a finite dimensional convex program as follows*

$$p^* = \min_{\mathbf{w}, \mathbf{w}' \in \mathcal{C}(\mathbf{X})} \mathcal{L}(f_{\theta_c}(\mathcal{A}(\mathbf{X})), \mathbf{y}) + \beta \sum_{i=1}^{2P} \|\mathbf{w}_i\|_p, \tag{23}$$

where $f_{\theta_c}(\mathcal{A}(\mathbf{X})) = \mathcal{A}(\mathbf{X})\mathbf{w}$ and the rest of definitions directly follow from Theorem II.1.

We note that the case where $p = 1$ is regularized via $\sum_{i=1}^{2P} \|\mathbf{w}_i^{(1)}\|_1$ with the squared loss is equivalent to the LASSO feature selection method with additional linear constraints [16, 71].

VI INTERPOLATION REGIME (WEAK REGULARIZATION)

We now consider the minimum-norm variant of (8), which corresponds to interpolation or weak regularization, i.e., $\beta \to 0$. Suppose that the minimum value of the loss $\mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y})$ is zero, which is satisfied by many popular choices, e.g., squared loss and hinge loss. Taking the $\beta \to 0$ limit yields the following optimization problem

$$p^*_{\beta \to 0} = \min_{\theta \in \Theta_s} \|\mathbf{w}^{(2)}\|_1 \,, \text{ s.t. } \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) = 0. \tag{24}$$

Then, by Theorem II.1, the equivalent convex program for (24) is

$$p^*_{\beta \to 0} = \min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2 \text{ s.t. } \mathcal{L}(f_{\theta_c}(\mathcal{A}(\mathbf{X})), \mathbf{y}) = 0, \tag{25}$$

given that the set $\mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) = 0$ is convex.

Remark VI.1. *Notice that (25) represents a convex optimization problem that seeks to find a solution with minimum group norm and zero training error. Considering the squared loss, (25) further simplifies to*

$$\min_{\mathbf{w}, \mathbf{w}' \in \mathcal{C}(\mathbf{X})} \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2 \text{ s.t. } \mathcal{A}(\mathbf{X})\mathbf{w} = \mathbf{y}.$$

The above form illustrates an interesting contrast between our exact formulation and various kernel characterizations such as infinitely wide networks, the Neural Tangent Kernel, and implicit bias of interpolating two-layer neural network [17, 18, 46, 50, 54, 55, 67, 76]. These kernel formulations are related to approximating the training problem (3) as a minimum ℓ_2-norm kernel interpolation using a fixed kernel matrix constructed from $\mathbf{X}$. In contrast, our characterization minimizes $\ell_{2,1}$-norm encouraging feature selection after a fixed high-dimensional feature map. More importantly, unlike approximations of neural networks via kernel methods, our approach provides an exact characterization of the training problem (3).

VII Hyperplane arrangements

In this section, we will explore the diagonal matrices that appear in our convex optimization problem (7). These matrices are determined by the hyperplane arrangements of the data matrix $\mathbf{X}$. We will also discuss how to exactly construct these arrangements in order to solve the convex program to global optimality. Additionally, we will introduce a class of data matrices for which the arrangements corresponding to non-zero variables at the optimum can be simplified. This provides a significant computational complexity reduction in finding the optimal solution.

VII-A Constructing and approximating arrangement patterns

Constructing hyperplane arrangements has long been an important area of study in discrete mathematics and computational geometry. There are several analytic approaches to construct all possible hyperplane arrangements for a given data matrix $\mathbf{X}$. We refer the reader to [5, 26, 44, 57]. In [26], the authors present an algorithm that enumerates all possible hyperplane arrangements in $\mathcal{O}(n^r)$ time for a rank-r data matrix.

An alternative approach to reduce computational cost is to randomly sample a subset of hyperplane arrangements, as described in Section III-A. This approximate solution to the convex neural network problem (7) involves solving a subsampled convex program and is backed by approximation guarantees, as shown in Theorem III.2. Our numerical experiments in Section IX demonstrate that this approximation scheme performs exceptionally well in practice.

VII-B Spike-free data matrices

Now we show that the convex program (7) simplifies significantly for a certain class of data matrices. We first define the minimal set of hyperplane arrangements that globally optimizes (3) as

$$\mathcal{D}^* := \underset{\mathcal{D}}{\operatorname{argmin}} \left| \{\mathcal{D} \subseteq \mathcal{D}_{opt} : p^* = d^*\} \right|,$$

where $\mathcal{D}_{opt}$ is defined as

$$\mathcal{D}_{opt} := \left\{ \mathbf{D}_i \ : \ \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2 \cap \mathcal{C}(\mathbf{X})} \left| {\mathbf{v}^*}^T \mathbf{D}_i \mathbf{X} \mathbf{w}^{(1)} \right| = \beta \right\}$$

based on the dual characterization in (9), $\mathbf{v}^*$ denotes the optimal dual parameter, and $\mathcal{C}(\mathbf{X})$ is defined in Theorem II.1. We define $P^* := \mathcal{D}^*$ as the minimum number of hyperplane arrangements required to solve the convex program (7) for a given data matrix $\mathbf{X} \in \mathcal{X}$.

Next, we introduce a set of data matrices $\mathcal{X}$, called spike-free[2], for which one hyperplane arrangement is sufficient to solve (7) exactly.

As an example, whitened high-dimensional ($n \leq d$) data matrices that satisfy $\mathbf{X}\mathbf{X}^T = \mathbf{I}_n$ are spike-free data matrices as shown in [29]. We first define the set $\mathcal{Q}_{\mathbf{X}} := \{\phi(\mathbf{X}\mathbf{w}^{(1)}) \ : \ \mathbf{w}^{(1)} \in \mathcal{B}_2\}$. Then, we say that a data matrix $\mathbf{X}$ is spike-free if $\mathcal{Q}_{\mathbf{X}}$ can be equivalently represented as $\mathbf{X}\mathcal{B}_2 \cap \mathbb{R}^n_+$, where $\mathbf{X}\mathcal{B}_2 = \{\mathbf{X}\mathbf{w}^{(1)} \ : \ \mathbf{w}^{(1)} \in \mathcal{B}_2\}$. More precisely, we define the set of spike-free data matrices as $\mathcal{X} = \{\mathbf{X} \in \mathbb{R}^{n \times d} : \mathcal{Q}_{\mathbf{X}} = \mathbf{X}\mathcal{B}_2 \cap \mathbb{R}^n_+\}$. Assuming $\mathbf{X}$ is spike-free, the output of the ReLU activation in (9), i.e., $\phi(\mathbf{X}\mathbf{w}^{(1)})$, can be replaced with $\{\mathbf{X}\mathbf{w}^{(1)} : \mathbf{X}\mathbf{w}^{(1)} \geq 0\}$, which corresponds to a single hyperplane arrangement $\mathcal{D}^* = \mathbf{I}_n$. Consequently, the number of hyperplane arrangements in Theorem II.1 reduces to one, i.e., $P^* = 1$ and $\mathcal{D}^* = \mathbf{I}_n$. Based on this observation, the equivalent convex program for spike-free data matrices is as follows.

Theorem VII.1. *Given a spike-free data matrix $\mathbf{X} \in \mathcal{X}$, the equivalent convex program for the non-convex problem in (3) is given by*

$$\min_{\mathbf{w}, \mathbf{w}' \in \mathcal{C}_s(\mathbf{X})} \mathcal{L}(\mathbf{X}(\mathbf{w}' - \mathbf{w}), \mathbf{y}) + \beta \left(\|\mathbf{w}\|_2 + \|\mathbf{w}'\|_2 \right), \tag{26}$$

where

$$\mathcal{C}_s(\mathbf{X}) := \{\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d \ : \ \mathbf{X}\mathbf{w} \geq 0, \ \mathbf{X}\mathbf{w}' \geq 0\}.$$

The above result shows that the convex program in (7) reduces to a simple mixture of two linear models for spike-free data matrices.

[2]The definition and further properties of spike-free matrices can be found in [27, 29].

VIII VECTOR OUTPUT NETWORKS

In this section, we consider a neural network with C outputs, which are commonly used for vector valued prediction, e.g., multi-class classification or vector regression. Here, we have matrix valued targets $\mathbf{Y} \in \mathbb{R}^{n \times C}$, and the non-convex regularized training problem is as follows

$$p_v^* := \min_{\theta \in \Theta} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{Y}) + \frac{\beta}{2}(\|\mathbf{W}^{(1)}\|_F^2 + \|\mathbf{W}^{(2)}\|_F^2)\,, \tag{27}$$

where $f_\theta(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)}$. By applying the scaling argument in Lemma II.1, (27) can be written as

$$p_v^* := \min_{\theta \in \Theta_s} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \beta \sum_{j=1}^{m} \|\mathbf{w}_j^{(2)}\|_2\,. \tag{28}$$

Then, applying similar steps as in Theorem II.1, we obtain the following result.

Theorem VIII.1. *The non-convex training problem in (27) can be cast as a finite dimensional convex program as follows*

$$p_v^* = \min_{\mathbf{W}_i \in \mathbb{R}^{d \times C}} \mathcal{L}(f_{\theta_c}(\mathcal{A}(\mathbf{X})), \mathbf{Y}) + \beta \sum_{i=1}^{P} \|\mathbf{W}_i\|_{\mathcal{C}_i}, \tag{29}$$

where $\theta_c := \{\{\mathbf{W}_i\}_{i=1}^P\}$, $f_{\theta_c}(\mathbf{X})$, *and the constrained nuclear norm* $\|\cdot\|_{\mathcal{C}_i}$ *are defined as*

$$f_{\theta_c}(\mathbf{X}) := \sum_{i=1}^{P} \mathbf{D}_i \mathbf{X} \mathbf{W}_i$$
$$\|\mathbf{W}\|_{\mathcal{C}_i} := \min_{t \geq 0} \; t \quad \text{s.t.} \quad \mathbf{W} \in t\,\mathrm{Conv}\left\{\mathbf{Z} = \mathbf{u}\mathbf{g}^T \;:\; \mathbf{g} \in \mathcal{B}_2,\, \mathbf{u} \in \mathcal{B}_2 \cap \mathcal{P}_i,\, \|\mathbf{Z}\|_* \leq 1\right\}$$

where Conv *denotes the convex hull of its argument and* $\mathcal{P}_i := \{\mathbf{u} \in \mathbb{R}^d \;:\; (2\mathbf{D}_i - \mathbf{I}_n)\mathbf{X}\mathbf{u} \geq 0\}$ *are linear constraints.*

We note that the norm $\|\cdot\|_{\mathcal{C}_i}$ is a constrained version of the nuclear norm, and therefore induces low-rank structure in the variables $\mathbf{W}_1, ..., \mathbf{W}_P$. Therefore, in contrast to scalar output networks, Theorem VIII.1 shows that weight decay regularized neural networks with piecewise linear activations can be equivalently characterized as piecewise low-rank convex models. We further observe that dropping the linear constraints $\mathcal{P}_i$ from the definition of $\|\mathbf{W}\|_{\mathcal{C}_i}$ reduces the constrained nuclear norm to the ordinary nuclear norm. In this case, the regularization term in the convex objective (29) simplifies to the sum of nuclear norms, which is a natural generalization of the group ℓ_1 regularizer in (7). A numerical algorithm to solve (29) to global optimality was proposed in [64]. Dropping the linear constraints was investigated in [43]. Additionally, recent literature showed that nuclear norm also plays a role in the implicit regularization of linear networks trained via gradient descent [3, 41, 58].

VIII-A ℓ_1^2 regularization for the second-layer

Although the problem in (29) is convex, handling the constrained nuclear norm $\|\cdot\|_{\mathcal{C}_i}$ can be challenging for high-dimensional problems. To alleviate this, we consider a modification of the weight decay regularization as follows

$$p_{v1}^* := \min_{\theta \in \Theta} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \frac{\beta}{2} \sum_{j=1}^{m} (\|\mathbf{w}_j^{(1)}\|_2^2 + \|\mathbf{w}_j^{(2)}\|_1^2)\,. \tag{30}$$

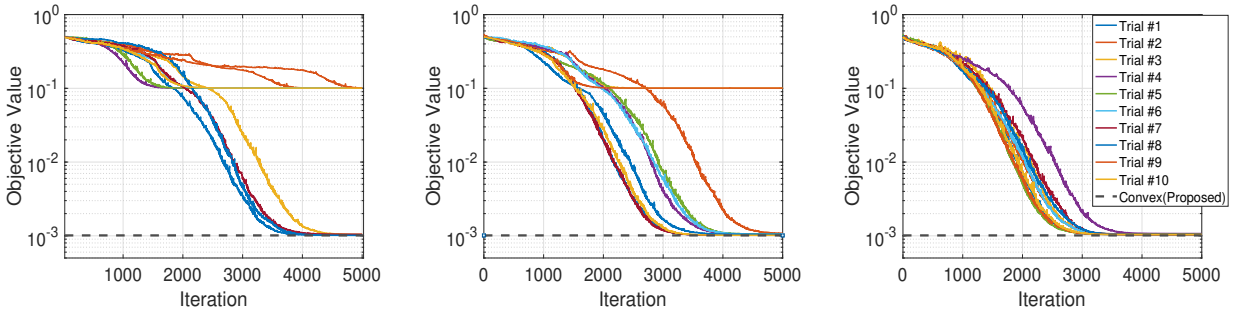
Next, we show that the above problem can be cast as a polynomial-time solvable convex program.

Theorem VIII.2. *The non-convex problem in (30) can be equivalently formulated as the following convex program*

$$p_{v1}^* = \min_{\mathbf{w}_l \in \mathcal{C}(\mathbf{X})} \sum_{l=1}^{C} \mathcal{L}(\mathcal{A}(\mathbf{X})\mathbf{w}_l, \mathbf{y}_l) + \beta \sum_{l=1}^{C} \sum_{i=1}^{2P} \|\mathbf{w}_{l,i}\|_2, \tag{31}$$

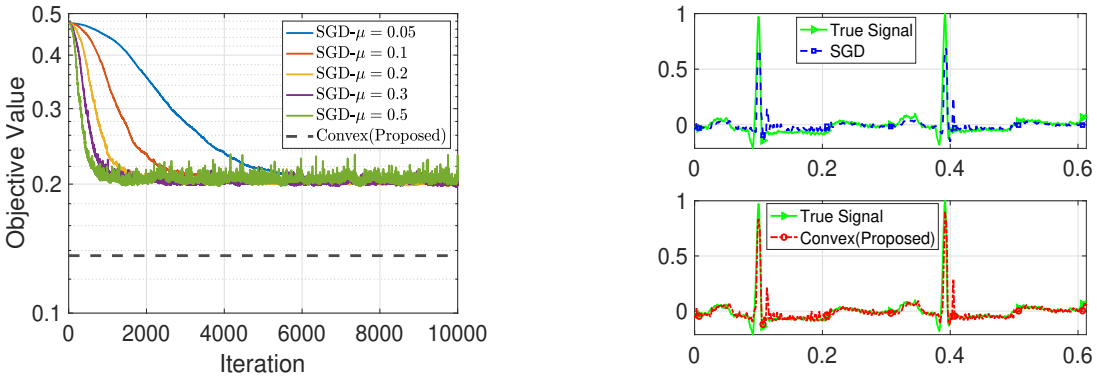
where the set $\mathcal{C}(\mathbf{X})$ *and* P *are defined as in Theorem II.1.*

We remark that (31) can be decomposed into C independent convex programs, each of which is the same as (7). Therefore, unlike (29), the problem in (31) can be efficiently solved via standard convex optimization solvers.



(a) $m = 8$ (b) $m = 15$ (c) $m = 50$

Fig. 7: Training cost of a two-layer ReLU network trained with SGD (10 initialization trials) on a one dimensional dataset with $(n, d, \beta) = (5, 1, 10^{-3})$, where Convex denotes proposed convex programming approach in (7). SGD can be stuck at local minima for small m, while the proposed approach is optimal as guaranteed by Theorem II.1.



(a) Objective value (b) Test prediction

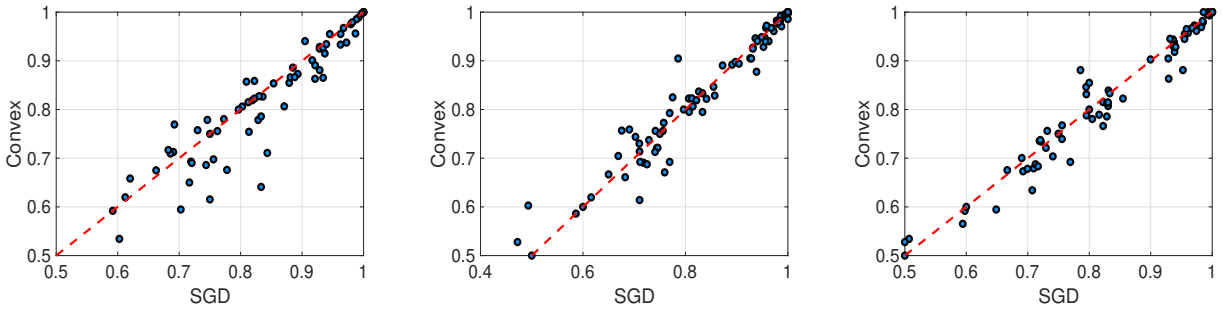
Fig. 8: Prediction performance comparison of a two-layer ReLU network trained with SGD and the convex program (7) on the ECG dataset, where $(n, d, m, \beta) = (2393, 3, 50, 0.005)$ and μ denotes the learning rate for SGD. As predicted by our theory, SGD provides poor training and test performance compared to the convex program (7).

IX Numerical Experiments

In this section[3], we present numerical experiments to verify our theoretical results. We start with a one-dimensional toy dataset with $n = 5$ given by $\mathbf{X} = [-2 \ -1 \ 0 \ 1 \ 2]^T$ and $y = [1 \ -1 \ 1 \ 1 \ -1]^T$, where we include a bias term by concatenating a column of ones to the data matrix $\mathbf{X}$. We then train a two-layer ReLU network with SGD and the proposed convex program using squared loss. In Figure 7, we plot the value of the regularized objective function with respect to the iteration index for SGD in 10 independent trials for initial parameters. We solve the convex program in (7) via CVX [39] and plot the objective value as a horizontal dashed line denoted as "Convex". Additionally, we repeat the same experiment for the different number of neurons: $m = 8, 15$, and 50. As demonstrated in the figure, SGD is likely to get stuck at local minima when the number of neurons is small. As we increase m, the number of trials that successfully converge to global minima gradually increases. We also note that Convex achieves the optimal objective value as claimed in the previous sections.

We also compare the prediction performance of neural network training algorithms on a time series prediction problem, where we use the ECG data in [38]. For each sample y_i, we consider the previously observed three samples

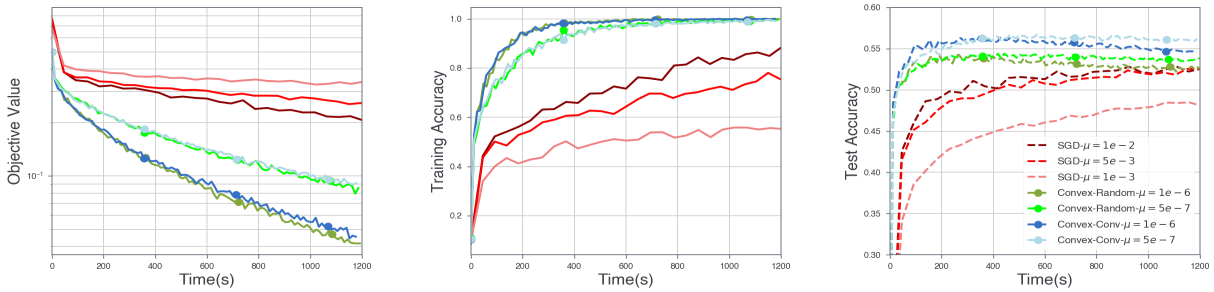
[3]We provide the details about our experimental setup and additional experiments in A.



(a) Test accuracy-$m = 10$ (b) Test accuracy-$m = 50$ (c) Test accuracy-$m = 200$

(d) Test error-$m = 10$ (e) Test error-$m = 50$ (f) Test error-$m = 200$

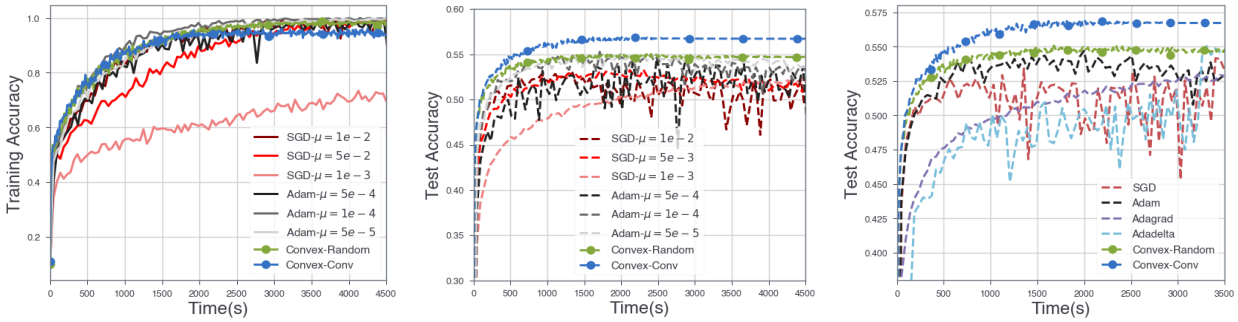
Fig. 9: Test accuracy and error values of a two-layer ReLU network trained with SGD and the convex program in (7) on the UCI datasets with $\beta = 10^{-3}$, where each blue dot denotes a certain dataset and the corresponding axis values represent the performance of training algorithms on the dataset.



(a) Objective value for different learning rates (μ) (b) Training accuracy (10-class) for different learning rates (μ) (c) Test accuracy (10-class) for different learning rates (μ)

Fig. 10: Comparison of the methods on the CIFAR-10 dataset, where $(n, d, m, C, \beta) = (50000, 3072, 4096, 10, 10^{-3})$, batch size is 1000, $P = m$, and the activation function is ReLU. The proposed convex optimization problem is solved using SGD. Here, we use solid and dashed lines for training and test results, respectively.

as our features, i.e., $\mathbf{x}_i = [y_{i-1},\, y_{i-2},\, y_{i-3}]^T$ and consider predicting the value y_i. Therefore, we obtained a time series dataset with $n = 2393$ and $d = 3$. In Figure 8, we plot the training objective in (3) and test predictions, where we use a batch size of 100 for SGD. In addition, we also experiment with different learning rates μ as demonstrated in 8a. Here, we observe that the SGD trials fail to achieve the optimal training objective value obtained by our convex optimization method. Consequently, SGD also exhibits poor predictive performance in the test set as shown in Figure 8b.

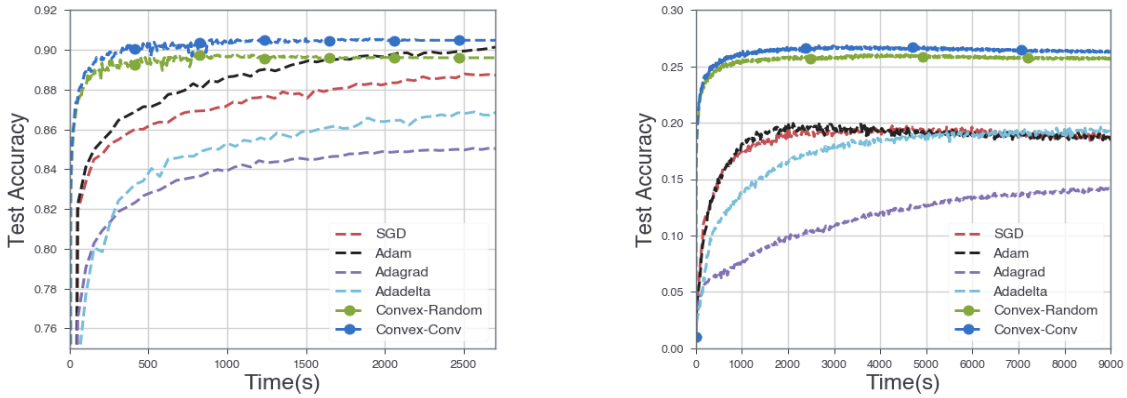


(a) Training accuracy (10-class) for different learning rates (μ)

(b) Test accuracy (10-class) for different learning rates (μ)

(c) Test accuracy (10-class) for different optimizers

Fig. 11: Comparison of the methods on the CIFAR-10 dataset, where $(n, d, m, C, \beta) = (50000, 3072, 4096, 10, 10^{-3})$, batch size is 1000, $P = m$, and the activation function is ReLU. The proposed convex optimization problem is solved using Adagrad. Here, we use solid and dashed lines for training and test results, respectively.



(a) Fashion MNIST (10-class)

(b) CIFAR-100 (100-class)

Fig. 12: Comparison of the methods on Fashion MNIST with $(n, d, m, C, \beta) = (60000, 784, 4096, 10, 10^{-3})$ and CIFAR-100 with $(n, d, m, C, \beta) = (50000, 3072, 512, 100, 10^{-3})$, where batch size is 1000, $P = m$, and the activation function is ReLU for both datasets. We use Adam to solve the proposed convex optimization problem.

Next, we present numerical experiments performed on several datasets taken from UCI machine learning repository [25]. In particular, we consider small/medium scale datasets used in [4] and then follow the same preprocessing steps. Specifically, we use 90 UCI datasets with the number of samples less than 5000. For each of these datasets, we use a conventional regression framework with squared loss and then plot the test accuracy and error obtained by SGD and Convex in Figure 9. Similar to Figure 7, as the number of neurons m increases, the performance gap between SGD and Convex closes, and the distribution of data points approaches a line with slope one.

We also perform experiments on some well-known image classification datasets, namely CIFAR-10, CIFAR-100, and Fashion-MNIST [48, 77]. For all of these experiments, we use the convex program in Theorem VIII.2, where the problem decomposes into C independent problems for a network with C outputs. Moreover, we use the approximate version of the convex program, where the hyperplane arrangements are sampled randomly as discussed in Section III-A. We sample hyperplane arrangements using a normal distribution and denote this approach as "Convex-Random". We also randomly generate convolutional filters and use their sign patterns as hyperplane arrangements for the convex program, which is denoted as "Convex-Conv". In addition, we apply K-Means based preprocessing as proposed in [20, 21] to the raw data matrix to obtain a richer set of features, which are presented as preprocessing+convex and preprocessing+non-convex in Table II (see Algorithm 2 in A for the full description of the

TABLE II: Highest test accuracies achieved by 1-layer Neural Network (NN), which is the conventional logistic regression method, 2-layer NN trained via the standard non-convex approaches, 2-layer NN trained via the proposed convex approaches, and 2-layer NN trained on a data matrix preprocessed via K-means clustering algorithm (see Algorithm 2 for the pseudocode)

	CIFAR-10	**Fashion MNIST**	**CIFAR-100**
1-layer NN (Logistic regression)	0.4076	0.8392	0.0939
2-layer NN (non-convex)	0.5416	0.9002	0.1995
2-layer NN (convex)	0.5688	0.9057	0.2684
2-layer NN (preprocessing+non-convex)	0.7770	0.9260	0.4771
2-layer NN (preprocessing+convex)	**0.8163**	**0.9327**	**0.5393**

algorithm). We first consider a ten class classification problem on CIFAR-10 with the parameters $(n, d, m, C, \beta) = (50000, 3072, 4096, 10, 10^{-3})$, batch size of 1000, and the ReLU activation. In Figure 10, we compare these two approaches against SGD with different learning rates (μ) and demonstrate the superior performance of our convex models in terms of objective value, training, and test accuracies. Among the convex models, we observe that Convex-Conv substantially improves upon Convex-Random. In addition, preprocessing+convex yields $\sim 25\%$ accuracy improvement compared to other convex models (see Table II). Furthermore, we compare our convex models against the non-convex formulation trained with different optimizers in Figure 11. Here, our convex models achieve better training and test performance compared to the non-convex methods. Similarly, we also validate the performance of the proposed convex model on Fashion MNIST with $(n, d, m, C, \beta) = (60000, 784, 4096, 10, 10^{-3})$ and CIFAR-100 with $(n, d, m, C, \beta) = (50000, 3072, 512, 100, 10^{-3})$, where the batch size is 1000 and the activation is ReLU. For Fashion-MNIST, even though the convex models again achieve higher test accuracies compared to the non-convex ones in Figure 12, Adam also provides comparable performance. However, for CIFAR-100 (with $C = 100$), we observe a notable accuracy improvement with respect to the non-convex approaches.

X Conclusion

We studied two-layer neural network architectures with piecewise linear activations and introduced a convex optimization framework to analyze the regularized training problem. We derived exact convex optimization formulations for the original non-convex training problem, which can be globally optimized by convex solvers with polynomial-time complexity. These convex representations reside in a higher dimensional space, where the data matrix is partitioned over all possible hyperplane arrangements and group sparsity or low-rankness is enforced via group ℓ_p, ℓ_1 or nuclear norm regularizers. In addition, our results show that the form of the structural regularization induced on the weights of the convex model is a function of the architecture, the number of outputs, and the regularization in the non-convex problem. We believe that this result sheds light into the generalization of neural network models and their architectural bias, which are extensively studied in the recent literature. Our results show that neural networks with piecewise linear activations can be seen as parsimonious piecewise linear models. We believe that this perspective offers a clearer interpretation of these non-convex models, as their convex counterparts are more transparent and easier to understand. Moreover, due to convexity, the equivalent training problems do not require non-convex optimization heuristics or extensive hyperparameter searches such as choosing a proper learning rate schedule and initialization scheme.. We showed that randomly sampling hyperplane arrangements and solving the subsampled convex problem works extremely well in practice. Furthermore, we proposed an approximation algorithm that leverages low-rank approximation of the data matrix such that the equivalent convex program can be globally optimized with polynomial-time complexity in terms of all the problem parameters, i.e., the number of samples n, the feature dimension d, and the number of neurons m. We also proved strong approximation bounds for this algorithm.

Our work poses multiple promising open problems to explore. First, one can obtain a better understanding of neural networks, their optimization landscapes, and their generalization properties by leveraging our equivalent convex formulations. In the light of our results, backpropagation can be viewed as a heuristic method to solve the

convex program. Moreover, the loss landscape of the non-convex objective and the dynamics of gradient based optimizers can be further investigated by utilizing the optimal set of the convex program. After our work, this was explored in [49], where the authors reported interesting results regarding the hidden convex landscape of the non-convex objective. Furthermore, one can extend our convex optimization framework to various other architectures, e.g., CNNs, recurrent networks, transformers, and autoencoders. Here, we extended our approach to certain simple CNNs. Recently, [28] further extended our approach to CNNs with ReLU activations and various pooling strategies. Similarly, based on this work, a series of follow-up papers analyzed deep linear networks [31], generative networks [43, 65], deep ReLU networks [30], and transformer networks [32, 66] via convex duality. In addition, [33] analyzed Batch Normalization, which is a popular heuristic to stabilize the training of deep neural networks via our convex methodology. Finally, to the best of our knowledge, this work provides the first polynomial-time training algorithm to *globally* train two-layer neural network architectures for any data matrix with fixed rank. We conjecture that more efficient solvers for the convex program can be developed for larger scale experiments by utilizing the connection to sparse models [14, 24].

APPENDIX A
DETAILS ABOUT OUR EXPERIMENTAL SETUP AND ADDITIONAL NUMERICAL RESULTS

Algorithm 2 `Convex neural network training via K-means feature embeddings`

Set P_c, ϵ, h, k (in our experiments $(P_c, \epsilon, h, k) = (4\text{x}10^5, 0.1, 6, 9)$)
Randomly extract P_c patches of size $h \times h$ from the dataset: $\{\mathbf{p}_i\}_{i=1}^{P_c}$
for $i = 1 : P_c$ **do**
    Normalize the patch: $\bar{\mathbf{p}}_i = \frac{\mathbf{p}_i - \text{mean}(\mathbf{p}_i)}{\sqrt{\text{var}(\mathbf{p}_i) + \epsilon}}$
end for
Form a patch matrix: $\mathbf{P} = [\bar{\mathbf{p}}_1 \ \ldots \ \bar{\mathbf{p}}_{P_c}]$
Apply ZCA whitening to the patch matrix:

$$[\mathbf{V}, \mathbf{D}] = \text{eig}(\text{cov}(\mathbf{P}))$$
$$\tilde{\mathbf{P}} = \mathbf{V}(\mathbf{D} + \epsilon \mathbf{I})^{-\frac{1}{2}} \mathbf{V}^T \mathbf{P}$$

Cluster patches using K-means as in [20, 21] to obtain m cluster means: $\{\mathbf{c}_j\}_{j=1}^{m}$
for $i = 1 : n$ **do**
    Extract all the patches of size $h \times h$ in the image $\mathbf{X}_i \in \mathbb{R}^{d \times d}$: $\mathbf{X}_{ip} \in \mathbb{R}^{h^2 \times (d-h+1)^2}$
    Compute pairwise distances between patches and cluster means and then threshold the distances: $\mathbf{K}_{dist} \in \mathbb{R}^{(d-h+1)^2 \times m}$
    Threshold the distances as: $\bar{\mathbf{K}}_{dist} = \max\{\mathbf{K}_{dist} - \mathbf{1}\boldsymbol{m}^T, 0\}$, where $\boldsymbol{m}$ is a vector of means for each row of $\mathbf{K}_{dist}$
    Apply $k \times k$ pooling (with stride k) on the reshaped data of size $(d - h + 1) \times (d - h + 1) \times m$: $\mathbf{Q} = \text{pooling}(\bar{\mathbf{K}}_{dist})$
    Flatten the resulting vector: $\bar{\mathbf{x}}_i = \text{flatten}(\mathbf{Q}) \in \mathbb{R}^{d_{new}}$
end for
Form a new data matrix consisting of $\{\bar{\mathbf{x}}_i\}_{i=1}^{n}$: $\bar{\mathbf{X}} \in \mathbb{R}^{n \times d_{new}}$
Solve the convex training problem in (32) using $\bar{\mathbf{X}}$

In this section, we provide detailed information about our experimental setup.

We note that for the synthetic experiment in Figure 5, we obtain the data labels $\mathbf{y} \in \mathbb{R}^n$ by first forward propagating the input data matrix through a randomly initialized two-layer ReLU network with five neurons and then adding a noise term. Particularly, we first randomly generate the layer weights as $\mathbf{w}_j^{(1)} \sim N(\mathbf{0}, \mathbf{I}_d)$ and $w_j^{(2)} \sim N(0, 1)$, $\forall j \in [5]$ and then obtain the labels as $\mathbf{y} = \left(\mathbf{X}\mathbf{W}^{(1)}\right)_+ \mathbf{w}^{(2)} + 0.1\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_n)$.

For small scale experiments in Figure 8 and 9, we use CVX [39] and CVXPY [1, 23] with the SDPT3 solver [74] to solve convex optimization problems in (7) and (31). Moreover, the training is performed on a CPU with 50GB of RAM. For ECG and UCI experiments, we use the $66\% - 34\%$, $60\% - 40\%$ splitting ratio for the training and test sets. Moreover, the learning rate of SGD is tuned via a grid-search on the training split. Specifically, we try different values and choose the best performing learning rate on the validation datasets.

For the image classification experiments in Figure 10, 11, and 12, we use a GPU with 50GB of memory. In particular, to solve the convex optimization problems in (7), we first introduce an equivalent unconstrained convex problem as follows

$$\min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \mathcal{L}(\mathcal{A}(\mathbf{X})\mathbf{w}, \mathbf{y}) + \beta \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2 + \rho \mathbf{1}^T \sum_{i=1}^{P} \Big(\big(-(2\mathbf{D}_i - \mathbf{I}_n)\mathbf{w}_i \big)_+ + \big(-(2\mathbf{D}_i - \mathbf{I}_n)\mathbf{w}_{i+P} \big)_+ \Big) \tag{32}$$

where $\rho > 0$ is a trade-off parameter. Now, since the equivalent problem in (32) is an unconstrained convex optimization problem, we can directly optimize its parameters using standard first order optimizers such as SGD and Adam. Therefore, we can use PyTorch to optimize both the non-convex objective in (3) and the convex objective in (32) on the larger scale datasets, e.g., CIFAR-10, CIFAR-100, and Fashion-MNIST. For the learning rates, we

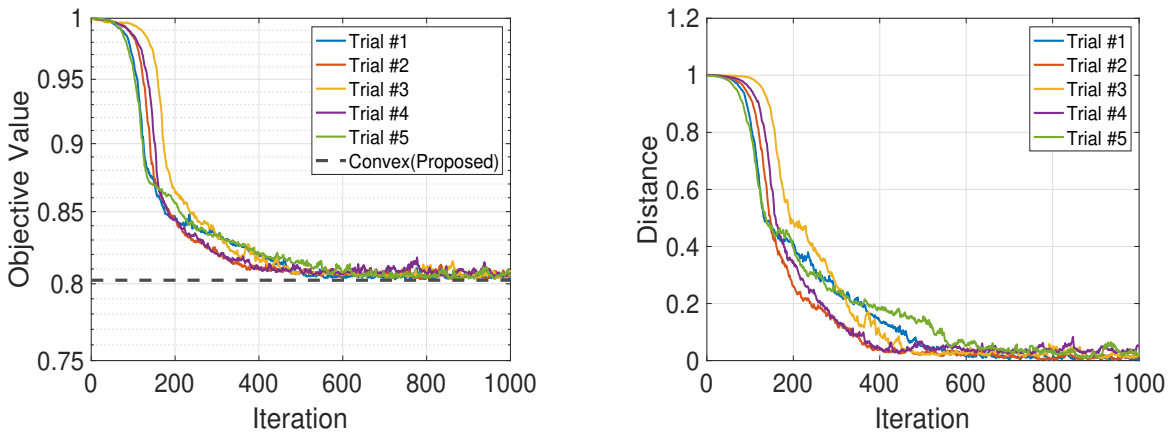
TABLE III: Layerwise learning test accuracies for CIFAR-10

Network depth	**2**	**3**	**4**
Layerwise convex NN test accuracy	0.477	0.617	0.663

again follow the same grid-search technique. In addition, for all the experiments, we set the trade-off parameter to $\rho = 0.01$.

We also train a two-layer linear CNN architecture on a subset of CIFAR-10, where we denote the proposed convex program in (20) as Convex. In Figure 13, we plot both the objective value and the Euclidean distance between the filters found by GD and Convex for 5 independent realizations with $n = 387$, $m = 30$, $h = 10$, and batch size of 60. In this experiment, all the independent realizations converge to the objective value obtained by Convex and find almost the same filters with Convex.

Finally, we conducted an additional experiment where we applied our convex training approach in a layerwise manner. Specifically, we followed the strategy considered in [8, 53, 63], which involves a greedy layerwise training procedure. This method starts by training a shallow two-layer convolutional network, which is done using our convex reformulations. After training the full two-layer network, the final layer is removed, and the process is repeated for the next layer, treating the previous layer's output as the input to the subsequent layers. This progressive training ensures that each layer learns a robust representation before moving on to deeper layers, ultimately leading to an efficient and scalable training process. As shown in Table III, this approach can significantly boost the test accuracy up to 66.3% by adding one more layer sequentially, whereas training a standard two-layer NN via convex optimization reaches at most 56.88% test accuracy in our experiments.



(a) Objective value

(b) Distance to the solution of the convex program

Fig. 13: Training accuracy of a two-layer linear CNN trained with SGD (5 initialization trials) on a subset of CIFAR-10, where Convex denotes the proposed convex program in (20). Filters found via SGD converge to the solution of (20).

APPENDIX B
PROOF OF LEMMA II.1

We first note that similar observations are also made in the previous studies [31, 51, 54, 56, 67, 72].

For any $\theta \in \Theta$, we can rescale the parameters as $\bar{\mathbf{w}}_j^{(1)} = \gamma_j \mathbf{w}_j^{(1)}$ and $\bar{w}_j^{(2)} = w_j^{(2)}/\gamma_j$, for any $\gamma_j > 0$. Then, the network output becomes

$$f_{\bar{\theta}}(\mathbf{X}) = \sum_{j=1}^{m} \phi\big(\mathbf{X}\bar{\mathbf{w}}_j^{(1)}\big)\bar{w}_j^{(2)} = \sum_{j=1}^{m} \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\gamma_j\big)\frac{w_j^{(2)}}{\gamma_j} = \sum_{j=1}^{m} \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big)w_j^{(2)} = f_\theta(\mathbf{X}),$$

which proves $f_{\bar{\theta}}(\mathbf{X}) = f_\theta(\mathbf{X})$. In addition to this, given $p \geq 1$, we have the following basic inequality

$$\frac{1}{2}\sum_{j=1}^{m}(\|\mathbf{w}_j^{(1)}\|_p^2 + |w_j^{(2)}|^2) \geq \sum_{j=1}^{m}(\|\mathbf{w}_j^{(1)}\|_p |w_j^{(2)}|\ ),$$

where the equality is achieved with the scaling choice $\gamma_j = \big(\frac{|w_j^{(2)}|}{\|\mathbf{w}_j^{(1)}\|_p}\big)^{\frac{1}{2}}$ is used. Since the scaling operation does not change the right-hand side of the inequality, we can set $\|\mathbf{w}_j^{(1)}\|_p = 1, \forall j \in [m]$. Therefore, the right-hand side becomes $\|\mathbf{w}^{(2)}\|_1$.

Now, let us consider a modified version of the problem, where the unit norm equality constraint is relaxed as $\|\mathbf{w}_j^{(1)}\|_p \leq 1$. Let us also assume that for a certain index j, we obtain $\|\mathbf{w}_j^{(1)}\|_p < 1$ with $w_j^{(2)} \neq 0$ as an optimal solution. This shows that the unit norm inequality constraint is not active for $\mathbf{w}_j^{(1)}$, and hence removing the constraint for $\mathbf{w}_j^{(1)}$ will not change the optimal solution. However, when we remove the constraint, $\|\mathbf{w}_j^{(1)}\|_p \to \infty$ reduces the objective value since it yields $w_j^{(2)} = 0$. Therefore, we have a contradiction, which proves that all the constraints that correspond to a nonzero $w_j^{(2)}$ must be active for an optimal solution. This also shows that replacing $\|\mathbf{w}_j^{(1)}\|_p = 1$ with $\|\mathbf{w}_j^{(1)}\|_p \leq 1$ does not change the solution to the problem.

APPENDIX C
CONVEX DUALITY FOR TWO-LAYER NETWORKS

Now we introduce our main technical tool for deriving convex representations of the non-convex objective function (3). We start with the ℓ_1 penalized representation, which is equivalent to (3)

$$p^* = \min_{\theta \in \Theta_s} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \beta\|\mathbf{w}^{(2)}\|_1\,. \tag{33}$$

Replacing the minimization problem for the output layer weights $\mathbf{w}^{(2)}$ with its convex dual, we obtain (see Appendix M-A)

$$p^* = \min_{\mathbf{w}_j^{(1)} \in \mathcal{B}_2} \max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \left|\mathbf{v}^T\phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big)\right| \leq \beta,\ \forall j \in [m],\ .$$

Interchanging the order of $\min$ and $\max$, we obtain the lower-bound d^* via weak duality

$$p^* \geq d^* := \max_{\mathbf{v}} \min_{\theta \in \Theta_s} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \left|\mathbf{v}^T\phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big)\right| \leq \beta. \tag{34}$$

The above problem is a convex *semi-infinite* optimization problem with n variables and infinitely many constraints. We will show that strong duality holds, i.e., $p^* = d^*$, as long as the number of hidden neurons m satisfies $m \geq m^*$ for some $m^* \in \mathbb{N}$, $m^* \leq n+1$, which will be defined in the sequel. As it is shown, m^* can be smaller than $n+1$. The dual of the dual program (34) can be derived using standard semi-infinite programming theory [36], and corresponds to the bi-dual of the non-convex problem (3).

Now we briefly introduce basic properties of signed measures that are necessary to state the dual of (34) and refer the reader to [6, 61] for further details. Consider an arbitrary measurable input space $\mathcal{X}$ with a set of continuous basis functions $\pi_{\mathbf{w}^{(1)}} : \mathcal{X} \to \mathbb{R}$ parameterized by $\mathbf{w}^{(1)} \in \mathcal{B}_2$. We then consider real-valued Radon measures equipped with the uniform norm [62]. For a signed Radon measure $\boldsymbol{\mu}$, we can define an infinite width neural network output for the input $\mathbf{x} \in \mathcal{X}$ as $f(\mathbf{x}) = \int_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \pi_{\mathbf{w}}^{(1)}(\mathbf{x}) d\boldsymbol{\mu}(\mathbf{w}^{(1)})$. The total variation norm of the signed measure $\boldsymbol{\mu}$ is defined as the supremum of $\int_{\mathbf{w}^{(1)} \in \mathcal{B}_2} q(\mathbf{w}^{(1)}) d\boldsymbol{\mu}(\mathbf{w}^{(1)})$ over all continuous functions $q(\mathbf{w}^{(1)})$ that satisfy $|q(\mathbf{w}^{(1)})| \leq 1$.

Consider the basis functions $\pi_{\mathbf{w}^{(1)}}(\mathbf{x}) = \phi(\mathbf{x}^T\mathbf{w}^{(1)})$. We may express networks with finitely many neurons as in (1) by

$$f(\mathbf{x}) = \sum_{j=1}^{m} \pi_{\mathbf{w}_j^{(1)}}(\mathbf{x}) w_j^{(2)} \,,$$

which corresponds to $\boldsymbol{\mu} = \sum_{j=1}^{m} w_j^{(2)} \delta(\mathbf{w}^{(1)} - \mathbf{w}_j^{(1)})$ where δ is the Dirac delta measure. And the total variation norm $\|\boldsymbol{\mu}\|_{TV}$ of $\boldsymbol{\mu}$ reduces to the ℓ_1-norm $\|\mathbf{w}^{(2)}\|_1$.

We state the dual of (34) (see Section 2 of [68] and Section 8.6 of [36]) as follows

$$d^* \leq p_\infty^* = \min_{\mu} \mathcal{L}\left(\int_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \phi(\mathbf{X}\mathbf{w}^{(1)}) d\boldsymbol{\mu}(\mathbf{w}^{(1)}), \mathbf{y} \right) + \beta \|\boldsymbol{\mu}\|_{TV}. \tag{35}$$

Furthermore, an application of Caratheodory's theorem shows that the infinite dimensional bi-dual (35) always has a solution that is supported with m^* Dirac deltas, where $m^* \leq n+1$ [61]. Therefore, we have

$$\begin{aligned} p_\infty * = & \min_{\substack{\mathbf{w}_j^{(1)} \in \mathcal{B}_2 \\ \{w_j^{(2)}, \mathbf{w}_j^{(1)}\}_{j=1}^{m^*}}} \mathcal{L}\left(\sum_{j=1}^{m^*} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) w_j^{(2)}, \mathbf{y} \right) + \beta \|\mathbf{w}^{(2)}\|_1 \,, \\ = & \, p^* \,, \end{aligned}$$

as long as $m \geq m^*$. We show that strong duality holds, i.e., $d^* = p^*$ below. In the sequel, we illustrate how m^* can be determined via a finite dimensional parameterization of (34) and its dual.

Semi-infinite strong duality Note that the semi-infinite problem (34) is convex. We first show that the optimal value is finite. For $\beta > 0$, it is clear that $\mathbf{v} = 0$ is strictly feasible, and achieves 0 objective value. Note that the optimal value p^* satisfies $p^* \leq \|\mathbf{y}\|_2^2$ since this value is achieved when all the parameters are zero. Consequently, Theorem 2.2 of [68] implies that strong duality holds, i.e., $p^* = d_\infty^*$, if the solution set of the semi-infinite problem in (34) is nonempty and bounded. Next, we note that the solution set of (34) is the Euclidean projection of $\mathbf{y}$ onto the polar set $(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ$ which is a convex, closed and bounded set since $\phi(\mathbf{X}\mathbf{w}^{(1)})$ can be expressed as the union of finitely many convex closed and bounded sets. □

Alternative proof of the semi-infinite strong duality It holds that $p^* \geq d^*$ by weak duality in (34). Theorem II.1 proves that the objective value of (37) is identical to the value of (3) as long as $m \geq m^*$. Therefore we have $p^* = d^*$. □

*C-1 A geometric insight: neural gauge function*An interesting geometric insight can be provided in the weakly regularized case where $\beta \to 0$. In this case, minimizers of (33) and hence (3) approach minimum-norm interpolation $p_{\beta\to 0}^* := \lim_{\beta\to 0} \beta^{-1} p^*$ given by

$$p_{\beta\to 0}^* = \min_{\{\mathbf{w}_j^{(1)}, w_j^{(2)}\}_{j=1}^m} \sum_{j=1}^{m} |w_j^{(2)}| \text{ s.t. } \sum_{j=1}^{m} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) w_j^{(2)} = \mathbf{y}, \, \mathbf{w}_j^{(1)} \in \mathcal{B}_2 \, \forall j.$$

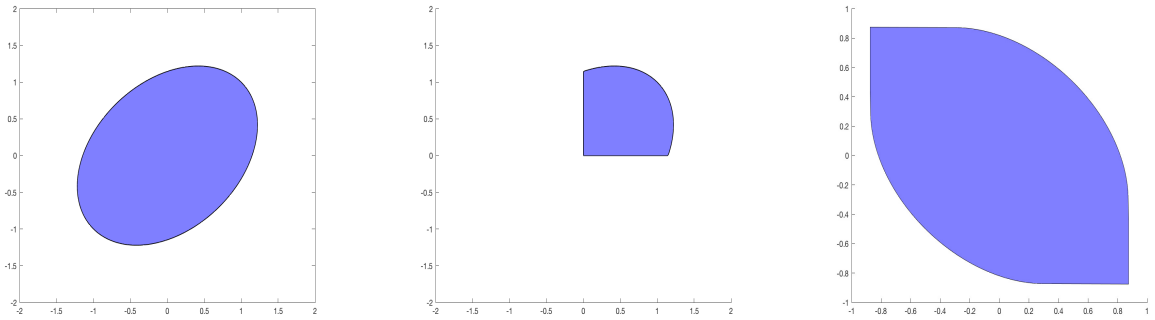
We show that $p_{\beta\to 0}^*$ is the gauge function of the convex hull of $\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}$ where $\mathcal{Q}_\mathbf{X} := \{\phi(\mathbf{X}\mathbf{w}^{(1)}) : \mathbf{w}^{(1)} \in \mathcal{B}_2\}$ (see Appendix N), i.e.,

$$p_{\beta\to 0}^* = \inf_{t: t\geq 0} \; t \text{ s.t. } \mathbf{y} \in t \, \text{Conv}\{\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}\} \,,$$

which we call *Neural gauge* due to the connection to the minimum-norm interpolation problem. Using classical polar gauge duality (see e.g. [60], it holds that

$$p_{\beta\to 0}^* = \max \mathbf{y}^T \mathbf{z} \text{ s.t. } \mathbf{z} \in (\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ \,, \tag{36}$$

where $(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ$ is the polar of the set $\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}$. Therefore, evaluating the support function of this polar set is equivalent to solving the neural gauge problem, i.e., minimum-norm interpolation $p_{\beta\to 0}^*$. These sets are illustrated in Figure 14. Note that the polar set $(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ$ is always convex (see Figure 14c), which also appears in the dual problem (34) as a constraint. In particular, $\lim_{\beta\to 0} \beta^{-1} d^*$ is equal to the support function. Our



(a) Ellipsoidal set: $\{\mathbf{X}\mathbf{w}^{(1)} : \mathbf{w}^{(1)} \in \mathbb{R}^d, \|\mathbf{w}^{(1)}\|_2 \leq 1\}$

(b) Rectified ellipsoidal set $\mathcal{Q}_{\mathbf{X}}$: $\{\phi(\mathbf{X}\mathbf{w}^{(1)}) : \mathbf{w}^{(1)} \in \mathbb{R}^d, \|\mathbf{w}^{(1)}\|_2 \leq 1\}$

(c) Polar set $(\mathcal{Q}_{\mathbf{X}} \cup -\mathcal{Q}_{\mathbf{X}})^\circ$: $\{\mathbf{v} : |\mathbf{v}^T\mathbf{w}| \leq 1, \forall \mathbf{w} \in \mathcal{Q}_{\mathbf{X}}\}$

Fig. 14: Sets involved in the construction of the Neural Gauge. Ellipsoidal set, rectified ellipsoid $\mathcal{Q}_{\mathbf{X}}$ and the polar of $\mathcal{Q}_{\mathbf{X}} \cup -\mathcal{Q}_{\mathbf{X}}$.

finite dimensional convex program leverages the convexity and an efficient description of this set as we discuss next.

Semi-infinite strong gauge duality Now we prove strong duality for (36). We invoke the semi-infinite optimality conditions for the dual (36), in particular we apply Theorem 7.2 of [36] and use the standard notation therein. We first define the set

$$\mathbf{K} = \mathbf{cone}\left\{ \begin{pmatrix} s\,\phi(\mathbf{X}\mathbf{w}^{(1)}) \\ 1 \end{pmatrix}, \mathbf{w}^{(1)} \in \mathcal{B}_2, s \in \{-1, +1\}; \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

Note that $\mathbf{K}$ is the union of finitely many convex closed sets, since $\phi(\mathbf{X}\mathbf{w}^{(1)})$ can be expressed as the union of finitely many convex closed sets. Therefore the set $\mathbf{K}$ is closed. By Theorem 5.3 of [36], this implies that the set of constraints in (37) forms a Farkas-Minkowski system. By Theorem 8.4 of [36], primal and dual values are equal, given that the system is consistent. Moreover, the system is discretizable, i.e., there exists a sequence of problems with finitely many constraints whose optimal values approach to the optimal value of (37). □

APPENDIX D
PROOF OF THEOREM II.1 AND PROPOSITION II.1

We now prove the main result for two-layer networks. We start with the dual representation derived in Section II-B

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} |\mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)})| \leq \beta\,. \tag{37}$$

Note that the constraint (37) can be represented as

$$\left\{\mathbf{v} : \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)}) \leq \beta\right\} \bigcap \left\{\mathbf{v} : \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} -\mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)}) \leq \beta\right\}.$$

We now focus on a single-sided dual constraint

$$\max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)}) \leq \beta, \tag{38}$$

by considering hyperplane arrangements and a convex duality argument over each partition. We first partition $\mathbb{R}^d$ into the following subsets

$$P_S := \{\mathbf{w}^{(1)} \,:\, \mathbf{x}_i^T\mathbf{w}^{(1)} \geq 0, \forall i \in S,\ \mathbf{x}_j^T\mathbf{w}^{(1)} \leq 0, \forall j \in S^c\}.$$

Let $\mathcal{H}$ be the set of all hyperplane arrangement patterns for the matrix $\mathbf{X}$, defined as the following set

$$\mathcal{H} = \bigcup \left\{ \{\text{sign}(\mathbf{X}\mathbf{w}^{(1)})\} \,:\, \mathbf{w}^{(1)} \in \mathbb{R}^d \right\}.$$

It is obvious that the set $\mathcal{H}$ is bounded, i.e., $\exists N_H \in \mathbb{N} < \infty$ such that $|\mathcal{H}| \leq N_H$. We next define an alternative representation of the sign patterns in $\mathcal{H}$, which is the collection of sets that correspond to positive signs for each element in $\mathcal{H}$. More precisely, let

$$\mathcal{S} := \left\{ \{\cup_{h_i=1}\{i\}\} \,:\, \mathbf{h} \in \mathcal{H} \right\}.$$

We also define a new diagonal matrix $\hat{\mathbf{D}}(S) \in \mathbb{R}^{n\times n}$ as

$$\hat{\mathbf{D}}(S)_{ii} := \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases},$$

where $S \in \mathcal{S}$.

Note that $\hat{\mathbf{D}}(S^c) = \mathbf{I}_n - \hat{\mathbf{D}}(S)$, since S^c is the complement of the set S. With this notation, we can represent P_S as

$$\begin{aligned} P_S &= \{\mathbf{w}^{(1)} \,:\, \hat{\mathbf{D}}(S)\mathbf{X}\mathbf{w}^{(1)} \geq 0,\ (\mathbf{I}_n - \hat{\mathbf{D}}(S))\mathbf{X}\mathbf{w}^{(1)} \leq 0\} \\ &= \{\mathbf{w}^{(1)} \,:\, (2\hat{\mathbf{D}}(S) - \mathbf{I}_n)\mathbf{X}\mathbf{w}^{(1)} \geq 0\}. \end{aligned}$$

We now express the maximization in the dual constraint in (38) over all possible hyperplane arrangement patterns as

$$\begin{aligned} \max_{\mathbf{w}^{(1)}\in\mathcal{B}_2} \mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)}) = &\max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\substack{\mathbf{w}^{(1)}\in\mathcal{B}_2\\ \mathbf{x}_i^T\mathbf{w}^{(1)}\geq 0\ \forall i\in S\\ \mathbf{x}_j^T\mathbf{w}^{(1)}\leq 0\ \forall j\in S^c}} \mathbf{v}^T\mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)} \\ = &\max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap P_S} \mathbf{v}^T\mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)}, \end{aligned}$$

where

$$\mathbf{D}(S)_{ii} := \begin{cases} 1 & \text{if } i \in S \\ \kappa & \text{otherwise} \end{cases}.$$

We also note that since $\kappa < 0.5$, P_S can be equivalently represented as

$$P_S = \{\mathbf{w}^{(1)} \,:\, (2\mathbf{D}(S) - \mathbf{I}_n)\mathbf{w}^{(1)} \geq 0\}.$$

Enumerating all hyperplane arrangements $\mathcal{H}$, or equivalently $\mathcal{S}$, we index them in an arbitrary order via $i \in [|\mathcal{S}|]$. We denote $P = |\mathcal{S}|$. Hence, $S_1, \ldots, S_P \in \mathcal{S}$ is the list of all P elements of $\mathcal{S}$. Next we use the strong duality result from Lemma D.1 for the inner maximization problem. The dual constraint (38) can be represented as

$$\begin{aligned} (38) &\iff \forall i \in [P],\ \min_{\substack{\alpha,\beta\in\mathbb{R}^n\\ \alpha,\beta\geq 0}} \|\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}+\boldsymbol{\gamma}) - \mathbf{X}^T\boldsymbol{\gamma}\|_2 \leq \beta \\ &\iff \forall i \in [P],\ \exists \boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i \in \mathbb{R}^n \text{ s.t. } \quad \boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i \geq 0,\ \|\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}_i+\boldsymbol{\gamma}_i) - \mathbf{X}^T\boldsymbol{\gamma}_i\|_2 \leq \beta. \end{aligned}$$

Therefore, recalling the two-sided constraint in (37), we can represent the dual optimization problem in (37) as a finite dimensional convex optimization problem with variables $\mathbf{v} \in \mathbb{R}^n, \boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i, \boldsymbol{\alpha}'_i, \boldsymbol{\gamma}'_i \in \mathbb{R}^n, \forall i \in [P]$, and $2P$ second order cone constraints as follows

$$\begin{aligned}
\max_{\substack{\mathbf{v}\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\geq 0,\,\forall i\in[P] \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\geq 0,\,\forall i\in[P]}} \quad & -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \|\mathbf{X}^T\mathbf{D}(S_1)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_1)(\boldsymbol{\alpha}_1+\boldsymbol{\gamma}_1) - \mathbf{X}^T\boldsymbol{\gamma}_1\|_2 \leq \beta \\
& \qquad \vdots \\
& \|\mathbf{X}^T\mathbf{D}(S_P)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_P)(\boldsymbol{\alpha}_P+\boldsymbol{\gamma}_P) - \mathbf{X}^T\boldsymbol{\gamma}_P\|_2 \leq \beta \\
& \|-\mathbf{X}^T\mathbf{D}(S_1)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_1)(\boldsymbol{\alpha}'_1+\boldsymbol{\gamma}'_1) - \mathbf{X}^T\boldsymbol{\gamma}'_1\|_2 \leq \beta \\
& \qquad \vdots \\
& \|-\mathbf{X}^T\mathbf{D}(S_P)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_P)(\boldsymbol{\alpha}'_P+\boldsymbol{\gamma}'_P) - \mathbf{X}^T\boldsymbol{\gamma}'_P\|_2 \leq \beta.
\end{aligned}$$

The above problem can be represented as a standard finite dimensional second order cone program. Note that the particular choice of parameters $\mathbf{v} = \boldsymbol{\alpha}_i = \boldsymbol{\gamma}_i = \boldsymbol{\alpha}'_i = \boldsymbol{\gamma}'_i = 0, \forall i \in [P]$, are strictly feasible in the above constraints as long as $\beta > 0$. Therefore Slater's condition and consequently strong duality holds [12]. The dual problem (37) can be written as

$$\begin{aligned}
\min_{\substack{\boldsymbol{\lambda},\boldsymbol{\lambda}'\in\mathbb{R}^P \\ \boldsymbol{\lambda},\boldsymbol{\lambda}'\geq 0}} \quad \max_{\substack{\mathbf{v}\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\geq 0,\,\forall i \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\geq 0,\,\forall i}} \quad & -\mathcal{L}^*(\mathbf{v}) + \sum_{i=1}^P \lambda_i\big(\beta - \|\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}_i+\boldsymbol{\gamma}_i) - \mathbf{X}^T\boldsymbol{\gamma}_i\|_2\big) \\
& + \sum_{i=1}^P \lambda'_i\big(\beta - \|-\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}'_i+\boldsymbol{\gamma}'_i) - \mathbf{X}^T\boldsymbol{\gamma}'_i\|_2\big).
\end{aligned}$$

Next, we introduce variables $\mathbf{r}_1, \ldots, \mathbf{r}_P, \mathbf{r}'_1, \ldots, \mathbf{r}'_P \in \mathbb{R}^d$ and represent the dual problem (37) as

$$\begin{aligned}
\min_{\substack{\boldsymbol{\lambda},\boldsymbol{\lambda}'\in\mathbb{R}^P \\ \boldsymbol{\lambda},\boldsymbol{\lambda}'\geq 0}} \quad \max_{\substack{\mathbf{v}\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\geq 0,\,\forall i \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\geq 0,\,\forall i}} \quad \min_{\substack{\mathbf{r}_i\in\mathbb{R}^d,\,\|\mathbf{r}_i\|_2\leq 1 \\ \mathbf{r}'_i\in\mathbb{R}^d,\,\|\mathbf{r}'_i\|_2\leq 1 \\ \forall i}} \quad & -\mathcal{L}^*(\mathbf{v}) + \sum_{i=1}^P \boldsymbol{\lambda}_i\big(\beta + \mathbf{r}_i^T\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{r}_i^T\mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}_i+\boldsymbol{\gamma}_i) - \mathbf{r}_i^T\mathbf{X}^T\boldsymbol{\gamma}_i\big) \\
& + \sum_{i=1}^P \lambda'_i\big(\beta - \mathbf{r}_i'^T\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{r}_i'^T\mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}'_i+\boldsymbol{\gamma}'_i) - \mathbf{r}_i'^T\mathbf{X}^T\boldsymbol{\gamma}'_i\big).
\end{aligned}$$

We note that the objective is concave in $\mathbf{v}, \boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i$ and convex in $\mathbf{r}_i, \mathbf{r}'_i$, $\forall i \in [P]$. Moreover, the constraint sets $\|\mathbf{r}_i\|_2 \leq 1$, $\|\mathbf{r}'_i\|_2 \leq 1$, $\forall i$ are convex and compact. Invoking Sion's minimax theorem [69] for the inner $\max\min$ problem, we may express the strong dual of the problem (37) as

$$\begin{aligned}
\min_{\substack{\boldsymbol{\lambda},\boldsymbol{\lambda}'\in\mathbb{R}^P \\ \boldsymbol{\lambda},\boldsymbol{\lambda}'\geq 0}} \quad \min_{\substack{\mathbf{r}_i\in\mathbb{R}^d,\,\|\mathbf{r}_i\|_2\leq 1 \\ \mathbf{r}'_i\in\mathbb{R}^d,\,\|\mathbf{r}'_i\|_2\leq 1}} \quad \max_{\substack{\mathbf{v}\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}_i,\boldsymbol{\gamma}_i\geq 0,\,\forall i \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\in\mathbb{R}^n \\ \boldsymbol{\alpha}'_i,\boldsymbol{\gamma}'_i\geq 0,\,\forall i}} \quad & -\mathcal{L}^*(\mathbf{v}) + \sum_{i=1}^P \lambda_i\big(\beta + \mathbf{r}_i^T\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{r}_i^T\mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}_i+\boldsymbol{\gamma}_i) - \mathbf{r}_i^T\mathbf{X}^T\boldsymbol{\gamma}_i\big) \\
& + \sum_{i=1}^P \lambda'_i\big(\beta - \mathbf{r}_i'^T\mathbf{X}^T\mathbf{D}(S_i)\mathbf{v} + \mathbf{r}_i'^T\mathbf{X}^T\hat{\mathbf{D}}(S_i)(\boldsymbol{\alpha}'_i+\boldsymbol{\gamma}'_i) - \mathbf{r}_i'^T\mathbf{X}^T\boldsymbol{\gamma}'_i\big).
\end{aligned}$$

Computing the maximum with respect to $\mathbf{v}, \boldsymbol{\alpha}_i, \boldsymbol{\gamma}_i, \boldsymbol{\alpha}'_i, \boldsymbol{\gamma}'_i$, $\forall i \in [P]$, analytically we obtain the strong dual to (37) as

$$\min_{\substack{\boldsymbol{\lambda},\boldsymbol{\lambda}' \in \mathbb{R}^P \\ \boldsymbol{\lambda},\boldsymbol{\lambda}' \geq 0}} \quad \min_{\substack{\mathbf{r}_i \in \mathbb{R}^d, \|\mathbf{r}_i\|_2 \leq 1 \\ \mathbf{r}'_i \in \mathbb{R}^d, \|\mathbf{r}'_i\|_2 \leq 1 \\ (2\hat{\mathbf{D}}(S_i) - \mathbf{I}_n)\mathbf{X}\mathbf{r}_i \geq 0 \\ (2\hat{\mathbf{D}}(S_i) - \mathbf{I}_n)\mathbf{X}\mathbf{r}'_i \geq 0}} \mathcal{L}\left(\sum_{i=1}^P \lambda_i \mathbf{D}(S_i)\mathbf{X}\mathbf{r}'_i - \lambda'_i \mathbf{D}(S_i)\mathbf{X}\mathbf{r}_i, \mathbf{y}\right) + \beta \sum_{i=1}^P (\lambda_i + \lambda'_i).$$

Now we apply a change of variables and define $\mathbf{w}_i = \lambda_i \mathbf{r}_i$ and $\mathbf{w}'_i = \lambda'_i \mathbf{r}'_i$, $\forall i \in [P]$. Note that we can take $\mathbf{r}_i = 0$ when $\lambda_i = 0$ without changing the optimal value. We obtain

$$\min_{\substack{\mathbf{w}_i, \mathbf{w}'_i \in P_{S_i} \\ \|\mathbf{w}_i\|_2 \leq \lambda_i \\ \|\mathbf{w}'_i\|_2 \leq \lambda'_i \\ \boldsymbol{\lambda},\boldsymbol{\lambda}' \geq 0}} \mathcal{L}\left(\sum_{i=1}^P \mathbf{D}(S_i)\mathbf{X}(\mathbf{w}'^*_i - \mathbf{w}_i), \mathbf{y}\right) + \beta \sum_{i=1}^P (\lambda_i + \lambda'_i).$$

The variables λ_i, λ'_i, $\forall i \in [P]$ can be eliminated since $\lambda_i = \|\mathbf{w}_i\|_2$ and $\lambda'_i = \|\mathbf{w}'^*_i\|_2$ are feasible and optimal. Plugging in these expressions, we get

$$\min_{\mathbf{w}_i, \mathbf{w}'_i \in P_{S_i}} \mathcal{L}\left(\sum_{i=1}^P \mathbf{D}(S_i)\mathbf{X}(\mathbf{w}'_i - \mathbf{w}_i), \mathbf{y}\right) + \beta \sum_{i=1}^P (\|\mathbf{w}_i\|_2 + \|\mathbf{w}'_i\|_2),$$

which is identical to (7), and proves that the objective values are equal. Constructing $\{\mathbf{w}_j^{(1)^*}, w_j^{(2)^*}\}_{j=1}^{m^*}$ as stated in the Proposition II.1, and plugging in (3), we obtain the value

$$\begin{aligned} p^* \leq \mathcal{L}\left(\sum_{j=1}^{m^*} \phi(\mathbf{X}\mathbf{w}_j^{(1)^*}) w_j^{(2)^*}, \mathbf{y}\right) \frac{1}{2} + \frac{\beta}{2} \sum_{i=1, \mathbf{w}'^*_i \neq 0}^P \left(\left\| \frac{\mathbf{w}'^*_i}{\sqrt{\|\mathbf{w}'^*_i\|_2}} \right\|_2^2 + \left\| \sqrt{\|\mathbf{w}'^*_i\|_2} \right\|_2^2 \right) \\ + \frac{\beta}{2} \sum_{i=1, \mathbf{w}^*_i \neq 0}^P \left(\left\| \frac{\mathbf{w}^*_i}{\sqrt{\|\mathbf{w}^*_i\|_2}} \right\|_2^2 + \left\| \sqrt{\|\mathbf{w}^*_i\|_2} \right\|_2^2 \right), \end{aligned}$$

which is identical to the objective value of the convex program (7). Since the value of the convex program is equal to the value of it's dual d^* in (37), and $p^* \geq d^*$, we conclude that $p^* = d^*$, which is equal to the value of the convex program (7) achieved by the prescribed parameters. □

Finite dimensional strong duality results for Theorem II.1

Lemma D.1. *Suppose $\mathbf{D}(S)$, $\hat{\mathbf{D}}(S)$, $\hat{\mathbf{D}}(S^c)$ are fixed diagonal matrices as described in the proof of Theorem II.1, and $\mathbf{X}$ is a fixed matrix. The dual of the convex optimization problem*

$$\max_{\substack{\mathbf{w}^{(1)} \in \mathcal{B}_2 \\ \hat{\mathbf{D}}(S)\mathbf{X}\mathbf{w}^{(1)} \geq 0 \\ (\mathbf{I}_n - \hat{\mathbf{D}}(S^c))\mathbf{X}\mathbf{w}^{(1)} \leq 0}} \mathbf{v}^T \mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)}$$

is given by

$$\min_{\substack{\boldsymbol{\alpha},\boldsymbol{\gamma} \in \mathbb{R}^n \\ \boldsymbol{\alpha},\boldsymbol{\gamma} \geq 0}} \|\mathbf{X}^T \mathbf{D}(S)\mathbf{v} + \mathbf{X}^T \hat{\mathbf{D}}(S)(\boldsymbol{\alpha} + \boldsymbol{\gamma}) - \mathbf{X}^T \boldsymbol{\gamma}\|_2$$

and strong duality holds.

Note that the linear inequality constraints specify valid hyperplane arrangements. Then there exists strictly feasible points in the constraints of the maximization problem. Standard finite second order cone programming duality implies that strong duality holds [13] and the dual is as specified. □

D-A Proof of Corollary II.1

To derive the convex program for a network with bias term, we first define a new variable by concatenating the bias and weights as $\hat{\mathbf{w}}_j^{(1)} := [\mathbf{w}_j^{(1)}; b_j]$. Then the rest of the derivations directly follows from the proof of Theorem II.1 when we replace $\mathbf{w}^{(1)}$ with $\hat{\mathbf{w}}^{(1)} = [\mathbf{w}^{(1)}; b]$. □

APPENDIX E
PROOF OF THEOREM II.2

Here, we first prove that Clarke stationary points of the nonconvex training problem of two-layer networks found by first order methods such as SGD/GD correspond to the global optimum of a version of our convex program based trichotomy arrangements. We then generalize this result to our convex program with standard dichotomy arrangements in (7).

Clarke Stationary Points and Convex Program with Trichotomies

We first note that [75] provided a similar result however their analysis is valid only for ReLU activations and convex programs with trichotomy arrangements. On the other hand, our proof extends to arbtirary piecewise linear activations and in the next section we generalize this result to our convex program with standard dichotomy arrangements (or diagonal matrices $\mathbf{D}$).

For piecewise linear activations, we define the hyperplane arrangements matrices $\mathbf{D}$ based on dichotomies as

$$\mathbf{D}_{ii} := \begin{cases} 1 & \text{if } \mathbf{x}_i^T \mathbf{w}^{(1)} \geq 0 \\ \kappa & \text{otherwise} \end{cases}, \tag{39}$$

whereas trichotomy arrangement matrix $\mathbf{T}$ is defined as

$$\mathbf{T}_{ii} := \begin{cases} 1 & \text{if } \mathbf{x}_i^T \mathbf{w}^{(1)} > 0 \\ 0 & \text{if } \mathbf{x}_i^T \mathbf{w}^{(1)} = 0 \\ -1 & \text{otherwise} \end{cases}. \tag{40}$$

Due to the nondifferentiability of the piecewise linear activations, we next review the definition of the Clarke subdifferential [19] of a given function f. Let $D \subset \mathbb{R}^d$ be the set of points at which f is differentiable. We assume that D has (Lebesgue) measure 1, meaning that f is differentiable *almost everywhere*. The Clarke subdifferential of f at $\mathbf{x}$ is then defined as

$$\partial_C f(\mathbf{x}) = \text{Conv}\left\{ \lim_{k\to\infty} \nabla f(\mathbf{x}_k) \mid \lim_{k\to\infty} \mathbf{x}_k \to \mathbf{x},\, \mathbf{x}_k \in D \right\}.$$

Then, we say that $\mathbf{x} \in \mathbb{R}^d$ is Clarke stationary with respect to f if $\mathbf{0} \in \partial_C f(\mathbf{x})$.

Based on the definition above, we now consider a nonconvex neural networks model with piecewise linear activations, i.e., $f_\theta(\mathbf{X}) = \sum_{j=1}^m \left(\mathbf{X}\mathbf{w}_j^{(1)}\right)_+ w_j^{(2)}$, and aim to optimize the parameters through the weight decay regularized objective function in (3). From the definition of Clarke stationary point, for $j \in [m]$ with $\mathbf{w}_j^{(1)} \neq \mathbf{0}$, we have

$$\begin{aligned} -\beta \mathbf{w}_j^{(1)} &\in \partial_{\mathbf{w}_j^{(1)}} \mathcal{L}\left(\sum_{j=1}^m \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big) w_j^{(2)}, \mathbf{y} \right), \\ -\beta w_j^{(2)} &= \mathbf{g}^T \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big) \end{aligned} \tag{41}$$

where

$$\mathbf{g} := \nabla_f \mathcal{L}\Big(\underbrace{\sum_{j=1}^m \phi\big(\mathbf{X}\mathbf{w}_j^{(1)}\big) w_j^{(2)}}_{=f}, \mathbf{y} \Big). \tag{42}$$

We note that (41) formulates the stationarity conditions of the nonconvex training problem and [47] proved that running GD to minimize this objective converges to a point, where these stationarity conditions are satified. Then, the first stationary condition in (41) implies that there exists $\boldsymbol{\delta}_j \in [-\kappa, 1]^n$ such that

$$-\beta \mathbf{w}_j^{(1)} = w_j^{(2)} \left(\mathbf{X}^T \mathbf{D}_j \mathbf{g} + \mathbf{X}^T \mathbf{S}_j \mathrm{diag}(\boldsymbol{\delta}_j) \mathbf{g} \right),$$

where $\mathbf{D}_j$ is defined in (39) and $\mathbf{S}_j = \mathrm{diag}(\mathbb{1}[\mathbf{X}\mathbf{w}_j^{(1)} = 0])$. Assuming $\mathbf{w}_j^{(1)} \neq \mathbf{0}$ and $w_j^{(2)} \neq 0$, the equality above implies that

$$-\beta \frac{\mathbf{w}_j^{(1)}}{w_j^{(2)}} = \mathbf{X}^T \mathbf{D}_j \mathbf{g} + \mathbf{X}^T \mathbf{S}_j \mathrm{diag}(\boldsymbol{\delta}_j) \mathbf{g}. \tag{43}$$

Additionally, from the second stationary condition in (41), we have

$$\begin{aligned} -\beta w_j^{(2)} &= \mathbf{g}^T \mathbf{D}_j \mathbf{X} \mathbf{w}_j^{(1)} \\ &= {\mathbf{w}_j^{(1)}}^T \mathbf{X}^T \mathbf{D}_j \mathbf{g} \\ &= {\mathbf{w}_j^{(1)}}^T \left(\mathbf{X}^T \mathbf{D}_j \mathbf{g} + \mathbf{X}^T \mathbf{S}_j \mathrm{diag}(\boldsymbol{\delta}_j) \mathbf{g} \right) \\ &= {\mathbf{w}_j^{(1)}}^T \left(-\beta \frac{\mathbf{w}_j^{(1)}}{w_j^{(2)}} \right) \\ &= -\beta \frac{\|\mathbf{w}_j^{(1)}\|_2^2}{w_j^{(2)}}. \end{aligned} \tag{44}$$

Thus, we have $|w_j^{(2)}| = \|\mathbf{w}_j^{(1)}\|_2$ and from (43)

$$\|\mathbf{X}^T \mathbf{D}_j \mathbf{g} + \mathbf{X}^T \mathbf{S}_j \mathrm{diag}(\boldsymbol{\delta}_j) \mathbf{g}\|_2 = \beta. \tag{45}$$

Now, given the following subsampled convex program with trichotomy arrangement

$$\min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \mathcal{L} \left(\sum_{i=1}^{\tilde{P}} \phi(\mathbf{T}_i) \mathbf{X} (\mathbf{w}_i - \mathbf{w}_{i+\tilde{P}}), \mathbf{y} \right) + \beta \sum_{i=1}^{2\tilde{P}} \|\mathbf{w}_i\|_2, \tag{46}$$

where $\mathcal{C}(\mathbf{X})$ are convex constraint enforcing weights to satisfy the trichotomy arrangement patterns in (40), the KKT conditions are given by: for $i \in [\tilde{P}]$, there exists $\boldsymbol{\zeta}_i \geq 0$ and $\boldsymbol{\xi}_i$

$$\begin{aligned} &\mathbf{X}^T \left(\phi(\mathbf{T}_i) \mathbf{v} + \mathbf{T}_i \boldsymbol{\zeta}_i + \tilde{\mathbf{S}}_i \boldsymbol{\xi}_i \right) + \beta \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2} = 0, && \text{if } \mathbf{w}_i \neq \mathbf{0} \\ &\left\| \mathbf{X}^T \left(\phi(\mathbf{T}_i) \mathbf{v} + \mathbf{T}_i \boldsymbol{\zeta}_i + \tilde{\mathbf{S}}_i \boldsymbol{\xi}_i \right) \right\|_2 \leq \beta, && \text{if } \mathbf{w}_i = \mathbf{0} \\ &\mathbf{X}^T \left(-\phi(\mathbf{T}_i) \mathbf{v} + \mathbf{T}_i \boldsymbol{\zeta}_{i+\tilde{P}} + \tilde{\mathbf{S}}_i \boldsymbol{\xi}_{i+\tilde{P}} \right) + \beta \frac{\mathbf{w}_{i+\tilde{P}}}{\|\mathbf{w}_{i+\tilde{P}}\|_2} = 0, && \text{if } \mathbf{w}_{i+\tilde{P}} \neq \mathbf{0} \\ &\left\| \mathbf{X}^T \left(-\phi(\mathbf{T}_i) \mathbf{v} + \mathbf{T}_i \boldsymbol{\zeta}_{i+\tilde{P}} + \tilde{\mathbf{S}}_i \boldsymbol{\xi}_{i+\tilde{P}} \right) \right\|_2 \leq \beta, && \text{if } \mathbf{w}_{i+\tilde{P}} = \mathbf{0} \end{aligned} \tag{47}$$

where $\tilde{\mathbf{S}}_i$ is a diagonal matrix satisfying that $\tilde{\mathbf{S}}_{jj} = 1$ if $\mathbf{T}_{i,jj} = 0$ and $\tilde{\mathbf{S}}_{jj} = 0$ otherwise. Also, $\mathbf{v} \in \mathbb{R}^n$ is defined as $\mathbf{v} = \nabla \mathcal{L} \left(\sum_{i=1}^{\tilde{P}} \phi(\mathbf{T}_i) \mathbf{X} (\mathbf{w}_i - \mathbf{w}_{i+\tilde{P}}), \mathbf{y} \right)$.

Due to the one to one mapping in Proposition II.1, we have $\mathbf{g} = \mathbf{v}$. Also, taking $\boldsymbol{\zeta}_i = 0$, $\boldsymbol{\xi} = \mathrm{diag}(\boldsymbol{\delta}_j)\mathbf{g}$, $\boldsymbol{\zeta}_{i+\tilde{P}} = 0$, $\boldsymbol{\xi}_{i+\tilde{P}} = -\mathrm{diag}(\boldsymbol{\delta}_j)\mathbf{g}$, $\mathbf{D}_j = \phi(\mathbf{T}_i)$, and $\mathbf{S}_j = \tilde{\mathbf{S}}_i$ satisfies the KKT condition in (47). Therefore, the Clarke stationary points of the nonconvex training objective in (41) is a global optimum of the subsampled convex program in (46). □

Extension to Our Convex Program with Dichotomies

In order to establish a similar proof for our hyperplane arrangement matrices based on dichotomies, here, we show

that the optimal solutions to the subsampled convex programs based on dichotomies and trichomoties coincide, i.e., the maximizers of the dual constraints for each case are the same.

We start with stating the dual constraint (dc) of (9) for each case with $\tilde{P}$ sampled arrangements as follows

$$\begin{aligned} dc &:= \max_{k\in\tilde{P}} \max_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap\mathcal{C}_k} \left|\mathbf{v}^T\mathbf{D}_k\mathbf{X}\mathbf{w}^{(1)}\right| \\ dc^t &:= \max_{k\in\tilde{P}} \max_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap\mathcal{C}_k^t} \left|\mathbf{v}^T\phi\big(\mathbf{T}_k\big)\mathbf{X}\mathbf{w}^{(1)}\right| \end{aligned}, \tag{48}$$

where

$$\begin{aligned} \mathcal{C}_k &:= \{\mathbf{w}\in\mathbb{R}^d : \mathbf{x}_i^T\mathbf{w}\geq 0, \forall i\in\{i:\mathbf{D}_{k,ii}=1\}, \mathbf{x}_i^T\mathbf{w}<0, \text{otherwise}\} \\ \mathcal{C}_k^t &:= \{\mathbf{w}\in\mathbb{R}^d : \mathbf{T}_{k,ii}\mathbf{x}_i^T\mathbf{w}>0, \forall i\in\{i:\mathbf{T}_{k,ii}\in\{\pm 1\}\}, \mathbf{x}_i^T\mathbf{w}=0, \text{otherwise}\} \end{aligned}. \tag{49}$$

We remark that $\mathcal{C}_k$ is a relaxation of $\mathcal{C}_k^t$ since $\mathcal{C}_k^t$ enforces certain entries to be exactly zero due to trichotomies.

We first note that if there are no zero entries in the optimal $\mathbf{T}_k$ for dc^t then the same solution will be optimal for dc since both problems will be exactly identical in that case. If the optimal $\mathbf{T}_k$ has a zero entry then we need to check if it matches to the solution of dc. To do so, we need to show that

$$\operatorname*{argmax}_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap\mathcal{C}_k^t} \left|\mathbf{v}^T\phi\big(\mathbf{T}_k\big)\mathbf{X}\mathbf{w}^{(1)}\right| \in \left\{ \operatorname*{argmax}_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap\mathcal{C}_i} \left|\mathbf{v}^T\mathbf{D}_l\mathbf{X}\mathbf{w}^{(1)}\right|, \operatorname*{argmax}_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap\mathcal{C}_j} \left|\mathbf{v}^T\mathbf{D}_j\mathbf{X}\mathbf{w}^{(1)}\right| \right\}, \tag{50}$$

where $\mathbf{D}_l$ and $\mathbf{D}_j$ are dichotomies that include the zero index in $\mathbf{T}_k$, say $\mathbf{x}_i^T\mathbf{w}^{(1)}=0$, in the nonnegative ($\mathbf{x}_i^T\mathbf{w}^{(1)}\geq 0$) and nonpositive ($\mathbf{x}_i^T\mathbf{w}^{(1)}\leq 0$) sides of the hyperplane. Other than that, all the entries of $\mathbf{D}_l, \mathbf{D}_j$, and $\mathbf{T}_k$ are the same, i.e., $\forall i\in[n]$,

$$\mathbf{D}_{l,ii} := \begin{cases} \phi\big(\mathbf{T}_{k,ii}\big) & \text{if } \mathbf{x}_i^T\mathbf{w}^{(1)}\neq 0 \\ 1 & \text{otherwise} \end{cases}, \quad \mathbf{D}_{j,ii} := \begin{cases} \phi\big(\mathbf{T}_{k,ii}\big) & \text{if } \mathbf{x}_i^T\mathbf{w}^{(1)}\neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

If one of the dichotomy problems in (50) achieves the same optimum when $\mathbf{x}_i^T\mathbf{w}^{(1)}=0$, then we can claim that there is an optimal dichotomy arrangement corresponding to the optimal trichotomy arrangement $\mathbf{T}_k$. If not, then this means that both dichotomy problems in (50) achieve the optimum when $\mathbf{x}_i^T\mathbf{w}^{(1)}>0$ and $\mathbf{x}_i^T\mathbf{w}^{(1)}<0$, respectively. However, this cannot be true. To illustrate this, let us first denote the solutions to each problem as $\mathbf{w}_1^{(1)}$ and $\mathbf{w}_2^{(1)}$ such that $\mathbf{x}_i^T\mathbf{w}_1^{(1)}>0$ and $\mathbf{x}_i^T\mathbf{w}_2^{(1)}<0$. Since the objective function is linear, we can find a linear interpolation between $\mathbf{w}_1^{(1)}$ and $\mathbf{w}_2^{(1)}$ as $\mathbf{w}_0^{(1)} := t\mathbf{w}_1^{(1)} + (1-t)\mathbf{w}_2^{(1)}$, where $t\in[0,1]$, such that $\mathbf{x}_i^T\mathbf{w}_0^{(1)}=0$. Then, since $h(\mathbf{w}^{(1)}) := \mathbf{x}_i^T\mathbf{w}^{(1)}$ is a linear function, the interpolation between them cannot achieve a value that is strictly less than both, i.e., $h(\mathbf{w}_0^{(1)}) \geq \min\{h(\mathbf{w}_1^{(1)}), h(\mathbf{w}_2^{(1)})\}$. Therefore, we have a contradiction due to the assumption that both dichotomies achieves optimal solution without zero entries. This concludes the proof. □

APPENDIX F
PROOF OF THEOREM III.1

Lemma F.1. *Given an L-Lipschitz convex loss $\mathcal{L}(\cdot,\mathbf{y})$ and an R-Lipschitz activation function $\phi(\cdot)$, consider the following nonconvex optimization problem with $\hat{\mathbf{X}}_k$*

$$(\hat{\mathbf{W}}^{(1)}, \hat{\mathbf{w}}^{(2)}) \in \operatorname*{argmin}_{\theta\in\Theta_s} \mathcal{L}(\phi\big(\hat{\mathbf{X}}_k\mathbf{W}^{(1)}\big)\mathbf{w}^{(2)}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)}\|_1$$

and the objective value with the original data $\mathbf{X}$ evaluated at any optimum $(\hat{\mathbf{W}}^{(1)}, \hat{\mathbf{w}}^{(2)})$

$$p_k := \mathcal{L}(\phi\big(\mathbf{X}\hat{\mathbf{W}}^{(1)}\big)\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + \beta\|\hat{\mathbf{w}}^{(2)}\|_1.$$

Then, we have the following approximation guarantee

$$p^* \leq p_k \leq p^*\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2.$$

Proof of Lemma F.1 We start with defining the optimal parameters for the original and rank-k approximation of the rescaled problem in (8) as

$$\begin{aligned}(\mathbf{W}^{(1)^*}, \mathbf{w}^{(2)^*}) &\in \operatorname*{argmin}_{\theta\in\Theta_s} \mathcal{L}(\phi(\mathbf{X}\mathbf{W}^{(1)})\mathbf{w}^{(2)}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)}\|_1 \\ (\hat{\mathbf{W}}^{(1)}, \hat{\mathbf{w}}^{(2)}) &\in \operatorname*{argmin}_{\theta\in\Theta_s} \mathcal{L}(\phi(\hat{\mathbf{X}}_k\mathbf{W}^{(1)})\mathbf{w}^{(2)}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)}\|_1\end{aligned} \tag{51}$$

and the objective value achieved by the parameters trained using $\hat{\mathbf{X}}_k$ as

$$p_k := \mathcal{L}(\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + \beta\|\hat{\mathbf{w}}^{(2)}\|_1.$$

Then, we have

$$\begin{aligned}p^* &= \mathcal{L}(\phi(\mathbf{X}\mathbf{W}^{(1)^*})\mathbf{w}^{(2)^*}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)^*}\|_1 \\ &\overset{(i)}{\leq} \mathcal{L}(\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + \beta\|\hat{\mathbf{w}}^{(2)}\|_1 = p_k \\ &\overset{(ii)}{\leq} \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + (\beta + LR\sigma_{k+1})\|\hat{\mathbf{w}}^{(2)}\|_1 \\ &\leq \left(\mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + \beta\|\hat{\mathbf{w}}^{(2)}\|_1\right)\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right) \\ &\overset{(iii)}{\leq} \left(\mathcal{L}(\phi(\hat{\mathbf{X}}_k\mathbf{W}^{(1)^*})\mathbf{w}^{(2)^*}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)^*}\|_1\right)\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right) \\ &\overset{(iv)}{\leq} \left(\mathcal{L}(\phi(\mathbf{X}\mathbf{W}^{(1)^*})\mathbf{w}^{(2)^*}, \mathbf{y}) + \beta\|\mathbf{w}^{(2)^*}\|_1\right)\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2 \\ &= p^*\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2,\end{aligned} \tag{52}$$

where (i) and (iii) follow from the optimality definitions of the original and approximated problems in (51). In addition, (ii) and (iv) follow from the relations below

$$\begin{aligned}\mathcal{L}(\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) &= \mathcal{L}(\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)} - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)} + \phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\overset{(1)}{\leq} \mathcal{L}(\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)} - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\overset{(2)}{\leq} L\left\|\phi(\mathbf{X}\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)} - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}\right\|_2 + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &= L\left\|\sum_{j=1}^m \left(\phi(\mathbf{X}\hat{\mathbf{w}}_j^{(1)}) - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{w}}_j^{(1)})\right)\hat{w}_j^{(2)}\right\|_2 + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\overset{(3)}{\leq} L\sum_{j=1}^m \left\|\phi(\mathbf{X}\hat{\mathbf{w}}_j^{(1)}) - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{w}}_j^{(1)})\right\|_2 \left|\hat{w}_j^{(2)}\right| + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\leq L\max_{j\in[m]}\left\|\phi(\mathbf{X}\hat{\mathbf{w}}_j^{(1)}) - \phi(\hat{\mathbf{X}}_k\hat{\mathbf{w}}_j^{(1)})\right\|_2 \|\hat{\mathbf{w}}^{(2)}\|_1 + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\overset{(4)}{\leq} LR\max_{j\in[m]}\|\hat{\mathbf{w}}_j^{(1)}\|_2\left\|\mathbf{X} - \hat{\mathbf{X}}_k\right\|_2\|\hat{\mathbf{w}}^{(2)}\|_1 + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}) \\ &\overset{(5)}{=} LR\sigma_{k+1}\|\hat{\mathbf{w}}^{(2)}\|_1 + \mathcal{L}(\phi(\hat{\mathbf{X}}_k\hat{\mathbf{W}}^{(1)})\hat{\mathbf{w}}^{(2)}, \mathbf{y}),\end{aligned}$$

where we use the convexity and L-Lipschitz property of the loss function, convexity of ℓ_2-norm, R-Lipschitz property of the activation, and $\max_j \|\hat{\mathbf{w}}_j^{(1)}\|_2 = 1$ from the rescaling in Lemma II.1 for $(1), (2), (3), (4)$, and (5), respectively.

Based on (52), we have

$$p^* \leq p_k \leq p^*\left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2.$$

□

Based on the approximation bound provided by Lemma F.1, we next show that the complexity of solving the convex reformulations can be reduced via rank-k approximations. Note that due to the rank-k data matrix $\hat{\mathbf{X}}_k$, the number of hyperplane arrangements in the corresponding convex formulation (7) is significantly reduced. We formalize this in the next corollary.

We first restate the exact convex program as follows

$$\begin{aligned} p^* &= \min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \mathcal{L}\left(\mathcal{A}(\mathbf{X})\mathbf{w}, \mathbf{y}\right) + \beta \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2 \\ &= \min_{\mathbf{w} \in \mathcal{C}(\mathbf{X})} \mathcal{L}\left(\sum_{i=1}^{P} \mathbf{D}_i \mathbf{X}(\mathbf{w}_i - \mathbf{w}_{i+P}), \mathbf{y}\right) + \beta \sum_{i=1}^{2P} \|\mathbf{w}_i\|_2. \end{aligned}$$

In addition to this, we define two rank-k approximated versions based on Theorem III.1

$$\hat{\mathbf{w}}^{(k)} \in \underset{\mathbf{w} \in \mathcal{C}(\hat{\mathbf{X}}_k)}{\operatorname{argmin}} \mathcal{L}\left(\sum_{i=1}^{\hat{P}} \mathbf{D}_i^k \hat{\mathbf{X}}_k(\mathbf{w}_i - \mathbf{w}_{i+\hat{P}}), \mathbf{y}\right) + \beta \sum_{i=1}^{2\hat{P}} \|\mathbf{w}_i\|_2 \tag{53}$$

$$\mathbf{w}^{(k)} \in \underset{\mathbf{w} \in \mathcal{C}(\hat{\mathbf{X}}_k)}{\operatorname{argmin}} \mathcal{L}\left(\sum_{i=1}^{\hat{P}} \mathbf{D}_i^k \mathbf{X}(\mathbf{w}_i - \mathbf{w}_{i+\hat{P}}), \mathbf{y}\right) + \beta \sum_{i=1}^{2\hat{P}} \|\mathbf{w}_i\|_2, \tag{54}$$

where $\mathbf{D}_i^k$ denotes the set of arrangements sampled from rank-k data matrix $\hat{\mathbf{X}}_k$. Note that the difference between (53) and (54) is that we use rank-k data for sampling arrangements of both problems while using the full rank data only for (54).

Let us first denote the objective values evaluated at $\hat{\mathbf{w}}^{(k)}$ and $\mathbf{w}^{(k)}$ using the original data $\mathbf{X}$ as $\hat{p}_{\text{cvx}-k}$ and $p_{\text{cvx}-k}$, respectively. Then from Lemma F.1, we can use $\hat{\mathbf{w}}^{(k)}$ to achieve the following approximation guarantee

$$p^* \leq \hat{p}_{\text{cvx}-k} \leq p^* \left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2. \tag{55}$$

Moreover, since (54) utilizes the full rank data, the network output can span a larger output space and therefore the corresponding optimal objective value of the minimization problem is smaller, i.e., $p_{\text{cvx}-k} \leq \hat{p}_{\text{cvx}-k}$. However, $p_{\text{cvx}-k} \geq p^*$ since $p_{\text{cvx}-k}$ has smaller number of arrangements due to using rank-k matrix for hyperplane arrangement sampling.

Combining these observations with (55) yields

$$p^* \leq p_{\text{cvx}-k} \leq \hat{p}_{\text{cvx}-k} \leq p^* \left(1 + \frac{LR\sigma_{k+1}}{\beta}\right)^2.$$

□

APPENDIX G
PROOF OF THEOREM III.2

Suppose that we randomly sample binary vectors $\mathbf{d} \in \{0,1\}^n$, which denotes the diagonal entries of $\mathbf{D}$. Then, we define the probability of $\mathbf{d}$ being the i^{th} arrangement as θ_i, i.e., $p_i := \mathbb{P}[\text{diag}(\mathbf{d}) = \mathbf{D}_i]$. Next, we compute an event where we miss at least one arrangement among P possible ones. Let this event be denoted as $\mathcal{A}$, which is defined as follows

$$\mathbb{P}[\mathcal{A}] = \mathbb{P}\left[\bigcup_{i=1}^{P} \{\text{miss } \mathbf{D}_i\}\right] \leq \sum_{i=1}^{P} \mathbb{P}[\text{diag}(\mathbf{d}) \neq \mathbf{D}_i] = \sum_{i=1}^{P} (1-\theta_i)^{\tilde{P}} \leq P(1-\theta_{min})^{\tilde{P}},$$

where the first inequality follows from the union bound, $\tilde{P}$ denotes the number of arrangements we sample, and $\theta_{min} := \min_i \theta_i$. Then, to be able to sample all arrangements with probability $1-\epsilon$, we choose $\tilde{P}$ such that

$$P(1-\theta_{min})^{\tilde{P}} \leq \epsilon \implies \tilde{P} \geq \frac{\log\left(P/\epsilon\right)}{\log(1/(1-\theta_{min}))}.$$

Next, we use the following identity $\log(1/(1-x)) \geq x$ given $x < 1$, to obtain an upper bound for the RHS of the inequality above

$$\frac{\log\left(P/\epsilon\right)}{\log(1/(1-\theta_{min}))} \geq \frac{\log\left(P/\epsilon\right)}{\theta_{min}} = \frac{P\log\left(P/\epsilon\right)}{\bar{\theta}},$$

where $\bar{\theta} := P\theta_{min}$. Therefore, the threshold for the number of hyperplane arrangements we need to sample simplifies to

$$\tilde{P} \geq \frac{P\log\left(P/\epsilon\right)}{\bar{\theta}} = \mathcal{O}\left(k\left(\frac{n}{k}\right)^k \log\left(\frac{n}{k}\right)\right).$$

Note that this threshold is a polynomial function of all problem parameters, i.e., the number of samples n and feature dimension d, since $P = \mathcal{O}((n/k)^k)$ given a fixed rank k based on Remark III.1 and $\bar{\theta}$ is a constant factor. □

Appendix H
Proof of Corollary V.1

We first replace $\|\mathbf{w}^{(1)}\|_2 \leq 1$ with $\|\mathbf{w}^{(1)}\|_p \leq 1$. Then, the rest of the derivations directly follows from the proof of Theorem II.1 and yield the claimed group ℓ_p regularized convex program in (23). □

Appendix I
Proof of Theorem VII.1

Following Theorem II.1, we have the following dual constraint

$$\begin{aligned}\max_{\mathbf{w}^{(1)}\in\mathcal{B}_2} \left|\mathbf{v}^T\phi(\mathbf{X}\mathbf{w}^{(1)})\right| &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap P_S} \left|\mathbf{v}^T\mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)}\right| \\ &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\substack{\mathbf{w}^{(1)}\in\mathcal{B}_2\\ \mathbf{X}\mathbf{w}^{(1)}\geq 0}} \left|\mathbf{v}^T\mathbf{X}\mathbf{w}^{(1)}\right|,\end{aligned}$$

where the second equality follows from the definition of spike-free matrices. We then apply the same steps in Theorem II.1 for a case with $P = 1$ and $\mathbf{D}_1 = \mathbf{I}_n$ to achieve the convex program claimed in (26). □

Appendix J
Proof of Theorem VIII.1

As in Theorem II.1, we start with the dual of the scaled primal problem in (28), which is formulated as

$$d_v^* = \min_{\mathbf{V}\in\mathbb{R}^{n\times C}} -\mathcal{L}^*(\mathbf{V}) \text{ s.t. } \max_{\mathbf{w}^{(1)}:\,\|\mathbf{w}^{(1)}\|_2\leq 1} \left\|\mathbf{V}^T\phi(\mathbf{X}\mathbf{w}^{(1)})\right\|_2 \leq \beta\,. \tag{56}$$

Now, let us focus on the dual constraint as follows

$$\begin{aligned}\max_{\mathbf{w}^{(1)}\in\mathcal{B}_2} \left\|\mathbf{V}^T\phi(\mathbf{X}\mathbf{w}^{(1)})\right\|_2 &= \max_{\mathbf{w}^{(1)},\mathbf{g}\in\mathcal{B}_2} \mathbf{g}^T\mathbf{V}^T\phi(\mathbf{X}\mathbf{w}^{(1)}) \\ &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\substack{\mathbf{w}^{(1)},\mathbf{g}\in\mathcal{B}_2\\ \mathbf{w}^{(1)}\in P_S}} \mathbf{g}^T\mathbf{V}^T\mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)} \\ &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\substack{\mathbf{w}^{(1)},\mathbf{g}\in\mathcal{B}_2\\ \mathbf{w}^{(1)}\in P_S}} \left\langle \mathbf{V}, \mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)}\mathbf{g}^T\right\rangle \\ &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\substack{\mathbf{Z}=\mathbf{w}^{(1)}\mathbf{g}^T\\ \mathbf{w}^{(1)}\in P_S\\ \mathbf{w}^{(1)},\mathbf{g}\in\mathcal{B}_2}} \langle \mathbf{V}, \mathbf{D}(S)\mathbf{X}\mathbf{Z}\rangle \\ &= \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}} \max_{\mathbf{Z}\in\mathcal{K}} \langle \mathbf{V}, \mathbf{D}(S)\mathbf{X}\mathbf{Z}\rangle\,,\end{aligned}$$

where $\mathcal{K} := \text{Conv}\{\mathbf{u}\mathbf{g}^T : \mathbf{w}^{(1)} \in P_S,\, \|\mathbf{w}^{(1)}\|_2, \|\mathbf{g}\|_2 \leq 1\}$. We also define a new convex norm over the set $\mathcal{K}$ as

$$\|\mathbf{Z}\|_{\mathcal{C}} := \min_{t \geq 0} t \text{ s.t. } \mathbf{W} \in t\,\mathcal{K}.$$

Then, the dual problem (56) can be equivalently written as

$$d_v^* = \min_{\mathbf{V} \in \mathbb{R}^{n \times C}} -\mathcal{L}^*(\mathbf{V}) \text{ s.t. } \max_{\mathbf{Z}: \mathbf{Z} \in \mathcal{K}_i} \langle \mathbf{V}, \mathbf{D}_i \mathbf{X} \mathbf{Z} \rangle \leq \beta \, \forall i \in [P].$$

where $\mathcal{K}_i := \text{Conv}\{\mathbf{w}^{(1)}\mathbf{g}^T : \mathbf{w}^{(1)} \in P_{S_i},\, \|\mathbf{w}^{(1)}\|_2 \leq 1,\, \|\mathbf{Z}\|_* \leq 1\}$ with the corresponding norm $\|\cdot\|_{\mathcal{C}_i}$. We then write the Lagrangian of the above problem form as follows

$$d_v^* = \max_{\mathbf{V} \in \mathbb{R}^{n \times C}} \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^P \\ \boldsymbol{\lambda} \geq 0}} \min_{\mathbf{Z}_i \in \mathcal{K}_i, \forall i} -\mathcal{L}^*(\mathbf{V}) + \sum_{i=1}^P \lambda_i \left(\beta - \langle \mathbf{V}, \mathbf{D}_i \mathbf{X} \mathbf{Z}_i \rangle\right). \tag{57}$$

Invoking Sion's minimax theorem [69] for the $\max\min$ problems, we may express the strong dual of the problem (56) as

$$d_v^* = \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^P \\ \boldsymbol{\lambda} \geq 0}} \min_{\mathbf{Z}_i \in \mathcal{K}_i, \forall i} \max_{\mathbf{V} \in \mathbb{R}^{n \times C}} -\mathcal{L}^*(\mathbf{V}) + \sum_{i=1}^P \lambda_i \left(\beta - \langle \mathbf{V}, \mathbf{D}_i \mathbf{X} \mathbf{Z}_i \rangle\right).$$

Computing the maximum with respect to $\mathbf{V}$, analytically we obtain the strong dual to (56) as

$$d_v^* = \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^P \\ \boldsymbol{\lambda} \geq 0}} \min_{\mathbf{Z}_i \in \mathcal{K}_i, \forall i} \mathcal{L}\left(\sum_{i=1}^P \lambda_i \mathbf{D}(S_i) \mathbf{X} \mathbf{Z}_i, \mathbf{y}\right) + \beta \sum_{i=1}^P \lambda_i.$$

Now we apply a change of variables and define $\mathbf{W}_i = \lambda_i \mathbf{Z}_i$, $\forall i \in [P]$. Thus, we obtain

$$d_v^* = \min_{\substack{\mathbf{W}_i \in \lambda_i \mathcal{K}_i \\ \boldsymbol{\lambda} \geq 0}} \mathcal{L}\left(\sum_{i=1}^P \mathbf{D}(S_i) \mathbf{X} \mathbf{W}_i^*, \mathbf{y}\right) + \beta \sum_{i=1}^P \lambda_i.$$

The variables λ_i, $\forall i \in [P]$ can be eliminated since $\lambda_i = \|\mathbf{W}_i\|_{\mathcal{C}_i}$ is feasible and optimal. Plugging in these expressions, we get

$$d_v^* = \min_{\mathbf{W}_i \in \mathbb{R}^{d \times C}} \mathcal{L}\left(\sum_{i=1}^P \mathbf{D}(S_i) \mathbf{X} \mathbf{W}_i, \mathbf{y}\right) + \beta \sum_{i=1}^P \|\mathbf{W}_i\|_{\mathcal{C}_i}, \tag{58}$$

which is identical to the objective value of the convex program (29). Since the value of the convex program is equal to the value of it's dual d_v^* in (57), and $p_v^* \geq d_v^*$, we conclude that $p_v^* = d_v^*$, which is equal to the value of the convex program (29) achieved by the prescribed parameters.

□

Appendix K
Proof of Theorem VIII.2

As in Theorem VIII.1, we start with scaling the primal problem in (30) as

$$p_{v1}^* := \min_{\theta \in \Theta_s} \mathcal{L}(f_\theta(\mathbf{X}), \mathbf{y}) + \beta \sum_{j=1}^m \|\mathbf{w}_j^{(2)}\|_1. \tag{59}$$

which has the following dual with respect to $\mathbf{w}_j^{(2)}$

$$p_{v1}^* = d_{v1}^* = \min_{\mathbf{V} \in \mathbb{R}^{n \times C}} -\mathcal{L}^*(\mathbf{V}) \text{ s.t. } \max_{\mathbf{w}^{(1)} \in \mathcal{B}_2} \left|\mathbf{v}_l^T \phi(\mathbf{X}\mathbf{w}^{(1)})\right| \leq \beta \, \forall l \in [C]. \tag{60}$$

Now, let us rewrite the dual constraint as follows

$$\max_{\mathbf{w}^{(1)}\in\mathcal{B}_2}\left|\mathbf{v}_l^T\phi(\mathbf{X}\mathbf{w}^{(1)})\right| = \max_{\substack{S\subseteq[n]\\ S\in\mathcal{S}}}\max_{\mathbf{w}^{(1)}\in\mathcal{B}_2\cap P_S}\left|\mathbf{v}_l^T\mathbf{D}(S)\mathbf{X}\mathbf{w}^{(1)}\right|.$$

Then, the dual problem (60) can be equivalently written as

$$d_{v1}^* = \min_{\mathbf{V}\in\mathbb{R}^{n\times C}} -\mathcal{L}^*(\mathbf{V}) \text{ s.t. } \max_{\mathbf{Z}_i\in\mathcal{K}_i}\left|\mathbf{v}_l^T\mathbf{D}_i\mathbf{X}\mathbf{w}^{(1)}\right| \leq \beta\,\forall i\in[P],\,\forall l\in[C].$$

The rest of the proofs directly follow from Theorem II.1, which yield the convex problem in (31). □

APPENDIX L
CONSTRUCTING HYPERPLANE ARRANGEMENTS IN POLYNOMIAL TIME

We consider the number of all distinct sign patterns $\text{sign}(\mathbf{X}\mathbf{z})$ for all possible choices $z\in\mathbb{R}^d$. Note that this number is the number of regions in a partition of $\mathbb{R}^d$ by hyperplanes passing through the origin, and are perpendicular to the rows of $\mathbf{X}$. We now show that the dimension d can be replaced with $\text{rank}(\mathbf{X})$ without loss of generality. Suppose that the data matrix $\mathbf{X}$ has rank r. We may express $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ using its Singular Value Decomposition in compact form, where $\mathbf{U}\in\mathbb{R}^{n\times r},\mathbf{\Sigma}\in\mathbb{R}^{r\times r},\mathbf{V}^T\in\mathbb{R}^{r\times d}$. For any vector $z\in\mathbb{R}^d$ we have $\mathbf{X}\mathbf{z}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{z}=\mathbf{U}\mathbf{z}'$ for some $\mathbf{z}'\in\mathbb{R}^r$. Therefore, the number of distinct sign patterns $\text{sign}(\mathbf{X}\mathbf{z})$ for all possible $\mathbf{z}\in\mathbb{R}^d$ is equal to the number of distinct sign patterns $\text{sign}(\mathbf{U}\mathbf{z}')$ for all possible $\mathbf{z}'\in\mathbb{R}^r$.

Consider an arrangement of n hyperplanes $\in\mathbb{R}^r$, where $n\geq r$. Let us denote the number of regions in this arrangement by $P_{n,r}$. In [22, 52] it's shown that this number satisfies

$$P_{n,r}\leq 2\sum_{k=0}^{r-1}\binom{n-1}{k}.$$

For hyperplanes in general position, the above inequality is in fact an equality. In [26], the authors present an algorithm that enumerates all possible hyperplane arrangements $\mathcal{O}(n^r)$ time, which can be used to construct the data for the convex program (7).

APPENDIX M
DUAL PROBLEMS FOR DIFFERENT FORMULATIONS

M-A Dual problem for (33)

The following lemma proves the dual form of (33).

Lemma M.1. *The dual form of the following primal problem*

$$\min_{\mathbf{w}_j^{(1)}\in\mathcal{B}_2}\min_{\{w_j^{(2)}\}_{j=1}^m}\mathcal{L}\left(\sum_{j=1}^m\phi(\mathbf{X}\mathbf{w}_j^{(1)})w_j^{(2)},\mathbf{y}\right)+\beta\sum_{j=1}^m|w_j^{(2)}|,$$

is given by the following

$$\min_{\mathbf{w}_j^{(1)}\in\mathcal{B}_2}\max_{\substack{\mathbf{v}\in\mathbb{R}^n \text{ s.t.}\\ |\mathbf{v}^T\phi(\mathbf{X}\mathbf{w}_j^{(1)})|\leq\beta}} -\mathcal{L}^*(\mathbf{v}).$$

[**Proof of Lemma M.1**] Let us first reparametrize the primal problem as follows

$$\min_{\mathbf{w}_j^{(1)}\in\mathcal{B}_2}\min_{\mathbf{r},w_j^{(2)}}\mathcal{L}(\mathbf{r},\mathbf{y})+\beta\sum_{j=1}^m|w_j^{(2)}| \text{ s.t. } \mathbf{r}=\sum_{j=1}^m\phi(\mathbf{X}\mathbf{w}_j^{(1)})w_j^{(2)},$$

which has the following Lagrangian

$$L(\mathbf{v},\mathbf{r},\mathbf{w}_j^{(1)},w_j^{(2)})=\mathcal{L}(\mathbf{r},\mathbf{y})+\beta\sum_{j=1}^m|w_j^{(2)}|-\mathbf{v}^T\mathbf{r}+\mathbf{v}^T\sum_{j=1}^m\phi(\mathbf{X}\mathbf{w}_j^{(1)})w_j^{(2)}.$$

Then, minimizing over $\mathbf{r}$ and $\mathbf{w}^{(2)}$ yields the proposed dual form.

M-B Dual problem for (17)

Let us first reparameterize the primal problem as follows

$$\max_{\mathbf{M},\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \sigma_{\max}(\mathbf{M}) \leq \beta,\ \mathbf{M} = [\mathbf{X}_1^T\mathbf{v} \dots \mathbf{X}_K^T\mathbf{v}].$$

Then the Lagrangian is as follows

$$\begin{aligned} L(\lambda, \mathbf{Z}, \mathbf{M}, \mathbf{v}) &= -\mathcal{L}^*(\mathbf{v}) + \lambda(\beta - \sigma_{\max}(\mathbf{M})) + \mathbf{tr}(\mathbf{Z}^T\mathbf{M}) - \mathbf{tr}(\mathbf{Z}^T[\mathbf{X}_1^T\mathbf{v} \dots \mathbf{X}_K^T\mathbf{v}]) \\ &= -\mathcal{L}^*(\mathbf{v}) + \lambda(\beta - \sigma_{\max}(\mathbf{M})) + \mathbf{tr}(\mathbf{Z}^T\mathbf{M}) - \mathbf{v}^T\sum_{k=1}^K \mathbf{X}_k\mathbf{z}_k, \end{aligned}$$

where $\lambda \geq 0$ and $\mathbf{tr}$ denotes the trace operation. Then maximizing over $\mathbf{M}$ and $\mathbf{v}$ yields the following dual form

$$\min_{\mathbf{z}_k \in \mathbb{R}^d, \forall k \in [K]} \mathcal{L}\left(\sum_{k=1}^K \mathbf{X}_k\mathbf{z}_k, \mathbf{y}\right) + \beta \Big\| [\mathbf{z}_1, \dots, \mathbf{z}_K] \Big\|_*,$$

where $\Big\| [\mathbf{z}_1, \dots, \mathbf{z}_K] \Big\|_* = \|\mathbf{Z}\|_* = \sum_i \sigma_i(\mathbf{Z})$ is the ℓ_1-norm of singular values, i.e., nuclear norm [59].

M-C Dual problem for (19)

Let us denote the eigenvalue decomposition of $\mathbf{W}_j^{(1)}$ as $\mathbf{W}_j^{(1)} = \mathbf{F}\mathbf{D}_j\mathbf{F}^H$, where $\mathbf{F} \in \mathbb{C}^{d\times d}$ is the Discrete Fourier Transform matrix and $\mathbf{D}_j \in \mathbb{C}^{d\times d}$ is a diagonal matrix. Then, applying the scaling in Lemma II.1 and then taking the dual as in Lemma M.1 yields

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \|\mathbf{v}^T\mathbf{X}\mathbf{F}\mathbf{D}\mathbf{F}^H\|_2 \leq \beta,\ \forall \mathbf{D} : \|\mathbf{D}\|_F^2 \leq d,$$

which can be equivalently written as

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \|\mathbf{v}^T\hat{\mathbf{X}}\mathbf{D}\|_2 \leq \beta,\ \forall \mathbf{D} : \|\mathbf{D}\|_F^2 \leq d.$$

Since $\mathbf{D}$ is diagonal, $\|\mathbf{D}\|_F^2 \leq d$ is equivalent to $\sum_{i=1}^d D_{ii}^2 \leq 1$. Therefore, the problem above can be further simplified as

$$\max_{\mathbf{v}} -\mathcal{L}^*(\mathbf{v}) \text{ s.t. } \|\mathbf{v}^T\hat{\mathbf{X}}\|_\infty \leq \frac{\beta}{\sqrt{d}}.$$

Then, taking the dual of this problem gives the following

$$\min_{\mathbf{z}\in\mathbb{C}^d} \mathcal{L}\left(\hat{\mathbf{X}}\mathbf{z}, \mathbf{y}\right) + \frac{\beta}{\sqrt{d}}\|\mathbf{z}\|_1.$$

APPENDIX N
NEURAL GAUGE FUNCTION AND EQUIVALENCE TO MINIMUM-NORM NETWORKS

Consider the gauge function

$$p^g = \min_{r\geq 0} r \text{ s.t. } r\mathbf{y} \in \text{conv}(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})$$

and its dual representation in terms of the support function of the polar of $\text{conv}(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})$

$$d^g = \max_{\mathbf{v}} \mathbf{v}^T\mathbf{y} \text{ s.t. } \mathbf{v} \in (\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ.$$

Since the set $\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}$ is a closed convex set that contains the origin, we have $p^g = d^g$ [60] and $(\text{conv}(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}))^\circ = (\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X})^\circ$. The result in Section C implies that the above value is equal to the semi-infinite dual value, i.e., $p^g = p^g_\infty$, where

$$p^g_\infty := \min_{\boldsymbol{\mu}} \|\boldsymbol{\mu}\|_{TV} \text{ s.t. } \int_{\mathbf{w}^{(1)}\in\mathcal{B}_2} \phi(\mathbf{X}\mathbf{w}^{(1)}) d\boldsymbol{\mu}(\mathbf{w}^{(1)}) = \mathbf{y}\,.$$

By Caratheodory's theorem, there exists optimal solutions of the above problem consisting of m^* Dirac deltas [60, 61], and therefore

$$p^g_\infty = \min_{\mathbf{w}_j^{(1)} \in \mathcal{B}_2, w_j^{(2)}} \sum_{j=1}^{m^*} |w_j^{(2)}| \text{ s.t. } \sum_{j=1}^{m^*} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) w_j^{(2)} = \mathbf{y}\,,$$

where we define m^* as the number of Dirac delta's in the optimal solution to p^g_∞. If the optimizer is non-unique, we define m^* as the minimum cardinality solution among the set of optimal solutions. Now consider the non-convex problem

$$\min_{\{\mathbf{w}_j^{(1)}, w_j^{(2)}\}_{j=1}^m} \|\mathbf{w}^{(2)}\|_1 \text{ s.t. } \sum_{j=1}^{m} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) w_j^{(2)} = \mathbf{y},\ \mathbf{w}_j^{(1)} \in \mathcal{B}_2\,.$$

Using the standard parameterization for ℓ_1-norm we get

$$\min_{\substack{\{\mathbf{w}_j^{(1)}\}_{j=1}^m \\ \mathbf{s}\geq 0 \\ \mathbf{t}\geq 0}} \sum_{j=1}^{m}(t_j + s_j) \text{ s.t.} \sum_{j=1}^{m} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) t_j - \phi(\mathbf{X}\mathbf{w}_j^{(1)}) s_j = \mathbf{y},\ \mathbf{w}_j^{(1)} \in \mathcal{B}_2\,, \forall j.$$

Introducing a slack variable $r \in \mathbb{R}_+$, an equivalent representation can be written as

$$\min_{\substack{\{\mathbf{w}_j^{(1)}\}_{j=1}^m \\ \mathbf{s},\mathbf{t},r\geq 0}} r \text{ s.t. } \sum_{j=1}^{m} \phi(\mathbf{X}\mathbf{w}_j^{(1)}) t_j - \phi(\mathbf{X}\mathbf{w}_j^{(1)}) s_j = \mathbf{y},\ \sum_{j=1}^{m}(t_j + s_j) = r,\ \mathbf{w}_j^{(1)} \in \mathcal{B}_2\,, \forall j.$$

Note that $r > 0$ as long as $\mathbf{y} \neq \mathbf{0}$. Rescaling variables by letting $t'_j = t_j/r$, $s'_j = s_j/r$ in the above program, we obtain

$$\min_{\substack{\{\mathbf{w}_j^{(1)}\}_{j=1}^m \\ \mathbf{s}',\mathbf{t}',r\geq 0}} r \text{ s.t.} \sum_{j=1}^{m} \left(\phi(\mathbf{X}\mathbf{w}_j^{(1)}) t'_j - \phi(\mathbf{X}\mathbf{w}_j^{(1)}) s'_j\right) = r\mathbf{y},\ \sum_{j=1}^{m}(t'_j + s'_j) = 1, \mathbf{w}_j^{(1)} \in \mathcal{B}_2, \forall j\,.$$

Suppose that $m \geq m^*$. It holds that

$$\begin{aligned} &\exists \mathbf{s}', \mathbf{t}' \geq 0\,, \{\mathbf{w}_j^{(1)}\}_{j=1}^m \text{ s.t. } \sum_{j=1}^{m}(t'_j + s'_j) = 1,\ \|\mathbf{w}_j^{(1)}\|_2 \leq 1,\ \forall j,\ \sum_{j=1}^{m}(\mathbf{X}\mathbf{w}_j^{(1)}) t'_j - \phi(\mathbf{X}\mathbf{w}_j^{(1)}) s'_j = r\mathbf{y} \\ &\quad \iff r\mathbf{y} \in \text{conv}(\mathcal{Q}_\mathbf{X} \cup -\mathcal{Q}_\mathbf{X}). \end{aligned} \tag{61}$$

We conclude that the optimal value of (61) is identical to the gauge function p_g.

Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under CAREER award CCF-2236829, in part by the U.S. Army Research Office Early Career Award W911NF-21-1-0242, and in part by the Office of Naval Research under Grant N00014-24-1-2164.

References

[1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[2] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[3] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization, 2019.

[4] Sanjeev Arora, Simon S. Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu. Harnessing the power of infinitely wide deep nets on small-data tasks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkl8sJBYvH.

[5] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete applied mathematics*, 65(1-3):21–46, 1996.
[6] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.
[7] Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31(1-58):26, 1997.
[8] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. pages 583–593, 2019.
[9] Yoshua Bengio, Nicolas L Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in neural information processing systems*, pages 123–130, 2006.
[10] Daniel Bienstock, Gonzalo Muñoz, and Sebastian Pokutta. Principled deep neural network training through linear programming, 2018.
[11] Digvijay Boob, Santanu S Dey, and Guanghui Lan. Complexity of training relu neural network, 2018.
[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004.
[13] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
[14] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Trans. Info Theory*, 51(12):4203–4215, December 2005.
[15] E. J. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.
[16] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Computing*, 20(1):33–61, 1998.
[17] Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming, 2018.
[18] Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1305–1338. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/chizat20a.html.
[19] Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.
[20] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012.
[21] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.
[22] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.
[23] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
[24] D. L. Donoho. Compressed sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, April 2006.
[25] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.
[26] Herbert Edelsbrunner, Joseph O'Rourke, and Raimund Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15(2):341–363, 1986.
[27] Tolga Ergen and Mert Pilanci. Convex geometry of two-layer relu networks: Implicit autoencoding and interpretable models. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4024–4033, Online, 26–28 Aug 2020. PMLR. URL http://proceedings.mlr.press/v108/ergen20a.html.
[28] Tolga Ergen and Mert Pilanci. Implicit convex regularizers of cnn architectures: Convex optimization of two- and three-layer networks in polynomial time. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0N8jUH4JMv6.
[29] Tolga Ergen and Mert Pilanci. Convex geometry and duality of over-parameterized neural networks. *Journal of Machine Learning Research*, 22(212):1–63, 2021. URL http://jmlr.org/papers/v22/20-1447.html.
[30] Tolga Ergen and Mert Pilanci. Global optimality beyond two layers: Training deep relu networks via convex

programs. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2993–3003. PMLR, 18–24 Jul 2021. URL http://proceedings.mlr.press/v139/ergen21a.html.

[31] Tolga Ergen and Mert Pilanci. Revealing the structure of deep neural networks via convex duality. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3004–3014. PMLR, 18–24 Jul 2021. URL http://proceedings.mlr.press/v139/ergen21b.html.

[32] Tolga Ergen, Behnam Neyshabur, and Harsh Mehta. Convexifying transformers: Improving optimization and understanding of transformer networks, 2022. URL https://arxiv.org/abs/2211.11052.

[33] Tolga Ergen, Arda Sahiner, Batu Ozturkler, John M. Pauly, Morteza Mardani, and Mert Pilanci. Demystifying batch normalization in reLU networks: Equivalent convex optimization models and implicit regularization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=6XGgutacQ0B.

[34] Cong Fang, Yihong Gu, Weizhong Zhang, and Tong Zhang. Convex formulation of overparameterized deep neural networks, 2019.

[35] M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford, 2002. Available online: http://faculty.washington.edu/mfazel/thesis-final.pdf.

[36] Miguel Angel Goberna and Marco López-Cerdá. *Linear semi-infinite optimization*. 01 1998. doi: 10.1007/978-1-4899-8044-1_3.

[37] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1004–1042, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

[38] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

[39] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

[40] Branko Grünbaum, Victor Klee, Micha A Perles, and Geoffrey Colin Shephard. *Convex polytopes*, volume 16. Springer, 1967.

[41] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization, 2017.

[42] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks, 2019.

[43] Vikul Gupta, Burak Bartan, Tolga Ergen, and Mert Pilanci. Convex neural autoregressive models: Towards tractable, expressive, and theoretically-backed models for sequential forecasting and generation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3890–3894, 2021. doi: 10.1109/ICASSP39728.2021.9413662.

[44] Dan Halperin and Micha Sharir. Arrangements. In *Handbook of discrete and computational geometry*, pages 723–762. Chapman and Hall/CRC, 2017.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[46] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[47] Guy Kornowski, Gilad Yehudai, and Ohad Shamir. From tempered to benign overfitting in relu neural networks, 2023.

[48] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset. http://www.cs.toronto.edu/kriz/cifar.html, 2014.

[49] Jonathan Lacotte and Mert Pilanci. All local minima are global for two-layer relu neural networks: The hidden convex optimization landscape, 2020.

[50] Edward Moroshko, Suriya Gunasekar, Blake Woodworth, Jason D. Lee, Nathan Srebro, and Daniel Soudry.

Implicit bias in deep linear classification: Initialization scale vs training accuracy, 2020.
[51] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning, 2014.
[52] Piyush C Ojha. Enumeration of linear threshold functions from the lattice of hyperplane intersections. *IEEE Transactions on Neural Networks*, 11(4):839–850, 2000.
[53] Batu Ozturkler, Arda Sahiner, Mert Pilanci, Shreyas Vasanawala, John Pauly, and Morteza Mardani. Scalable and interpretable neural mri reconstruction via layer-wise training. In *International Society for Magnetic Resonance in Medicine*, 2021.
[54] Rahul Parhi and Robert D. Nowak. Minimum "norm" neural networks are splines, 2019.
[55] Rahul Parhi and Robert D Nowak. What kinds of functions do deep neural networks learn? insights from variational spline theory. *SIAM Journal on Mathematics of Data Science*, 4(2):464–489, 2022.
[56] Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7695–7705, Virtual, 13–18 Jul 2020. PMLR. URL http://proceedings.mlr.press/v119/pilanci20a.html.
[57] Miroslav Rada and Michal Cerny. A new algorithm for enumeration of cells of hyperplane arrangements and a comparison with avis and fukuda's reverse search. *SIAM Journal on Discrete Mathematics*, 32(1):455–473, 2018.
[58] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *Advances in neural information processing systems*, 33:21174–21187, 2020.
[59] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.
[60] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.
[61] Saharon Rosset, Grzegorz Swirszcz, Nathan Srebro, and Ji Zhu. L1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pages 544–558. Springer, 2007.
[62] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 1964.
[63] Arda Sahiner, Tolga Ergen, Batu Ozturkler, John M Pauly, Morteza Mardani, and Mert Pilanci. Scaling convex neural networks with burer-monteiro factorization.
[64] Arda Sahiner, Tolga Ergen, John M. Pauly, and Mert Pilanci. Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=fGF8qAqpXXG.
[65] Arda Sahiner, Tolga Ergen, Batu Ozturkler, Burak Bartan, John M. Pauly, Morteza Mardani, and Mert Pilanci. Hidden convexity of wasserstein GANs: Interpretable generative models with closed-form solutions. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=e2Lle5cij9D.
[66] Arda Sahiner, Tolga Ergen, Batu Ozturkler, John Pauly, Morteza Mardani, and Mert Pilanci. Unraveling attention via convex duality: Analysis and interpretations of vision transformers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19050–19088. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/sahiner22a.html.
[67] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *CoRR*, abs/1902.05040, 2019. URL http://arxiv.org/abs/1902.05040.
[68] Alexander Shapiro. Semi-infinite programming, duality, discretization and optimality conditions. *Optimization*, 58(2):133–161, 2009.
[69] Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.
[70] Richard P Stanley et al. An introduction to hyperplane arrangements. *Geometric combinatorics*, 13:389–496, 2004.
[71] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.
[72] Ryan J Tibshirani. Equivalences between sparse models and neural networks, 2021.

[73] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Info Theory*, 50(10): 2231–2242, 2004.

[74] RH Tütüncü, KC Toh, and MJ Todd. Sdpt3—a matlab software package for semidefinite-quadratic-linear programming, version 3.0, 2001.

[75] Yifei Wang, Jonathan Lacotte, and Mert Pilanci. The hidden convex optimization landscape of regularized two-layer reLU networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Z7Lk2cQEG8a.

[76] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/woodworth20a.html.

[77] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[78] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

=2