Fast Convex Optimization for Two-Layer ReLU Networks: Equivalent Model Classes and Cone Decompositions

Aaron Mishkin [1] **Arda Sahiner** [2] **Mert Pilanci** [2]

Abstract

We develop fast algorithms and robust software for convex optimization of two-layer neural networks with ReLU activation functions. Our work leverages a convex reformulation of the standard weight-decay penalized training problem as a set of group-ℓ_1-regularized *data-local* models, where locality is enforced by polyhedral cone constraints. In the special case of zero-regularization, we show that this problem is exactly equivalent to *unconstrained* optimization of a convex "gated ReLU" network. For problems with non-zero regularization, we show that convex gated ReLU models obtain data-dependent approximation bounds for the ReLU training problem. To optimize the convex reformulations, we develop an accelerated proximal gradient method and a practical augmented Lagrangian solver. We show that these approaches are faster than standard training heuristics for the non-convex problem, such as SGD, and outperform commercial interior-point solvers. Experimentally, we verify our theoretical results, explore the group-ℓ_1 regularization path, and scale convex optimization for neural networks to image classification on MNIST and CIFAR-10.

1. Introduction

It is well-known that global optimization of neural networks is NP-Hard (Blum & Rivest, 1988). Despite the theoretical difficulty, highly accurate models are trained in practice using stochastic gradient methods (SGMs) (Bengio, 2012). Unfortunately, SGMs cannot guarantee convergence to a local optimum of the non-convex training loss (Ge et al., 2015) and existing methods rarely certify convergence to a stationary point of any type (Goodfellow et al., 2016). SGMs are also sensitive to hyper-parameters; they converge slowly, to different stationary points (Neyshabur et al., 2017), or even diverge depending on the choice of step-size. Parameters like the random seed complicate replications and can produce model churn, where networks learned using the same procedure give different predictions for the same inputs (Henderson et al., 2018; Bhojanapalli et al., 2021). See Figure 1 for an example. For most applications, practitioners use domain knowledge and costly hyper-parameter search to cope with these challenges.

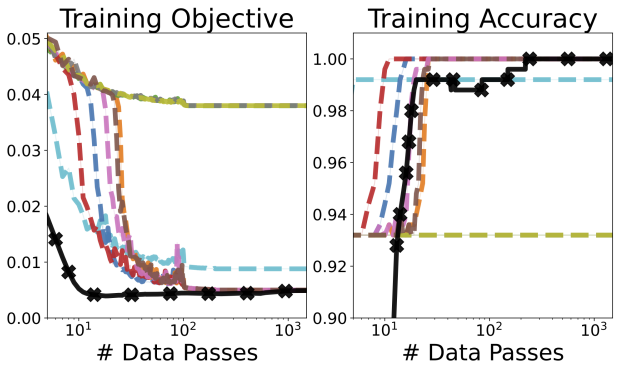

Figure 1. Convex (solid line) and non-convex (dashed) optimization of a two-layer ReLU network for a realizable synthetic classification problem. We plot only one run of the convex solver since they are nearly identical and all reach perfect accuracy. In contrast, $4/10$ runs of SGD on the non-convex problem converge to sub-optimal stationary points.

In contrast, we propose to optimize shallow models via *convex reformulations* of the training objective. Recent work by Pilanci & Ergen (2020) uses duality theory to show two-layer neural networks with ReLU activations and weight decay regularization may be re-expressed as a linear model with a group-ℓ_1 penalty and polyhedral cone constraints. Subsequent research extends this model space, deriving convex formulations for convolutions (Sahiner et al., 2021c; Ergen & Pilanci, 2021b; Gupta et al., 2021), vector-outputs (Sahiner et al., 2021b), batch normalization (Ergen et al., 2021), generative models (Sahiner et al., 2021a) and deeper networks (Ergen & Pilanci, 2021a;c). However, existing work is largely focused on model classes, rather than leveraging convexification to train neural networks.

This paper develops fast optimization algorithms for two-layer ReLU models by carefully studying the space of equivalent models. We show that unregularized ReLU networks

[1]Department of Computer Science, Stanford University [2]Department of Electrical Engineering, Stanford University. Correspondence to: Aaron Mishkin <amishkin@cs.stanford.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

can be trained by decomposing the solution to an unconstrained generalized linear model (GLM) onto a difference of polyhedral cones. With non-zero regularization, the same unconstrained problem yields a data-dependent approximation of the optimal solution which differs only in the norm of the model weights. To fit this GLM, we develop a proximal-gradient method that combines the convex optimization toolbox with GPU acceleration. We also use this optimizer as a sub-routine for an augmented Lagrangian method that quickly and robustly trains ReLU networks via the (constrained) convex reformulation. Our deterministic optimizers give both convergence and optimality guarantees.

To summarize, our main contributions are the following:

- A new class of *unconstrained* convex optimization problems which are equivalent to training an unregularized ReLU model and approximation guarantees for the case of non-zero regularization.
- An accelerated proximal-gradient method for this unconstrained problem that improves the complexity of computing a global optimum from $O(1/\epsilon^2)$ to $O(1/\sqrt{\epsilon})$ iterations compared to subgradient methods.
- An augmented Lagrangian method for the constrained convex reformulation which uses our unconstrained solver as a sub-routine and outperforms commercial interior-point software such as MOSEK (ApS, 2019).
- Extensive experiments which validate our theoretical results, carefully explore the properties of our optimization methods, and scale convex optimization for ReLU networks to MNIST and CIFAR-10.

Quality software is key to practical use of our methods. As such, we also provide `scnn`, an open-source package for training neural networks by convex optimization.[1]

1.1. Related Work

Our work combines ideas from the literature on convex neural networks, accelerated methods, and constrained solvers.

Convex Neural Networks: There have been repeated attempts to develop convex neural networks. Bengio et al. (2006) view two-layer neural networks as convex models, but their work requires the first-layer weights to be fixed. Similarly, extreme learning machines (Huang et al., 2006) obtain a convex problem by using a random first-layer; these models can obtain zero training error for over-parameterized problems (Woodworth et al., 2020), but do not learn parsimonious latent representations as in our approach.

Bach (2017) analyze infinite-width two-layer networks; these methods are not implementable, but may be viewed as convex problems. Other research considers the separate problem of neural networks for which the prediction function is convex (Amos et al., 2017; Sivaprasad et al., 2021).

In concurrent work, Bai et al. (2022) consider training two-layer ReLU networks via convex reformulations using ADMM. Their approach requires solving a linear system at each iteration, or uses coordinate descent to solve the ADMM sub-problems. In practice, our solvers scale to larger datasets and allow for more activation patterns.

Accelerated Proximal Gradient: Beck & Teboulle (2009); Nesterov (2013) were the first to extend optimal gradient methods (Nesterov, 1983) to composite problems. Work since then includes extensions to stochastic (Schmidt et al., 2011) and non-convex (Li & Lin, 2015) optimization. See Parikh & Boyd (2014) for a survey of proximal algorithms, including proximal gradient.

Augmented Lagrangian Methods: The convergence theory was initially developed by Rockafellar (Rockafellar, 1976a;b). More recent work includes practical guidelines (Birgin & Martínez, 2014) and acceleration techniques (Kang et al., 2015). See Bertsekas (2014) for exhaustive theoretical developments.

2. Convex Reformulations

Let $X \in \mathbb{R}^{n \times d}$ be a data matrix and $y \in \mathbb{R}^n$ the associated targets. We are interested in two-layer ReLU networks,

$$h_{W_1,w_2}(X) = \sum_{i=1}^{m} (XW_{1i})_+ w_{2i},$$

where $W_1 \in \mathbb{R}^{m \times d}$, $w_2 \in \mathbb{R}^m$ are the weights of the first and second layers, m is the number of hidden units, and $(\cdot)_+ = \max\{\cdot, 0\}$ is the ReLU activation. Fitting h_{W_1,w_2} by minimizing convex loss $\mathcal{L}$ with weight decay (ℓ_2) regularization leads to the optimization problem (NC-ReLU),

$$\min_{W_1,w_2} \mathcal{L}\Big(h_{W_1,w_2}(X), y\Big) + \frac{\lambda}{2} \sum_{i=1}^{m} \|W_{1i}\|_2^2 + |w_{2i}|^2, \quad (1)$$

where $\lambda \geq 0$ is the regularization strength. While Problem 1 is non-convex, Pilanci & Ergen (2020) show that there is an equivalent convex optimization problem with the same optimal value if $m \geq m^*$ for some $m^* \leq n + 1$. Furthermore, Wang et al. (2021) showed that all optimal solutions to (1) can be found via the convex problem.

2.1. Sub-Sampled ReLU Convex Programs

The convex reformulation for the NC-ReLU objective is based on "enumerating" the possible activations of a single neuron in the hidden layer. The activation patterns a ReLU neuron $(Xw)_+$ can take for fixed X are described by

$$\mathcal{D}_X = \left\{ D = \text{diag}(\mathbb{1}(Xu \geq 0)) : u \in \mathbb{R}^d \right\},$$

[1] https://github.com/pilancilab/scnn

which grows as $|\mathcal{D}_X| \in O(r(n/r)^r)$ for $r := \text{rank}(X)$ (Pilanci & Ergen, 2020). For $D_i \in \mathcal{D}_X$, the set of vectors u which achieve the corresponding activation pattern, meaning $D_i X u = (Xu)_+$, is the following convex cone:

$$\mathcal{K}_i = \left\{ u \in \mathbb{R}^d : (2D_i - I)Xu \succeq 0 \right\}.$$

For any subset $\tilde{\mathcal{D}} \subseteq \mathcal{D}_X$, we define the sub-sampled convex optimization problem (C-ReLU):

$$\min_{v,w} \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X(v_i - w_i), y \Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \quad \text{s.t.} \quad v_i, w_i \in \mathcal{K}_i. \tag{2}$$

Pilanci & Ergen (2020) prove NC-ReLU and C-ReLU are equivalent using linear semi-infinite duality theory (Goberna & López, 2002). However, this result requires $m \geq m^*$ and the full enumeration of the activations of a neuron: $\tilde{\mathcal{D}} = \mathcal{D}_X$. In practice, learning with $\mathcal{D}_X$ is computationally infeasible except for special cases where the data are low rank. By introducing sub-sampled models, we relax the dependencies on m^* and $\mathcal{D}_X$ to simple inclusions involving $\tilde{\mathcal{D}}$.

Theorem 2.1. *Suppose (W_1^*, w_2^*) and (v^*, w^*) are global minima of the NC-ReLU (12) and C-ReLU (2) problems, respectively. If the number of hidden units satisfies*

$$m \geq b := \sum_{D_i \in \tilde{\mathcal{D}}} \left| \{ v_i^* : v_i^* \neq 0 \} \cup \{ w_i^* : w_i^* \neq 0 \} \right|,$$

and the optimal activations are in the convex model,

$$\{ \text{diag}\left(XW_{1i}^* \geq 0 : i \in [m]\right) \} \subseteq \tilde{\mathcal{D}},$$

then the two problems have same the optimal value.

See Appendix A for proof. The advantages of this theorem over existing results are (i) the simple and duality-free proof, and (ii) the dependence on $\tilde{\mathcal{D}}$, which we show in Section 5 can be much smaller than $\mathcal{D}_X$ while still performing comparably to NC-ReLU. Theorem 2.1 also reveals that m^* is determined by the number of active "neurons" at the optimal solution of the full C-ReLU problem with $\tilde{\mathcal{D}} = \mathcal{D}_X$.

2.2. Unconstrained Relaxation: Gated ReLUs

Solving C-ReLU using scalable first-order methods typically requires projecting on $\mathcal{K}_i$, which is an expensive quadratic program in the general case. To circumvent this, we consider the following unconstrained relaxation (C-GReLU):

$$\min_u \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i, y \Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i\|_2. \tag{3}$$

At first look, this problem is a high-dimensional GLM with group-ℓ_1 regularization. In fact, C-GReLU is the convex reformulation of another neural network optimization problem.

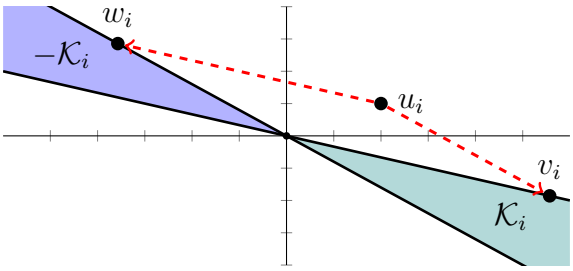

Figure 2. An illustration of the Cone Decomposition (CD) procedure: u_i is decomposed onto the Minkowski difference $\mathcal{K}_i - \mathcal{K}_i$.

Let $\mathcal{G} \subset \mathbb{R}^d$ and consider the model,

$$h_{W_1,w_2}(X) = \sum_{g_i \in \mathcal{G}} \phi_g(X, W_{1i}) w_{2i},$$

where $\phi_g(X, u) = \text{diag}(\mathbb{1}(Xg \geq 0))Xu$ is a "gated ReLU" activation function with fixed gate vector g (Fiat et al., 2019). C-GReLU is equivalent to training this gated ReLU network.

Theorem 2.2. *Let $g_i \in R^d$ such that $diag(Xg_i \geq 0) = D_i$ and $\tilde{\mathcal{G}} = \left\{ g_i : D_i \in \tilde{\mathcal{D}} \right\}$. Then, C-GReLU is equivalent to the following gated ReLU problem (NC-GReLU):*

$$\min_{W_1,w_2} \mathcal{L}\Big(\sum_{g_i \in \tilde{\mathcal{G}}} \phi_{g_i}(X, W_{1i}) w_{2i}, y \Big) + \frac{\lambda}{2} \sum_{g_i \in \tilde{\mathcal{G}}} \|W_{1i}\|_2^2 + w_{2i}^2. \tag{4}$$

See Appendix A for proof. We can use Theorem 2.2 to fit Gated ReLU networks by (much easier) unconstrained minimization. However, as the next section shows, we can also leverage C-GReLU to approximate or exactly solve the original ReLU problem.

3. Equivalence of ReLU and Gated ReLU

This section builds upon our sub-sampled convex reformulations to show that the C-ReLU and C-GReLU problems are equivalent up the norm of their optimal solutions. As a consequence, we give an approximation algorithm for ReLU networks that first computes the solution to C-GReLU and then solves an auxiliary *cone decomposition* problem. The cone decomposition can be formulated as a linear program (LP) or second-order cone program (SOCP), and admits a closed form when X is full row-rank. Before presenting these fully-general results, we study the unregularized setting, where we show the approximation is exact. All proofs are deferred to Appendix B.

Let $\lambda = 0$ and consider the C-GReLU problem (3). For each $D_i \in \tilde{\mathcal{D}}$, we seek to decompose the optimal data-local models as $u_i^* = v_i - w_i \in \mathcal{K}_i - \mathcal{K}_i$. If these decompositions exist, collecting them into $(v, w) = \{(v_i, w_i)\}$ gives a feasible point for the C-ReLU problem with the same optimal objective value as C-GReLU.The next proposition gives sufficient conditions on the data for this to happen.

Proposition 3.1. *If X is full row-rank, then $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ for every $D_i \in \mathcal{D}_\mathcal{X}$. As a result, the C-ReLU, C-GReLU, NC-ReLU, and NC-GReLU problems are all equivalent.*

Unfortunately, Proposition 3.1 does not extended to $n > d$; in Proposition B.2, we give full-rank X for which some $\mathcal{K}_i$ is contained in a subspace of $\mathbb{R}^d$, implying $\mathcal{K}_i - \mathcal{K}_i \subset \mathbb{R}^d$. We call such cones *singular*.

Proposition 3.2. *Suppose $\mathcal{K}_i$ is singular for $D_i \in \mathcal{D}_X$. Then $\exists D_j \in \mathcal{D}_X$ such that $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$ and $\mathcal{K}_i \subset \mathcal{K}_j$.*

That is, every singular cone is contained within a non-singular cone. As a result, we show that these "bad" $\mathcal{K}_i$ can be safely ignored when forming the convex programs.

Theorem 3.3. *Let $\lambda = 0$. For every training set (X, y), there exists $\tilde{\mathcal{D}} \subseteq \mathcal{D}_X$ such that the sub-sampled C-GReLU and C-ReLU problems are both equivalent to the full C-ReLU problem with $\tilde{\mathcal{D}} = \mathcal{D}_X$.*

Algorithm 1 provides a template for training ReLU networks by leveraging cone decompositions and Theorem 3.3. Note that $\tilde{\mathcal{D}}$ is generated by randomly sampling gate vectors $g_i \sim \mathcal{N}(0, I)$. This is sufficient to recover the C-ReLU problem as Theorem 3.3 implies singular cones, for which the sampling probability is zero, don't contribute to the solution.

3.1. Approximating ReLU by Cone Decompositions

We have seen that decomposing $u_i^* = v_i - w_i$ allows us to map the C-GReLU problem into the C-ReLU problem. However, triangle inequality shows $\|u_i^*\|_2 \leq \|v_i\|_2 + \|w_i\|_2$, meaning the cone decomposition can only increase the norm of the model (see Figure 2). For $\lambda > 0$, this increases the penalty term in objective (Eq. 2), although the loss $\mathcal{L}$ is unchanged. This section develops cone decomposition algorithms for which we know the "blow-up" of the norm is not too large. As a result, we obtain approximation guarantees for solving C-ReLU by solving C-GReLU.

In what follows, $\mathcal{K} = \{w : (2D - I)Xw \succeq 0\}$ denotes a non-singular cone, $\tilde{X} = (2D - I)X$, and $\kappa(A)$ is the ratio of the largest and smallest *non-zero* singular values of A. Our first result gives conditions for the existence of a closed-form decomposition.

Proposition 3.4. *Suppose X is full row-rank. If $\mathcal{I} = \{i \in [n] : \langle \tilde{x}_i, u \rangle < 0\}$, then for every $u \in \mathbb{R}^d$,*

$$u = (u + w) - w, \text{where } w = -\tilde{X}_\mathcal{I}^\dagger \tilde{X}_\mathcal{I} u,$$

is a valid decomposition onto $\mathcal{K} - \mathcal{K}$ satisfying,

$$\|u + w\|_2 + \|w\|_2 \leq 2 \|u\|_2 .$$

In general, we cannot hope for constant approximations since $n \gg d$ implies the cones $\mathcal{K}$ are very "narrow".

Algorithm 1 Solving C-ReLU by Cone Decomposition

Input: data (X, y), $\lambda \geq 0$, num. samples p, objective R.
Sample: $\tilde{\mathcal{D}} = \{\text{diag}(\mathbb{1}(Xg_i \geq 0)) : g_i \sim \mathcal{N}(0, I), i \in [p]\}$
Solve C-GReLU:
$u^* \in \arg\min_u \mathcal{L}(\sum_{\tilde{\mathcal{D}}} D_i X u_i, y) + \lambda \sum_{\tilde{\mathcal{D}}} \|u_i\|_2$
Solve Cone Decomposition:
$\bar{v}, \bar{w} \in \arg\min_{v,w} \{R(v, w) : u_i^* = v_i - w_i, i \in [p]\}$
Return: $(\bar{v}, \bar{w})$

Proposition 3.5. *There does not exist a decomposition $u = v - w$, where $v, w \in \mathcal{K}$, such that*

$$\|v\|_2 + \|w\|_2 \leq C \|u\| ,$$

holds for an absolute constant C.

When X is not full row-rank, we can solve

$$\textbf{CD}: \quad \min_{v,w \in \mathcal{K}} \{R(v, w) : v - w = u\} , \tag{5}$$

where $R : \mathbb{R}^{d \times d} \mapsto \mathbb{R}$ is some loss function. Taking $R(v, w) = 0$ reduces to a linear feasibility problem which can be handled by off-the-shelf LP solvers. Choosing $R(v, w) = \|v\|_2 + \|w\|_2$ yields a second-order cone program (SOCP) for which we have the following guarantee.

Proposition 3.6. *For every $u \in \mathbb{R}^d$, if $(\bar{v}, \bar{w})$ is a solution to the cone-decomposition program (5) with $R(v, w) = \|v\|_2 + \|w\|_2$, then there exists $\mathcal{J} \subseteq [n]$ such that*

$$\|\bar{v}\|_2 + \|\bar{w}\|_2 \leq \left(1 + 2\kappa(\tilde{X}_\mathcal{J})\right) \|u\|_2 .$$

Note that the general setting incurs a penalty of $\kappa(\tilde{X}_\mathcal{J})$ compared to Proposition 3.4. Intuitively, this term measures the narrowness of $\mathcal{K}$ and the difficulty of the decomposition. Combining Proposition 3.6 with Proposition 3.2 gives our main approximation result.

Theorem 3.7. *Let $\lambda \geq 0$ and let p^* be the optimal value of the full C-ReLU problem with training set (X, y). There exists $\mathcal{J} \subseteq [n]$ and sub-sampled C-GReLU problem with minimizer u^* and optimal value d^* satisfying,*

$$d^* \leq p^* \leq d^* + 2\lambda\kappa(\tilde{X}_\mathcal{J}) \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2 .$$

As a consequence of Theorem 3.7, Algorithm 1 is guaranteed to approximate the C-ReLU problem if $R(v, w) = \|v\|_2 + \|w\|_2$ and p is sufficiently large. As $\lambda \to 0$, this result smoothly recovers Theorem 3.3, implying we can control the approximation by adjusting the regularization.

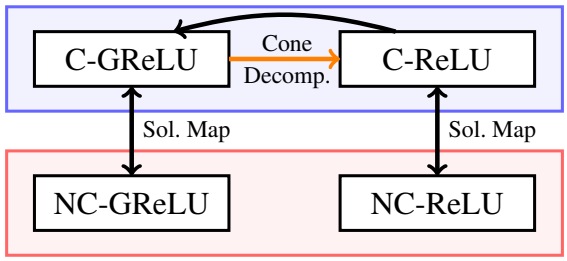

Figure 3. Summary of equivalences between convex (blue) and non-convex (red) neural network training problems with gated ReLU (left) and ReLU (right) activations. The convex programs C-GReLU and C-ReLU are equivalent to the standard non-convex training problems NC-GReLU and NC-GReLU and are related to each other via the cone decomposition procedure.

4. Efficient Global Optimization

We now have two options for convex optimization of ReLU models: directly tackling the C-ReLU problem or solving C-GReLU and a cone decomposition program (see Figure 3). This section develops efficient and scalable methods for both approaches. For simplicity, we assume $\mathcal{L}$ is squared loss; our results are easily extended to other loss functions.

4.1. Solving the Gated ReLU Problem

Our goal is a fast and reliable method for the C-GReLU problem even when $\tilde{\mathcal{D}}$ is very large. To be practical, it should benefit from GPU acceleration, provide convergence certificates, and be "tuning-free". To be theoretically satisfying, it should come with complexity guarantees.

Our starting place is the observation that C-GReLU is exactly the classic *group lasso* with basis expansion,

$$M(X) = [D_1 X \; D_2 X \; \cdots \; D_{|\tilde{\mathcal{D}}|} X].$$

A naive approach to huge-scale group lasso is the stochastic subgradient method; this approach benefits from auto-differentiation engines such as PyTorch (Paszke et al., 2019) and TensorFlow (Abadi et al., 2016) and is simple to code. However, subgradient methods require decreasing step-sizes to converge and are extremely slow — they require $O(\epsilon^{-2})$ iterations to to compute an ϵ-optimal point.

Instead, we use the composite structure of the objective as sum of a convex quadratic $f(u) = \| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i - y \|_2^2$ and the non-smooth penalty $g(u) = \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i\|_2$. The FISTA algorithm (Beck & Teboulle, 2009) is an accelerated method that treats g exactly using the iteration,

$$u_{k+1} = \arg\min_y Q_{y_k,\eta_k}(y) + g(y) \tag{6}$$
$$y_{k+1} = u_{k+1} + \frac{t_k - 1}{t_{k+1}} (u_{k+1} - u_k),$$

where $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$ and

$$Q_{u_k,\eta_k}(y) = f(u_k) + \langle \nabla f(u_k), y - u_k \rangle + \frac{1}{2\eta_k} \|y - u_k\|_2^2,$$

majorizes f as long as $\eta_k \leq \lambda_{\max}(M^\top M)^{-1}$. Using the convergence guarantee for FISTA when f, g are convex and f is Lipschitz smooth (Duchi & Singer, 2009; Beck & Teboulle, 2009; Nesterov, 2013) gives the complexity of *global* optimization of the NC-GReLU problem.

Theorem 4.1. *Let (W_1^*, w_2^*) be the minimum-norm global minimizer of the NC-GReLU problem with gates $\mathcal{G}$. Then, we can compute an ϵ-optimal point $(W_{1\epsilon}, w_{2\epsilon})$ in iterations*

$$T \leq \left(2\epsilon^{-1} \lambda_{max} \left(M^\top M\right) \sum_{D_i \in \tilde{\mathcal{D}}} \|W_{1i}^* w_{2i}^*\|_2^2\right)^{1/2}.$$

Proof in Appendix C. Theorem 4.1 can also be expressed directly in terms of the C-GReLU problem by using a mapping between minimizers of the convex and non-convex formulations. Data normalization gives $\lambda_{\max}\left(M^\top M\right) \leq d \cdot |\tilde{\mathcal{D}}|$, which is fully polynomial when $\text{rank}(X)$ is constant (see Appendix C.1). Such a condition holds for convolutional networks with fixed filter sizes (Pilanci & Ergen, 2020).

4.1.1. Developing an Efficient Optimizer

In theory, it is sufficient to run FISTA with small enough step-size to obtain Theorem 4.1, but this approach works poorly in practice. Additional enhancements are required for fast and reliable solvers.

Line-Search: Constant step-sizes converge slowly, so we use a line-search with the test condition proposed by Beck & Teboulle (2009):

$$f(u_{k+1}(\eta_k)) \leq Q_{y_k,\eta_k}(u_{k+1}(\eta_k)). \tag{7}$$

Computing this condition requires evaluating $f(u_{k+1}(\eta_k))$, but does not need additional gradient evaluations like the alternative proposed by Nesterov (2013). Simple backtracking along $u_{k+1} - u_k$ works poorly and does not converge; instead, we probe the arc of solutions to (6) by reducing the step-size. As evaluating the proximal operator is slower than backtracking, it is important to initialize η_k effectively.

Initializing the Step-size: Warm-starting with $\eta_k = \eta_{k-1}$ can lead to overly-small steps, particularly later in optimization. An alternative is *forward-tracking* as $\eta_k = \alpha \eta_{k-1}$ for $\alpha > 1$ (Fridovich-Keil & Recht, 2019). This can partially adapt to local Lipschitz smoothness of f, but may also lead to unnecessary evaluations of the proximal operator. Instead, we check the tightness of (7) before forward-tracking (Liu et al., 2009). Let $l_{y_k}(u) = f(y_k) + \langle \nabla f(y_k), u - y_k \rangle$ and

$$\omega_k := \frac{\|u_k - y_{k-1}\|_2^2}{2\eta_{k-1}\left(f(u_k) - l_{y_{k-1}}(u_k)\right)}, \tag{8}$$

to get $\eta_k = \eta_{k-1} + (1-\alpha)\eta_{k-1}\mathbb{1}(\omega_k \geq c)$; $c = 1$ obtains forward-tracking, while $c \gg 1$ gives a conservative strategy.

Restarts: Resetting $(y_k, t_k) \leftarrow (u_k, 1)$ in the middle of optimization is called *restarting*. Restarting methods adapt to strong convexity and can attain a fast linear rate of convergence (Nesterov, 2013; Allen Zhu & Orecchia, 2017). Although C-GReLU is not strongly-convex, restarts can allow FISTA to adapt to local curvature (Giselsson & Boyd, 2014). We restart FISTA when $\langle u_{k+1} - u_k, u_{k+1} - y_k \rangle > 0$ — that is, u_{k+1} is not a descent step with respect to the proximal gradient mapping (O'Donoghue & Candès, 2015).

Data Normalization: The proximal step (7) is equivalent to composition of a gradient update with the group soft-thresholding operator. Thresholding is highly sensitive to rounding errors in computation of the gradient and, since errors accumulate in the "memory" y_k, it is critical to improve condition of this computation. Appendix C.1 describes a simple data transformation which works well in practice.

Combining these elements together gives an efficient algorithm for C-GReLU which we call R-FISTA.

4.2. Tractable Cone Decompositions

Training a ReLU network using Algorithm 1 requires solving a large-scale LP or SOCP. Empirically, the complexity of solving these problems with commercial software is similar to directly solving C-ReLU (see Table 1). Instead, we propose an *approximate* decomposition procedure which can be solved efficiently using R-FISTA.

Manipulating the cone decomposition $v - w = u, v, w \in \mathcal{K}$, we obtain the equivalent conditions $\tilde{X}w \geq (-\tilde{X}u)_+$ and $v = u + w$. Given $\rho \geq 0$, and $b = (-\tilde{X}u)_+$, the regularized one-sided quadratic

$$\textbf{CD-A}: \min_w \frac{1}{2}\|(b - \tilde{X}w)_+\|_2^2 + \rho\|w\|_2, \tag{9}$$

approximates the exact cone-decomposition as follows:

Proposition 4.2. *Suppose $\tilde{w}$ is a minimizer of (9) and let $\tilde{v} = u + \tilde{w}$. If X is full row-rank, then*

$$\|(\tilde{X}\tilde{w})_-\|_2 + \|(\tilde{X}\tilde{v})_-\|_2 \leq \frac{2\rho}{\sigma_{min}(\tilde{X})}.$$

Furthermore, if $\rho > 0$, then the norm bound in Proposition 3.6 also holds for the approximate solution $(\tilde{v}, \tilde{w})$.

Alternatively, suppose X is not full row-rank. As $\rho_k \to 0$, every convergent subsequence of $(\tilde{v}_k, \tilde{w}_k)$ is a feasible cone decomposition. Moreover, at least one such sequence exists.

Proof in Appendix C, where we also provide Proposition C.1, an alternative characterization in terms of submatrices $\tilde{X}_{\mathcal{J}}$. Proposition 4.2 shows it is straightforward to control the quality of the approximation by tuning ρ. In practice, we find CD-A with $\rho \approx 10^{-10}$ yields competitive performance and is easily solved using R-FISTA.

4.3. Solving the ReLU Problem

The main difficulty in solving C-ReLU is the constraints. Interior point methods (Nesterov & Nemirovskii, 1994) and specialized conic solvers (O'Donoghue et al., 2016) can handle $\mathcal{K}_i$, but such methods require second-order information or repeated linear-system solves and scale poorly in both n and d. Instead, we develop an augmented Lagrangian (AL) method that uses R-FISTA as a sub-routine.

Recall Theorem 3.3 established a sub-sampled problem equivalent to the full C-ReLU problem for which each $\mathcal{K}_i$ is non-singular. These cones have an interior point if and only if they are non-singular (see Lemma B.1), implying the sub-sampled problem is strictly feasible and satisfies strong duality. Letting $\gamma, \zeta \in \mathbb{R}^{|\tilde{\mathcal{D}}| \times n}$ be estimates of the optimal Lagrange multipliers, the augmented Lagrangian for (2) is

$$\begin{aligned}\mathcal{L}_\delta(v,w,\gamma,\zeta) := (\delta/2) \sum_{D_i \in \tilde{\mathcal{D}}} \big[\|(\gamma_i/\delta - \tilde{X}_i v_i)_+\|_2^2 \\ + \|(\zeta_i/\delta - \tilde{X}_i w_i)_+\|_2^2\big] + F(v,w),\end{aligned} \tag{10}$$

where $F(v,w)$ is the primal objective and $\tilde{X}_i = (2D_i - I)X$. Eq. 10 is a penalty method and can recover an optimal primal-dual pair from $(v_k, w_k) \in \arg\min \mathcal{L}_{\delta_k}(v, w, 0, 0)$ as $\delta_k \to \infty$ (Nocedal & Wright, 1999). However, choosing δ_k is challenging in practice.

Instead, the AL method performs proximal-point iterations on the dual (Rockafellar, 1976b;a) via the iterations,

$$\begin{aligned}(v_{k+1}, w_{k+1}) &= \arg\min_{v,w} \mathcal{L}_\delta(v, w, \gamma_k, \zeta_k), \\ \gamma_{k+1} = (\gamma_k - \delta\tilde{X}_i v_i)_+, &\quad \zeta_{k+1} = (\zeta_k - \delta\tilde{X}_i w_i)_+.\end{aligned} \tag{11}$$

The dual iterates of the AL method converge as $O(1/\delta \cdot \epsilon)$. See Theorem C.3 for a proof going through proximal-point.

4.3.1. Reliable Constrained Optimization

AL methods are typically *exterior point* solvers: (v_k, w_k) will approach the constraint set only as the dual problem is solved. The complexity of maximizing the dual depends on the penalty strength, with $\delta \gg 1$ producing fast convergence. However, δ also affects the Lipschitz smoothness of $\mathcal{L}_\delta$ — large δ increasing the curvature of the (one-sided) quadratic penalties — and can make solving (11) prohibitively expensive for first-order methods. Thus, we choose δ to balance convergence on the primal and dual problems.

Choosing the Penalty Strength: it is common to set δ to aggressively decrease the constraint gap (Conn et al., 2013; Murtagh & Saunders, 1983),

$$c_{\text{gap}} = \sum_{D_i \in \tilde{\mathcal{D}}} \|(\tilde{X}_i v_i)_-|_2^2 + \|(\tilde{X}_i w_i)_-\|_2^2.$$

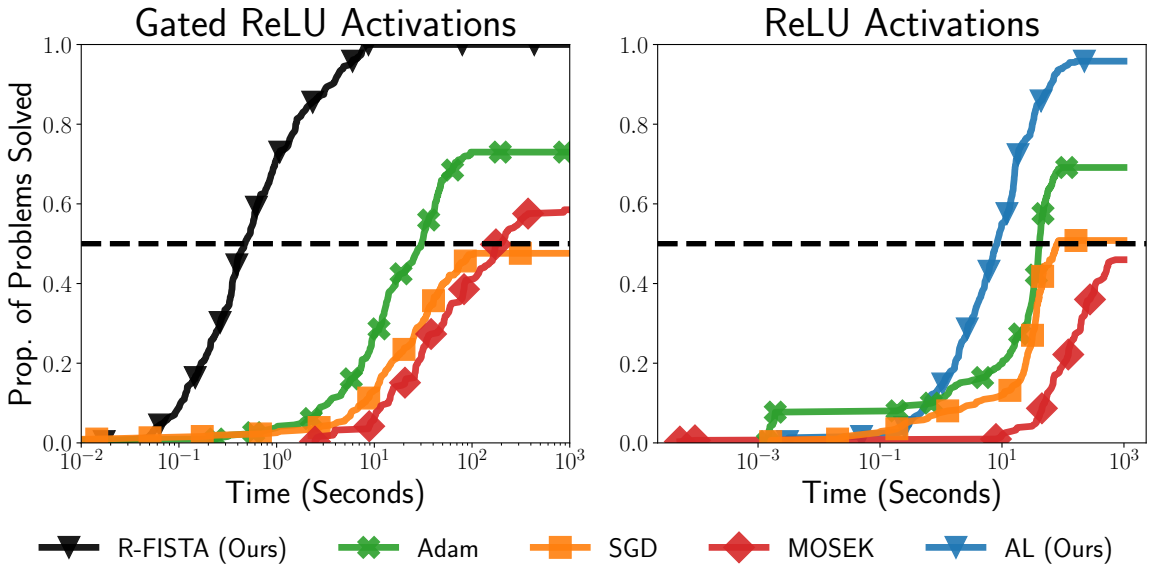

Figure 4. Performance profiles comparing (left) R-FISTA and MOSEK for the C-GReLU problem to Adam and SGD for NC-GReLU, and (right) the AL method and baselines for C-ReLU/NC-ReLU. A problem is solved when $(F(x_k) - F(x^*))/F(x^*) \leq 1$, where $F(x^*)$ is the smallest objective value found by any method. This rule is method-independent as the convex and non-convex problems share the same optimal objective value. See Appendix D.2 for alternative thresholds. Methods are judged by comparing time to a fixed proportion of problems solved (see dashed line at 50%). R-FISTA and the AL method solve a higher proportion of problems faster than the baselines.

These rules pre-suppose second-order solvers and lead to very poor conditioning of $\mathcal{L}_\delta$. Instead, we propose a simple "windowing" heuristic: when solving Eq. 11 for (γ_1, ζ_1), take δ to ensure that $c_{\text{gap}} \in [r_\text{l}, r_\text{u}]$. This condition can be checked and enforced with minimal overhead by using a mild convergence criterion initially and helps avoid extreme behavior. We found $[r_\text{l}, r_\text{u}] = [0.01, 0.1]$ works well.

Warm Starts: The contours of $\mathcal{L}_\delta(\cdot, \cdot, \gamma_{k+1}, \zeta_{k+1})$ typically change slowly when δ is moderate. In such cases, minimizing the augmented Lagrangian can be greatly sped-up by warm-starting with (v_k, w_k).

We obtain an efficient and robust AL method by combining warm-starts, our heuristic for δ, and the R-FISTA sub-solver.

5. Experiments

We now present experiments validating our optimizers. We show that training neural networks via convex reformulations is faster and more robust than attempting to solve the non-convex training problem with SGD (Robbins & Monro, 1951) or Adam (Kingma & Ba, 2015). Moreover, the models learned by convex optimization are consistent and generalize as well as Adam/SGD without their failure modes.

5.1. Optimization Performance

Synthetic Classification: Convex-reformulations offer a stable approach to model training, especially outside of the over-parameterized setting. To illustrate this, we create a realizable problem with $X \sim \mathcal{N}(0, \Sigma)$ and $y = \text{sign}(h_{W_1,w_2}(X))$, where h_{W_1,w_2} is a two-layer ReLU network with $m = 100$ and random Gaussian weights. We try to recover this model with ten independent runs of SGD and compare against our AL method on the C-ReLU problem. For C-ReLU, $\tilde{\mathcal{D}}$ is 100 random arrangements augmented with all activations generated while solving the non-convex problem with SGD.[2] Figure 1 shows that SGD converges to sub-optimal stationary points four times, while every run of the convex solver yields a model with perfect training accuracy. See Appendix D.1 for additional results.

Table 1. Approximating the C-ReLU problem with cone decompositions. We compare the solution to C-GReLU (FISTA) with cone-decomposition by solving the min-norm program (CD-SOCP), the approximate cone decomposition (CD-A), and directly solving C-ReLU using the AL method. Exactly solving CD-SOCP is costly compared to direct solutions. Although CD-A gives only an approximate decomposition, it yields similar test performance to CD-SOCP and is two orders of magnitude faster.

	R-FISTA		**CD-SOCP**		**CD-A**		**AL**	
Dataset	Acc.	Time	Acc.	Time	Acc.	Time	Acc.	Time
energy	86.3	0.12	86.3	134.6	86.3	1.56	83.7	5.05
ecoli	71.6	0.07	71.6	149.7	70.1	0.29	70.1	3.38
glass	64.3	0.13	64.3	68.76	64.3	0.57	61.9	3.0
pima	73.2	0.36	73.2	37.68	73.2	4.24	75.8	4.72
oocytes	78.6	0.98	79.1	136.3	78.0	4.68	74.2	81.68

Large-Scale Comparison: Figure 4 presents two perfor-

[2] This guarantees the non-convex model is in the model space of the convex program.

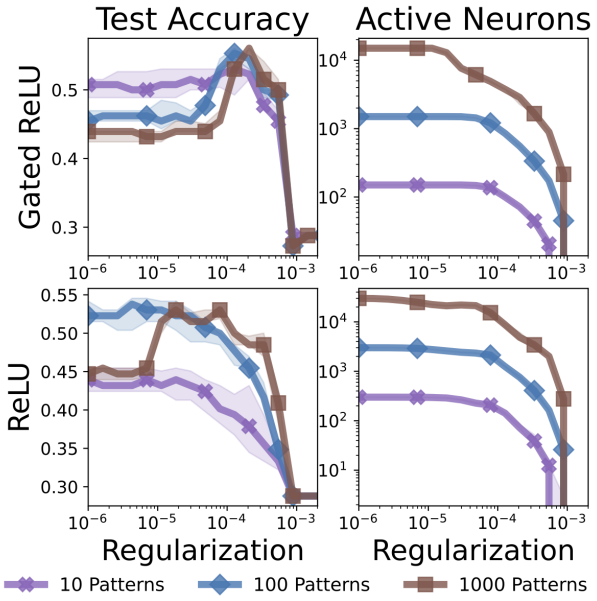

Figure 5. Effect of sampling activation patterns on test accuracy for networks trained using the C-ReLU and C-GReLU problems on the `primary-tumor` dataset. We consider a grid of regularization parameters and plot median (solid line) and first and third quartiles (shaded region) over 10 random samplings of $\tilde{\mathcal{D}}$, where $|\tilde{\mathcal{D}}|$ is limited to 10, 100, or 1000 patterns.

mance profiles (Dolan & Moré, 2002) comparing the optimization performance of R-FISTA and our AL method to Adam, SGD, and the interior-point solver MOSEK (ApS, 2019). MOSEK solves the convex reformulations, while Adam and SGD solve the original non-convex problems. The profiles aggregate performance on 438 problems generated by considering six regularization parameters for 73 datasets taken from the UCI repository (Dua & Graff, 2017). We use the default parameters for MOSEK; for Adam and SGD, we use a batch-size of 10% of the data and take the *best* run per-problem over a grid of seven step-sizes and three different random seeds. See Appendix D.2 for details.

We make the following observations: (i) R-FISTA solves 50% of problems two orders of magnitude faster than Adam and SGD; (ii) MOSEK scales poorly and frequently runs out of memory despite being allocated 32GB — 3× more than the other solvers; (iii) although the ReLU problem is significantly harder, the AL solver converges faster and solves 25% more problems than the best baseline.

Cone Decompositions: We compare optimizing the C-ReLU problem directly using our AL method against Algorithm 1. We try two decomposition methods: CD-SOCP, which sets $R(u, v) = \|u\|_2 + \|v\|_2$ and solves the resulting SOCP, and CD-A, which approximates the cone decomposition problem by solving Eq. (9). We use MOSEK to solve the SOCP. Table 1 gives median test accuracy and time-to-solution for each approach on five UCI datasets.

Table 2. Test accuracies for our convex solvers, random forests (RF), SVMs with a linear kernel (Linear) and SVMs with an RBF kernel (RBF) for binary classification on 18 UCI datasets. C-GReLU and C-ReLU both obtain the best test accuracy on 9 datasets, while the most competitive baseline is best on just 4.

Dataset	C-GReLU	C-ReLU	RF	Linear	RBF
blood	79.9	**80.5**	75.8	74.5	77.9
chess-krvkp	**99.2**	98.6	98.9	97.2	98.4
conn-bench	**90.2**	85.4	73.2	68.3	85.4
cylinder-bands	76.5	**78.4**	77.5	71.6	71.6
fertility	**80.0**	**80.0**	75.0	75.0	75.0
heart-hung.	**86.2**	**86.2**	84.5	84.5	**86.2**
hill-valley	**76.0**	68.6	57.9	62.0	70.2
ilpd-liver	72.4	**74.1**	66.4	71.6	71.6
mammographic	77.6	78.6	**80.7**	**80.7**	80.2
monks-1	**100**	**100**	95.8	79.2	83.3
musk-1	94.7	**95.8**	92.6	86.3	**95.8**
ozone	**97.6**	**97.6**	97.4	97.2	97.4
pima	74.5	74.5	**76.5**	75.2	73.2
planning	**69.4**	63.9	66.7	66.7	**69.4**
spambase	93.5	93.6	**94.1**	92.2	93.6
spectf	**87.5**	75.0	68.8	68.8	68.8
statlog-german	74.0	**77.5**	73.5	75.0	75.5
tic-tac-toe	99.0	99.0	99.5	98.4	**100**

R-FISTA is an order of magnitude faster than AL and two orders faster CD-SOCP, primarily because SOCPs must be solved on CPU. CD-A performs comparably to CD-SOCP and is faster than solving the C-ReLU problem with our AL method. See Appendix D.3 for experimental details and additional results, including model norms.

5.2. Model Performance

Sensitivity and Regularization: Figure 5 shows the effects of sub-sampling activation patterns on the C-ReLU and C-GReLU problems for the `primary-tumor` dataset. Surprisingly, we find that the distribution of test-accuracies is stable across regularization parameters even when the number of patterns is small. We also observe an inverted-U shaped bias-variance trade-off as the regularization strength is increased, with sparse models showing the best generalization. This contrasts the double descent phenomena frequently observed with non-convex neural networks (Belkin et al., 2019; Loog et al., 2020; Nakkiran et al., 2020). See Appendix E for results on a further nine UCI datasets.

UCI Classification: Table 2 compares the performance of C-ReLU and C-GReLU with random forests (Breiman, 2001) and SVMs (Boser et al., 1992) for binary classification on 18 UCI datsasets. For all methods, we report test accuracy for the best hyperparameters as selected by cross-validation. Taken together, C-ReLU and C-GReLU perform best on 14 problems, showing two-layer neural networks offer an effective, easy-to-train alternative to common baselines. Results for additional datasets are given in

Table 3. Median test accuracies from five restarts on a subset of the UCI datasets. Results are presented as `Gated ReLU` / `ReLU`. Overall, we find the convex reformulations have comparable generalization to the non-convex networks. Note the catastrophic failure of SGD on `ecoli`. See Appendix E.2 for quartiles.

Dataset	Convex	Adam	SGD
magic	86.9 / 85.9	82.9 / 86.9	82.1 / 86.4
statlog-heart	79.6 / 83.3	85.2 / 83.3	83.3 / 79.6
mushroom	100 / 100	97.6 / 100	96.9 / 99.9
vertebral-col.	87.1 / 90.3	90.3 / 90.3	90.3 / 88.7
cardiotocogr.	90.1 / 89.9	85.6 / 36.5	85.2 / 88.9
abalone	63.8 / 66.2	58.7 / 65.3	58.1 / 66.1
annealing	90.6 / 90.6	86.2 / 93.7	86.2 / 88.7
car	89.9 / 87.8	83.8 / 94.8	83.2 / 90.1
bank	89.8 / 89.8	89.9 / 90.8	89.8 / 90.5
breast-cancer	68.4 / 68.4	68.4 / 64.9	70.2 / 68.4
page-blocks	96.8 / 94.0	92.1 / 97.1	92.4 / 96.9
contrac	45.9 / 55.1	53.1 / 54.4	53.4 / 53.7
congressional	63.2 / 63.2	64.4 / 62.1	66.7 / 67.8
spambase	93.4 / 93.3	91.6 / 93.5	91.2 / 93.2
synthetic	97.5 / 98.3	98.3 / 96.7	97.5 / 96.7
musk-1	93.7 / 93.7	93.7 / 96.8	94.7 / 95.8
ringnorm	69.8 / 77.0	77.0 / 77.3	77.2 / 77.4
ecoli	82.1 / 80.6	79.1 / 82.1	4.5 / 80.6
monks-2	69.7 / 69.7	66.7 / 69.7	60.6 / 72.7
hill-valley	62.0 / 65.3	57.0 / 62.8	58.7 / 55.4

Appendix E.1

Non-Convex Solvers: We compare the generalization of C-ReLU and C-GReLU with that of the non-convex problems on 20 UCI datasets. For each dataset/problem, we select the regularization strength using five-fold cross validation. For NC-ReLU and NC-GReLU, we use Adam and SGD and tune the step-sizes by cross-validation. See Appendix E.2 for details. Table 2 summarizes the test accuracy results. We find that our convex programs generalize as well as the non-convex baselines for a fraction of the training time.

Image Classification: We study the generalization performance of the Gated ReLU model for image classification on the MNIST and CIFAR-10 datasets (LeCun et al., 1998; Krizhevsky et al., 2009). We compare R-FISTA for C-GReLU to solving the NC-GReLU problem with SGD, Adam, and Adagrad (Duchi et al., 2011). We choose the regularization strength and step sizes for each dataset-method pair using a train/validation split (see Appendix F). Table 4 shows that R-FISTA scales well to these large-scale experiments, with generalization comparable with the non-convex solvers. This reflects our theory, which shows that these methods are fundamentally solving the same problem.

5.3. Additional Experiments

We defer additional experiments to the supplementary material due to space constraints. In Appendix D.4, we study the effects of acceleration, restarts, and line-search on the performance of the R-FISTA method and conclude that all three components are key to the efficiency of the optimization procedure. Appendix D.5 presents an ablation study for the step-size initialization procedure in R-FISTA, which is shown to be robust to the choice of c. Similarly, Appendix D.6 examines the windowing heuristic for the penalty strength in our AL method and shows the strategy is comparable to the best fixed δ found by grid search.

Table 4. Test accuracy of R-FISTA for the C-GReLU problem compared to SGMs for NC-GReLU on two image classification tasks. Models trained using the convex program have comparable test accuracy to the non-convex formulation on MNIST and are slightly better on CIFAR-10.

Dataset	Convex	Adam	SGD	Adagrad
MNIST	97.6	98.0	97.2	97.5
CIFAR-10	56.4	50.1	54.3	54.2

6. Conclusion

We propose optimization algorithms for convex reformulations of two-layer neural networks with ReLU activations. By studying the problem constraints, we split the space of ReLU activations into "singular" patterns, which may be safely ignored, and non-singular patterns. As a result, we show that ReLU networks can be trained by decomposing the solution to an unconstrained Gated ReLU training problem onto a difference of polyhedral cones. Experimentally, we test our algorithms on more than 70 different datasets, demonstrating that convex optimization is faster and more reliable than popular training methods like Adam and SGD.

Many directions are left to future work. Efficiently solving the cone decomposition problem is key to improving on our augmented Lagrangian method, but existing conic solvers rely on CPU computation. We believe developing methods which can natively leverage GPU acceleration is necessary. Finally, we hope to extend convex optimization to deeper networks by layer-wise training, which has been shown to perform well on ImageNet (Belilovsky et al., 2019).

Acknowledgements

This work was partially supported by an Army Research Office Early Career Award, and the National Science Foundation under grants ECCS-2037304, DMS-2134248. Aaron Mishkin was supported by the NSF Graduate Research Fellowship Program, Grant No. DGE-1656518 and the NSERC PGS D program, Grant No. PGSD3-547242-2020. Arda Sahiner was supported by the National Institutes of Health, Grant No. R01EB009690, U01EB029427. Computational resources were provided by the Stanford Research Computing Center. We would like to thank Frederik Kunstner and Tolga Ergen for many insightful discussions.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. TensorFlow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Allen Zhu, Z. and Orecchia, L. Linear Coupling: An ultimate unification of gradient and mirror descent. In Papadimitriou, C. H. (ed.), *8th Innovations in Theoretical Computer Science Conference, ITCS 2017*, volume 67 of *LIPIcs*, pp. 3:1–3:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155. PMLR, 2017.

ApS, M. *MOSEK Optimizer API for Python 9.3.6*, 2019. URL https://docs.mosek.com/latest/pythonapi/index.html.

Ausubel, L. M. and Deneckere, R. J. A generalized theorem of the maximum. *Economic Theory*, 3(1):99–107, 1993.

Bach, F. R. Breaking the curse of dimensionality with convex neural networks. *J. Mach. Learn. Res.*, 18:19:1–19:53, 2017.

Bai, Y., Gautam, T., and Sojoudi, S. Efficient global optimization of two-layer ReLU networks: Quadratic-time algorithms and adversarial training. *arXiv preprint arXiv:2201.01965*, 2022.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

Belilovsky, E., Eickenberg, M., and Oyallon, E. Greedy layerwise learning can scale to ImageNet. In *International conference on machine learning*, pp. 583–593. PMLR, 2019.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In Montavon, G., Orr, G. B., and Müller, K. (eds.), *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pp. 437–478. Springer, 2012.

Bengio, Y., Roux, N. L., Vincent, P., Delalleau, O., and Marcotte, P. Convex neural networks. *Advances in neural information processing systems*, 18:123, 2006.

Bertsekas, D. *Convex optimization theory*, volume 1. Athena Scientific, 2009.

Bertsekas, D. P. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

Bertsekas, D. P. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

Bhojanapalli, S., Wilber, K., Veit, A., Rawat, A. S., Kim, S., Menon, A. K., and Kumar, S. On the reproducibility of neural network predictions. *CoRR*, abs/2102.03349, 2021.

Birgin, E. G. and Martínez, J. M. *Practical augmented Lagrangian methods for constrained optimization*, volume 10 of *Fundamentals of algorithms*. SIAM, 2014.

Blum, A. and Rivest, R. L. Training a 3-node neural network is NP-Complete. In Touretzky, D. S. (ed.), *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pp. 494–501. Morgan Kaufmann, 1988.

Boser, B. E., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In Haussler, D. (ed.), *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, pp. 144–152. ACM, 1992.

Breiman, L. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

Conn, A. R., Gould, G., and Toint, P. L. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, volume 17. Springer Science & Business Media, 2013.

Delgado, M. F., Cernadas, E., Barro, S., and Amorim, D. G. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1): 3133–3181, 2014.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Dolan, E. D. and Moré, J. J. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Duchi, J. C. and Singer, Y. Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 10:2899–2934, 2009.

Duchi, J. C., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

Ergen, T. and Pilanci, M. Global optimality beyond two layers: Training deep ReLU networks via convex programs. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2993–3003. PMLR, 2021a.

Ergen, T. and Pilanci, M. Implicit convex regularizers of CNN architectures: Convex optimization of two- and three-layer networks in polynomial time. In *International Conference on Learning Representations: ICLR 2021*, 2021b.

Ergen, T. and Pilanci, M. Revealing the structure of deep neural networks via convex duality. In *International Conference on Machine Learning*, pp. 3004–3014. PMLR, 2021c.

Ergen, T., Sahiner, A., Ozturkler, B., Pauly, J. M., Mardani, M., and Pilanci, M. Demystifying batch normalization in relu networks: Equivalent convex optimization models and implicit regularization. In *International Conference on Learning Representations*, 2021.

Fiat, J., Malach, E., and Shalev-Shwartz, S. Decoupling gating from linearity. *arXiv preprint arXiv:1906.05032*, 2019.

Fridovich-Keil, S. and Recht, B. Choosing the step size: Intuitive line search algorithms with efficient convergence. In *The 11th Workshop on Optimization for Machine Learning (OPT 2019)*, 2019.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle points - online stochastic gradient for tensor decomposition. In Grünwald, P., Hazan, E., and Kale, S. (eds.), *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pp. 797–842. JMLR.org, 2015.

Giselsson, P. and Boyd, S. P. Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pp. 5058–5063. IEEE, 2014.

Goberna, M. A. and López, M. A. Linear semi-infinite programming theory: An updated survey. *Eur. J. Oper. Res.*, 143(2):390–405, 2002. doi: 10.1016/S0377-2217(02)00327-2. URL https://doi.org/10.1016/S0377-2217(02)00327-2.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Güler, O. On the convergence of the proximal point algorithm for convex minimization. *SIAM journal on control and optimization*, 29(2):403–419, 1991.

Gupta, V., Bartan, B., Ergen, T., and Pilanci, M. Exact and relaxed convex formulations for shallow neural autoregressive models. In *International Conference on Acoustics, Speech, and Signal Processing*, 2021.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3207–3214. AAAI Press, 2018.

Huang, G., Zhu, Q., and Siew, C. K. Extreme learning machine: Theory and applications. *Neurocomputing*, 70 (1-3):489–501, 2006.

Kang, M., Kang, M., and Jung, M. Inexact accelerated augmented Lagrangian methods. *Comput. Optim. Appl.*, 62(2):373–404, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H. and Lin, Z. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28:379–387, 2015.

Liu, J., Chen, J., and Ye, J. Large-scale sparse logistic regression. In IV, J. F. E., Fogelman-Soulié, F., Flach, P. A., and Zaki, M. J. (eds.), *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pp. 547–556. ACM, 2009.

Loog, M., Viering, T., Mey, A., Krijthe, J. H., and Tax, D. M. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

Murtagh, B. A. and Saunders, M. A. MINOS 5.0 user's guide. Technical report, Stanford Univ CA Systems Optimization Lab, 1983.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady an USSR*, volume 269, pp. 543–547, 1983.

Nesterov, Y. E. Gradient methods for minimizing composite functions. *Math. Program.*, 140(1):125–161, 2013.

Nesterov, Y. E. and Nemirovskii, A. *Interior-point polynomial algorithms in convex programming*, volume 13 of *Siam studies in applied mathematics*. SIAM, 1994.

Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. Geometry of optimization and implicit regularization in deep learning. *CoRR*, abs/1705.03071, 2017.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, 1999.

O'Donoghue, B. and Candès, E. J. Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 15 (3):715–732, 2015.

O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016. URL http://stanford.edu/~boyd/papers/scs.html.

Parikh, N. and Boyd, S. P. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, 2014.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Pilanci, M. and Ergen, T. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7695–7705. PMLR, 2020.

Robbins, H. and Monro, S. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

Rockafellar, R. T. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 1(2):97–116, 1976a.

Rockafellar, R. T. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976b.

Sahiner, A., Ergen, T., Ozturkler, B., Bartan, B., Pauly, J., Mardani, M., and Pilanci, M. Hidden convexity of wasserstein gans: Interpretable generative models with closed-form solutions. *International Conference on Learning Representations*, 2021a.

Sahiner, A., Ergen, T., Pauly, J. M., and Pilanci, M. Vector-output ReLU neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b.

Sahiner, A., Mardani, M., Ozturkler, B., Pilanci, M., and Pauly, J. M. Convex regularization behind neural reconstruction. In *ICLR*, 2021c.

Schmidt, M., Le Roux, N., and Bach, F. R. Convergence rates of inexact proximal-gradient methods for convex optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24: NeurIPS 2011*, pp. 1458–1466, 2011.

Sivaprasad, S., Singh, A., Manwani, N., and Gandhi, V. The curious case of convex neural networks. In Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., and Lozano, J. A. (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part I*, volume 12975 of *Lecture Notes in Computer Science*, pp. 738–754. Springer, 2021.

Sra, S., Nowozin, S., and Wright, S. J. *Optimization for machine learning*. Mit Press, 2012.

Wang, Y., Lacotte, J., and Pilanci, M. The hidden convex optimization landscape of regularized two-layer relu networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*, 2021.

Woodworth, B. E., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In Abernethy, J. D. and Agarwal, S. (eds.), *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3635–3673. PMLR, 2020.

A. Convex Reformulations: Proofs

Lemma A.1. *The non-convex problem NC-ReLU (Problem 1) is equivalent to the mixed-integer program,*

$$\begin{aligned} \min_{W_1, w_2} \; & \mathcal{L}\Big(\sum_{i=1}^{m} (XW_{1i})_+ w_{2i}, y \Big) + \lambda \sum_{i=1}^{m} \|W_{1i}\|_2 \\ & \text{s.t. } w_2 \in \{-1, 1\}^m . \end{aligned} \tag{12}$$

Proof. The proof proceeds in two steps: first we transform the objective into an equivalent problem which is invariant to certain scale re-parameterizations of the network parameters. Then, we use these scale re-parameterizations reduce the two optimization problems to each-other.

Let p^* be the optimal value of the non-convex optimization problem and d^* the optimal value of the mixed-integer program. The ReLU activation function,

$$(a)_+ = \max\{a, 0\},$$

is positively homogeneous, meaning $(a * \beta)_+ = \beta (a)_+$ for any scalar $\beta \geq 0$. Defining $W'_{1i} = \beta_i W_{1i}$, $w'_{2i} = w_{2i}/\beta_i$, $i \in [m]$, we have

$$\begin{aligned} h_{W_1, w_2}(X) &= \sum_{i=1}^{m} (XW_{1i})_+ w_{2i} = \sum_{i=1}^{m} \left(XW_{1i} \frac{\beta_i}{\beta_i} \right)_+ w_{2i} \\ &= \sum_{i=1}^{m} (XW'_{1i})_+ w'_{2i} = h(W'_1, w'_2), \end{aligned}$$

implying that the loss $\mathcal{L}(\sum_{i=1}^{m} (XW_{1i})_+ w_{2i}, y)$ is invariant to "scale-shift" re-parameterizations of this form. To extend the invariance to the full objective function, recall Young's inequality,

$$2\langle a, b \rangle \leq a^2 + b^2,$$

which yields,

$$\mathcal{L}\left(\sum_{i=1}^{m} (XW_{1i})_+ w_{2i}, y \right) + \frac{\lambda}{2} \sum_{i=1}^{m} \|W_{1i}\|_2 + |w_{2i}|^2 \geq \mathcal{L}\left(\sum_{i=1}^{m} (XW_{1i})_+ w_{2i}, y \right) + \lambda \sum_{i=1}^{m} \|W_{1i}\| \, |w_{2i}| .$$

For any choice of parameters W_{1i}, w_{2i}, equality in this expression is achieved with the rescaling

$$W'_{1i} = W_{1i} * \beta_i, \qquad w'_{2i} = w_{2i}/\beta_i,$$

where $\beta_i = \sqrt{\frac{w_{2i}}{\|W_{1i}\|_2}}$. As this rescaling does not affect h_{W_1, w_2}, it must be that any global minimizer $\theta^* = (W_1^*, w_2^*)$ of Problem 1 achieves the lower-bound in Young's inequality and,

$$\mathcal{L}\left(\sum_{i=1}^{m} (XW_{1i}^*)_+ w_{2i}^*, y \right) + \frac{\lambda}{2} \sum_{i=1}^{m} \|W_{1i}^*\|_2^2 + |w_{2i}^*|^2 = \mathcal{L}\left(\sum_{i=1}^{m} (XW_{1i}^*)_+ w_{2i}^*, y \right) + \lambda \sum_{i=1}^{m} \|W_{1i}^*\|_2 \, |w_{2i}^*| . \tag{13}$$

The right-hand side of this equation is invariant to scale re-parameterizations of the form $W'_{1i} = \beta W_{1i}^*, w'_{2i} = w_{2i}^*/\beta$ for $\beta > 0$. Taking $\beta = |w_{2i}|$, we deduce

$$p^* = \mathcal{L}\left(\sum_{i=1}^{m} (XW'_{1i})_+ w'_{2i}, y \right) + \lambda \sum_{i=1}^{m} \|W'_{1i}\|_2 \geq d^*.$$

To show the reverse inequality, observe that every global minimum (W_1^*, w_2^*) of Problem 12 is trivially in the domain of the non-convex ReLU training problem. Using the mapping

$$(W'_{1i}, w'_{2i}) = \left(\frac{W_{1i}^*}{\sqrt{\|W_{1i}^*\|_2}}, w_{2i}^* \sqrt{\|W_{1i}^*\|_2} \right),$$

and plugging (W_1', w_2') into Problem 1 shows $d^* \geq p^*$. We have shown $p^* = d^*$ and so the problems are formally equivalent with mappings between the solutions as given above.

□

Theorem 2.1. *Suppose (W_1^*, w_2^*) and (v^*, w^*) are global minima of the NC-ReLU (12) and C-ReLU (2) problems, respectively. If the number of hidden units satisfies*

$$m \geq b := \sum_{D_i \in \tilde{\mathcal{D}}} \left| \{v_i^* : v_i^* \neq 0\} \cup \{w_i^* : w_i^* \neq 0\} \right|,$$

and the optimal activations are in the convex model,

$$\{\text{diag}\left(XW_{1i}^* \geq 0 : i \in [m]\right)\} \subseteq \tilde{\mathcal{D}},$$

then the two problems have same the optimal value.

Proof. The proof proceeds by showing the equivalence of C-ReLU and the mixed integer problem given in Equation (12) and the invoking Lemma A.1. Let p^* be the optimal value of the mixed-integer problem in (12) and d^* the optimal value of the convex program in (2). We first show that $d^* \geq p^*$.

Suppose (v^*, w^*) is a global minimizer of Problem 2 and let

$$\{(W_{1k}^*, w_{2k}^*)\} = \bigcup_{D_i \in \tilde{\mathcal{D}}_i} \{(v_i^*, 1) : v_i^* \neq 0\} \cup \{(w_i^*, -1) : w_i^* \neq 0\},$$

where we set $W_{1k}^* = 0$, and $w_{2k}^* = 0$ for all $k \in [m], k > b$. It holds by assumption that $b \leq m$ and thus (W_1^*, w_2^*) is a valid input for the mixed-integer problem.

Recalling the constraints $(2D_i - I)Xv_i^* \geq 0$, and $(2D_i - I)Xw_i^* \geq 0$, we see that $D_iXv_i^* = (Xv_i^*)_+$, $D_iXw_i^* = (Xw_i^*)_+$, and thus

$$\left(XW_{1k}^*\right) w_{2j} = \begin{cases} D_iXv_i^* & \text{if } W_{1k}^* = v_i^* \text{ for some } i \in [b] \\ -D_iXw_i^* & \text{if } W_{1k}^* = w_i^* \text{ for some } i \in [b] \\ 0 & \text{otherwise.} \end{cases}$$

Using this fact in the optimization objective for the convex program, we find

$$\begin{aligned} d^* &= \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_iX(v_i - w_i), y\right) + \lambda \sum_{D_i \in \mathcal{D}} \|v_i\|_2 + \|w_i\|_2 \\ &= \mathcal{L}\left(\sum_{k=1}^{m} \left(XW_{1k}^*\right)_+ w_{2k}^* - y\right) + \lambda \sum_{k=1}^{m} \|W_{1k}^*\|_2 \\ &\geq p^*, \end{aligned}$$

as required.

To show the reverse inequality, let (W_1^*, w_2^*) be a solution to (12) and consider the set-function

$$T(j) = \{i \in [m] : \text{diag}\left(XW_{1i}^* > 0\right) = D_j\}.$$

Recalling $\{\text{diag}\left(XW_{1i}^* > 0 : i \in [m]\right)\} \subseteq \tilde{\mathcal{D}}$, by assumption, we define a valid candidate solution as

$$\left\{\left(v_j^*, w_j^*\right)\right\}_{D_j \in \tilde{\mathcal{D}}} = \left\{ \sum_{i \in T(j)} W_{1i}^* \mathbb{1}(w_{2i}^* = 1), \sum_{i \in T(j)} W_{1i}^* \mathbb{1}(w_{2i}^* = -1) \right\}_{D_j \in \tilde{\mathcal{D}}}$$

We start by showing that the neurons indexed by $T(j)$ can be merged without changing the objective of the mixed-integer problem. In particular, let $j \in [m]$ be arbitrary and suppose that there exists $l, k \in T(j)$ such that $w_{2l}^* = w_{2k}^* = 1$. By definition of T, it holds that $(XW_{1l}^*)_+$ and $(XW_{1k}^*)_+$ have the same activation pattern. Accordingly, we have $(XW_{1l}^*)_+ + (XW_{1k}^*)_+ = (X(W_{1l}^* + W_{1k}^*))_+$ by definition of the ReLU activation. Thus, merging these two parameter vectors as $Z_{lk}^* = W_{1l}^* + W_{1k}^*$ does not change the prediction of the model in the mixed-integer program.

Now we consider the group ℓ_1 penalty term. Triangle inequality implies

$$\left\|\tilde{Z}_{lk}^*\right\|_2 \leq \|W_{1l}^*\|_2 + \|W_{1k}^*\|_2 ,$$

with equality if and only if $W_{1l}^* = 0$, $W_{1k}^* = 0$, or the vectors are collinear. Suppose that equality does not hold. Then the penalty term could be reduced setting $W_{1l}^* = Z_{lk}$ and $W_{1k}^* = 0$ while leaving the squared-loss term unchanged. But, this contradicts global optimality of W_1^*, w_2^*. Thus, it must be that $W_{1l}^* = 0$, $W_{1k}^* = 0$, or the vectors are collinear. In each case, we have that the merged vector $\tilde{Z}_{lk}^*$ also attains the optimal value p^*. Clearly a symmetric argument holds in the case $w_{2k} = w_{2l} = -1$.

Arguing by induction if necessary, we deduce that the vectors v^*, w^* given by the solution mapping also attain p^*. Recalling that $D_j X v_j^* = (Xv_j)_+$ and $D_j X w_j^* = (Xw_j)_+$ by choice of $T(j)$ and definition of D_j gives

$$\begin{aligned} p^* &= \mathcal{L}\left(\sum_{i=j}^{P} D_j X(v_j^* - w_j^*) - y\right) + \lambda \sum_{j=1}^{P} \left\|v_j^*\right\|_2 + \left\|w_j^*\right\|_2 \\ &\geq d^*, \end{aligned}$$

where v_j^*, w_j^* are feasible. This completes the proof. □

Theorem 2.2. *Let $g_i \in R^d$ such that diag$(Xg_i \geq 0) = D_i$ and $\tilde{\mathcal{G}} = \left\{g_i : D_i \in \tilde{\mathcal{D}}\right\}$. Then, C-GReLU is equivalent to the following gated ReLU problem (NC-GReLU):*

$$\min_{W_1, w_2} \mathcal{L}\Big(\sum_{g_i \in \tilde{\mathcal{G}}} \phi_{g_i}(X, W_{1i}) w_{2i}, y\Big) + \frac{\lambda}{2} \sum_{g_i \in \tilde{\mathcal{G}}} \|W_{1i}\|_2^2 + w_{2i}^2. \tag{4}$$

Proof. The proof proceeds similarly to the proof Lemma A.1.

Let p^* be the optimal value of the Problem 4 and let d^* be the optimal value of the C-GReLU problem (3). For each $z_i \in \tilde{\mathcal{Z}}$ and any $v \in \mathbb{R}^d$, we have the following equality by construction:

$$\phi_{z_i}(X, v) = (Xz_i > 0) \circ Xv = D_i X v.$$

Accordingly, the non-convex optimization problem (4) can be written as

$$\min_{W_1, w_2} \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X W_{1i} w_{2i}, y\Big) + \frac{\lambda}{2} \sum_{z_i \in \tilde{\mathcal{Z}}} \|W_{1i}\|_2^2 + w_{2i}^2, \tag{14}$$

which makes the connection to C-GReLU clear. Applying Young's inequality gives,

$$\mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X W_{1i} w_{2i}, y\Big) + \frac{\lambda}{2} \sum_{z_i \in \tilde{\mathcal{Z}}} \|W_{1i}\|_2^2 + w_{2i}^2 \geq \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X W_{1i} w_{2i}, y\Big) + \lambda \sum_{z_i \in \tilde{\mathcal{Z}}} \|W_{1i}\|_2 |w_{2i}|$$

For any choice of parameters W_{1i}, w_{2i}, equality in this expression is achieved with the rescaling

$$W_{1i}' = W_{1i} * \beta_i, \qquad w_{2i}' = w_{2i}/\beta_i,$$

where $\beta_i = \sqrt{\frac{w_{2i}}{\|W_{1i}\|_2}}$. As this rescaling does not affect $D_i X W_{1i}\, w_{2i}$ for each i, it must be that any global minimizer $\theta^* = \{W_{1i}^*, w_{2i}^*\}$ of Problem 4 achieves the lower-bound in Young's inequality. Defining $v_i' = W_{1i} * w_{2i}$, we have shown

$$\begin{aligned} p^* &= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X W_{1i} w_{2i}, y\Big) + \lambda \sum_{z_i \in \tilde{\mathcal{Z}}} \|W_{1i}\|_2 \, |w_{2i}| \\ &= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i', y\Big) + \lambda \sum_{z_i \in \tilde{\mathcal{Z}}} \|v_i'\|_2 \geq d^*, \end{aligned}$$

where we have used absolute homogeneity of the norm.

To obtain the reverse inequality, let v^* be a global minimizer of C-GReLU and define $W_{1i}' = \frac{v_i^*}{\sqrt{\|v_i^*\|_2}}$, $w_{2i}' = \sqrt{\|v_i^*\|_2}$ for all i to obtain

$$\begin{aligned} d^* &= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 \\ &= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X W_{1i}' w_{2i}', y\Big) + \frac{\lambda}{2} \sum_{z_i \in \tilde{\mathcal{Z}}} \|W_{1i}'\|_2^2 + |w_{2i}'|^2 \geq p^*, \end{aligned}$$

which completes the proof.

□

Proposition A.2. *Problem 3 is equivalent to following unconstrained relaxation of the ReLU training problem's convex reformulation (Problem 2):*

$$\begin{aligned} \min_{v,w} \frac{1}{2} \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X (v_i - w_i), y \right) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2 + \|w_i\|_2 \\ \text{s.t. } (2D_i - I_n) X v_i \succeq 0, (2D_i - I_n) X w_i \succeq 0, \end{aligned} \tag{15}$$

Proof. Suppose that (v^*, w^*) is an optimal solution (15). Defining $r^* = v^* - w^*$, it holds by the triangle inequality that

$$\|r_i^*\|_2 \leq \|v_i^*\|_2 + \|w_i^*\|_2 ,$$

for each $i \in [P]$. Since replacing $v_i^* - w_i^*$ with r_i^* does not change the model prediction

$$\hat{y} = \sum_{D_i \in \tilde{\mathcal{D}}} D_i X (v_i - w_i),$$

we have shown that $(r^*, 0)$ defines an equivalent model with the same or smaller objective value. Accordingly, any solution to (15) must have $v_i^* = 0$ or $w_i^* = 0$. Equivalence of the two problems follows immediately. □

A.1. Extension to Multi-class Classification

Now we extend our sub-sampled C-ReLU and C-GReLU formulations to vector-valued problems, such as occur in multi-class classification. Our starting place is the following vector-output variant of the NC-ReLU problem

$$\min_{W_1, W_2} \mathcal{L}\Big(\sum_{i=1}^m (X W_{1i})_+ W_{2i}^\top, Y\Big) + \frac{\lambda}{2} \sum_{i=1}^m \|W_{1i}\|_2^2 + \|W_{2i}\|_1^2 , \tag{16}$$

where now labels $Y \in \mathbb{R}^{n \times C}$. We note that the main difference between this formulation and Equation (1) is that each row of X now maps to a vector rather than a single scalar, and the use of ℓ_1-squared regularization on the second-layer weights. We now present a similar result to Lemma A.1 for this particular problem.

Lemma A.3. *The non-convex problem (16) is equivalent to the following program,*

$$\min_{\{W_1^k, w_2^k\}_{k=1}^C} \mathcal{L}\Big(\sum_{i=1}^m (XW_{1i})_+ W_{2i}^\top, Y\Big) + \lambda \sum_{i=1}^m \|W_{1i}\|_2 \quad \text{s.t. } \|W_{2i}\|_1 = 1 \; \forall i \in [m] \tag{17}$$

Proof. Follow from the proof of Lemma A.1 (Appendix A), i.e. apply Young's inequality to achieve

$$p^* = \min_{W_1, W_2} \mathcal{L}\Big(\sum_{i=1}^m (XW_{1i})_+ W_{2i}^\top, Y\Big) + \frac{\lambda}{2} \sum_{i=1}^m \|W_{1i}\|_2^2 + \|W_{2i}\|_1^2 = \min_{W_1, W_2} \mathcal{L}\Big(\sum_{i=1}^m (XW_{1i})_+ W_{2i}^\top, Y\Big) + \lambda \sum_{i=1}^m \|W_{1i}\|_2 \|W_{2i}\|_1$$

Then, this is clearly equivalent to

$$\min_{W_1, W_2} \mathcal{L}\Big(\sum_{i=1}^m (XW_{1i})_+ W_{2i}^\top, Y\Big) + \lambda \sum_{i=1}^m \|W_{1i}\|_2 \quad \textit{s.t. } \|W_{2i}\|_1 = 1 \; \forall i \in [m]$$

□

We can form the one-vs-all convex reformulation as follows:

$$\min_{\{v^k, w^k\}_{k=1}^C} \mathcal{L}\Big(\sum_{k=1}^C \sum_{D_i \in \tilde{\mathcal{D}}} D_i X (v_i^k - w_i^k) e_k^\top, Y\Big) + \lambda \sum_{k=1}^C \sum_{D_i \in \tilde{\mathcal{D}}} \left\|v_i^k\right\|_2 + \left\|w_i^k\right\|_2 \quad \text{s.t. } (2D_i - I_n) X v_i^k \succeq 0, \; (2D_i - I_n) X w_i^k \succeq 0, \tag{18}$$

where e_k is the kth standard basis vector.

Then, we have the following analog of Theorem 2.1 for the vector-output case:

Theorem A.4. *Suppose $(W_1^* W_2^*)$ and $\{(v^{k*}, w^{k*})\}_{k=1}^C$ are global minimizers of Problems 17 and Problem 18, respectively. If the number of hidden units satisfies*

$$m^* \geq b := \sum_{k=1}^C \sum_{D_i \in \tilde{\mathcal{D}}} \left| \{v_i^{k*} : v_i^{k*} \neq 0\} \cup \{w_i^{k*} : w_i^{k*} \neq 0\} \right|,$$

and the optimal activations are in the convex model,

$$\{\text{diag}\,(XW_{1i}^* > 0 : i \in [m])\} \subseteq \tilde{\mathcal{D}},$$

then the two problems have same the optimal value.

Proof. We follow from the proof of Theorem 2.1. Let p^* be the optimal value of (17) and d^* be the optimal value of (18). First, suppose $\{(v^{k*}, w^{k*})\}_{k=1}^C$ is a global minimizer of Problem 18. Then, let

$$\left(W_{1(i,k)}^*, W_{2(i,k)}^*\right) = \bigcup_{D_i \in \tilde{\mathcal{D}}} \bigcup_{k=1}^C \{(v_i^{k*}, e_k) : v_i^{k*} \neq 0\} \cup \{(w_i^{k*}, -e_k) : w_i^{k*} \neq 0)\}$$

where we set $W_{1(i,k)}^* = 0$ and $W_{2(i,k)}^* = 0$ for non-assigned neurons. It holds by assumption that $b \leq m$ and thus this is a valid input for (17). Further, we have, due to the constraints,

$$(XW_{1(i,k)}^*)_+ W_{2(i,k)}^{*\top} = \begin{cases} D_i X v_i^{k*} e_k^\top & \text{if } W_{1(i,k)}^* = v_i^{k*} \\ D_i X w_i^{k*} e_k^\top & \text{if } W_{1(i,k)}^* = w_i^{k*} \\ 0 & o.w. \end{cases}$$

Inserting to the objective for the convex program,

$$d^* = \mathcal{L}\Big(\sum_{k=1}^{C}\sum_{D_i\in\tilde{\mathcal{D}}} D_i X(v_i^{k*} - w_i^{k*})e_k^\top, Y\Big) + \lambda\sum_{k=1}^{C}\sum_{D_i\in\tilde{\mathcal{D}}} \left\|v_i^{k*}\right\|_2 + \left\|w_i^{k*}\right\|_2 \tag{19}$$

$$= \mathcal{L}\Big(\sum_{(i,k)}(XW_{1(i,k)}^*)_+ W_{2(i,k)}^{*\top}, Y\Big) + \lambda\sum_{(i,k)}\left\|W_{1(i,k)}^*\right\|_2 \tag{20}$$

$$\geq p^* \tag{21}$$

Now, we seek to find the other direction, i.e. show $p^* \geq d^*$ and show a mapping. Let (W_{1i}^*, W_{2i}^*) be a solution to (17). Defining, as in Theorem 2.1,

$$T(j) = \{i \in [m] : \text{diag}\,(XW_{1i}^* > 0) = D_j\}\,.$$

Recalling $\{\text{diag}\,(XW_{1i}^* > 0 : i \in [m])\} \subseteq \tilde{\mathcal{D}}$, by assumption, we define a valid candidate solution as

$$\left\{\left\{(v_j^{k*}, w_j^{k*})\right\}_{k\in[C]}\right\}_{D_j\in\tilde{\mathcal{D}}} = \left\{\left\{\sum_{i\in T(j)} W_{1i}^* w_{2i}^{k*}\mathbb{1}(W_{2i}^{k*}\geq 0)\right\}_{k\in[C]}, \left\{-\sum_{i\in T(j)} W_{1i}^* w_{2i}^{k*}\mathbb{1}(W_{2i}^{k*} < 0)\right\}_{k\in[C]}\right\}_{D_j\in\tilde{\mathcal{D}}}$$

Then, by the same co-linearity arguments as in Theorem 2.1, we have

$$\begin{aligned}
p^* &= \mathcal{L}\Big(\sum_{i=1}^{m}(XW_{1i}^*)_+ W_{2i}^{*\top}, Y\Big) + \lambda\sum_{i=1}^{m}\|W_{1i}^*\|_2 \\
&= \mathcal{L}\Big(\sum_{i=1}^{m}\sum_{k=1}^{C}(XW_{1i}^*)_+ W_{2i}^{k*}e_k^\top, Y\Big) + \lambda\sum_{i=1}^{m}\|W_{1i}^*\|_2\sum_{k=1}^{C}|W_{2i}^{k*}| \\
&= \mathcal{L}\Big(\sum_{i=1}^{m}\sum_{k=1}^{C}(XW_{1i}^*)_+ W_{2i}^{k*}e_k^\top, Y\Big) + \lambda\sum_{i=1}^{m}\sum_{k=1}^{C}\|W_{1i}^*\|_2\,|W_{2i}^{k*}| \\
&= \mathcal{L}\Big(\sum_{D_j\in\tilde{\mathcal{D}}}\sum_{k=1}^{C} D_j X(v_j^{k*} - w_j^{k*})e_k^\top, Y\Big) + \lambda\sum_{D_j\in\tilde{\mathcal{D}}}\sum_{k=1}^{C}\left\|v_j^{k*}\right\|_2 + \left\|w_j^{k*}\right\|_2 \\
&\geq d^*
\end{aligned}$$

□

Thus, the vector-output NC-ReLU training problem (16) is equivalent to the one-vs-all C-ReLU problem (18) if the conditions of Theorem A.4 are satisfied. Further, taking $\tilde{\mathcal{D}} = \mathcal{D}_X$ and applying Theorem A.4 yields

$$m^* = \sum_{k=1}^{C}\sum_{D_i\in\mathcal{D}_X}\left|\{v_i^{k*} : v_i^{k*}\neq 0\}\cup\{w_i^{k*} : w_i^{k*}\neq 0\}\right|.$$

It follows that global optimization of the vector-output NC-ReLU problem requires $m \geq m^*$ neurons, where $m^* \leq C(n+1)$.

The gated ReLU analogs to vector-output ReLU architectures can be formulated in the same fashion.

B. Equivalence of ReLU and Gated ReLU: Proofs

First we give a simple lemma that will be useful when characterizing the span of $\mathcal{K}_i - \mathcal{K}_i$.

Lemma B.1. *The cone $\mathcal{K}_i$ has a non-empty interior if and only if $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$.*

Proof. Let $x \in \text{aff}(\mathcal{K}_i)$. Then $x = \sum_{j=1}^m \alpha_j y_j$, $y_j \in \mathcal{K}_i$. Let $j \in [m]$. If $\alpha_j \geq 0$, then $\alpha_j y_j \in \mathcal{K}_i$ since $\mathcal{K}_i$ is a cone. Otherwise, $\alpha_j y_j \in -\mathcal{K}_i$. Either way, we have $\alpha_j y_j \in \mathcal{K}_i - \mathcal{K}_i$.

Observe that $\mathcal{K}_i - \mathcal{K}_i$ is a convex cone since $\mathcal{K}_i$ is a convex cone. Thus, $\alpha_1 y_1 + \alpha_2 y_2 \in \mathcal{K}_i - \mathcal{K}_i$. Induction on $j \in [m]$ now implies $x \in \mathcal{K}_i - \mathcal{K}_i$ and thus $\text{aff}(\mathcal{K}_i) \subseteq \mathcal{K}_i - \mathcal{K}_i$. Now suppose $x \in \mathcal{K}_i - \mathcal{K}_i$ so that $x = y_1 - y_2$, where $y_1, y_2 \in \mathcal{K}_i$. It is trivial to deduce $x \in \text{aff}(\mathcal{K}_i)$; we conclude that $\text{aff}(\mathcal{K}_i) = \mathcal{K}_i - \mathcal{K}_i$.

Since $0 \in \text{aff}(\mathcal{K}_i)$, this set is a linear subspace of $\mathbb{R}^d$. If $\mathcal{K}_i$ has an interior point, then $\text{aff}(\mathcal{K}_i) = \mathbb{R}^d$ and we must have $\text{aff}(\mathcal{K}_i) = \mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$. On the other hand, if $\mathcal{K}_i$ does not have an interior point, then $\text{aff}(\mathcal{K}_i) \subset \mathbb{R}^d$ and $\text{aff}(\mathcal{K}_i) = \mathcal{K}_i - \mathcal{K}_i \subset \mathbb{R}^d$ must hold; we have shown the reverse implication by the contrapositive. □

Now we show that $\mathcal{K}_i$ has an interior point when X is full row-rank. The proof proceeds by studying a relative interior point of $\mathcal{K}_i$.

Proposition 3.1. *If X is full row-rank, then $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ for every $D_i \in \mathcal{D}_\mathcal{X}$. As a result, the C-ReLU, C-GReLU, NC-ReLU, and NC-GReLU problems are all equivalent.*

Proof. Let $\bar{w} \in \text{relint}(\mathcal{K}_i)$, which exists since the relative interior of a non-empty convex set is non-empty (Bertsekas, 2009). Assume that the inequality,

$$(2D_i - I)X\bar{w} \succeq 0,$$

is tight for at least one index $j \in [n]$; let $X_\mathcal{I}$ be the submatrix of X formed by the rows of X for which the inequality is tight. Define $\tilde{D} = (2D_i - I)$. Since X is full row-rank, the rows of $X_\mathcal{I}$ are linearly independent. Let x_k be an arbitrary row of $X_\mathcal{I}$ (noting $x_k \neq 0$ by linear independence) and define z_k to be the component of x_k which is orthogonal to the remaining rows of $X_\mathcal{I}$. Clearly such a vector exists since the rows of $X_\mathcal{I}$ are linearly independent. Define $w' = \bar{w} + [\tilde{D}_i]_{kk} z_k$ to obtain

$$[\tilde{D}_i]_{kk} x_k^\top w' = [\tilde{D}_i]_{kk} x_k^\top \bar{w} + \|z_k\|_2^2 = \|z_k\|_2^2 > 0,$$

and, for $j \neq k$,

$$[\tilde{D}_i]_{jj} x_j^\top w' = [\tilde{D}_i]_{jj} x_j^\top \bar{w} + [\tilde{D}_i]_{jj} [D_i]_{kk} x_j^\top z_k \geq 0,$$

since x_j and z_k are orthogonal. This contradicts $\bar{w} \in \text{relint}(\mathcal{K}_i)$ and we deduce that $(2D_i - I)X\bar{w} \succ 0$. Lemma B.1 now implies that $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$.

Let u^* be an optimal solution to the C-GReLU problem with $\tilde{\mathcal{D}} = \mathcal{D}_X$. Since the Minkowski difference $\mathcal{K}_i - \mathcal{K}_i$ spans $\mathbb{R}^d$ for every $D_i \in \mathcal{D}_X$, we can find v_i, w_i such that $u_i^* = v_i - u_i$. Moreover, we can always reparameterize the optimal solution to the C-ReLU problem as $u_i = v_i^* - w_i^*$. A simple reduction argument now shows the two problems are equivalent. Applying theorems 2.1 and 2.2 extends the equivalence to NC-ReLU and NC-GReLU. □

The main difficulty extending Proposition 3.1 to general, full-rank X is showing that none of the cone-constraints are tight at $\bar{w}$. Unfortunately, the following shows that these difficulties cannot be resolved.

Proposition B.2. *There exists a full-rank data matrix X and activation pattern $D_i \in \mathcal{D}_X$ such that $\mathcal{K}_i$ is contained in a linear subspace of $\mathbb{R}^d$.*

Proof. Let $d = 3$, $n = 4$ and take

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

It is easy to see that X is full-rank, although it does not have full row-rank since x_1, x_2, x_3 are collinear. The cone $\mathcal{K}_i = \{w : Xw \succeq 0\}$, which corresponds to positive activations for each example, has the following alternative representation:

$$\mathcal{K}_i = \{\alpha * e_3 : \alpha \geq 0\}.$$

Clearly $\mathcal{K}_i$ is contained in a subspace of dimension one. Thus, we cannot hope for $\mathcal{K}_i$ to have full affine dimension in this more general setting. □

B.1. Singular Cones are Contained in Non-Singular Cones

Considering the counter-example in Proposition B.2, we find the "bad" $\mathcal{K}_i$ is contained within the subspace $\mathcal{S}$ spanned by e_3. By construction, every $w \in \mathcal{K}_i$ is orthogonal to x_1, x_2, and x_3, meaning these examples don't contribute to the constraints on $\mathcal{K}_i$ once it is restricted to $\mathcal{S}$. Intuitively, changing the activation associated with x_1, x_2, or x_3 can only lead to cones which contain $\mathcal{K}_i$. For example, consider $\mathcal{K}_i' = \{w : \langle x_3, w\rangle \leq 0\}, X_{-3}w \succeq 0$, which is equal to the non-negative orthant, $\mathbb{R}^3_+$. We immediately observe $\mathcal{K}_i \subset \mathcal{K}_i'$ and we may replace the degenerate cone with the alternative, full-dimensional $\mathcal{K}_2$. The rest of this section formalizes these observations.

Definition B.3. Let $\tilde{X} \in \mathbb{R}^{m\times d}$ and consider a cone $\mathcal{K} = \left\{w : \tilde{X}w \succeq 0\right\}$ such that $\text{aff}(\mathcal{K}) = \mathcal{S} \subset \mathbb{R}^d$. We call an index set $\mathcal{I} \subseteq [m]$ minimal for S if

$$\mathcal{K}_{\mathcal{I}} = \left\{w : \tilde{X}_{\mathcal{I}}w \succeq 0\right\} \subseteq \mathcal{S}$$

and, for any $j \in \mathcal{I}$,

$$\mathcal{K}_{\mathcal{I}\setminus j} \not\subseteq \mathcal{S}.$$

That is, removing any half-space constraint indexed by $\mathcal{I}$ ensures $\mathcal{K}_{\mathcal{I}}$ is not contained in $\mathcal{S}$.

Note that there may be many minimal index sets for a singular cone and these sets may have varying cardinalities. However, each minimal index set shares a key property: every row $\tilde{x}_i$ indexed by such $\mathcal{I}$ must be orthogonal to $\mathcal{S}$.

Lemma B.4. *Let $\tilde{X} \in \mathbb{R}^{m\times d}$ such that the cone $\mathcal{K} = \left\{w : \tilde{X}w \succeq 0\right\}$ is singular. Let $\mathcal{S} = \text{aff}(\mathcal{K})$ be the smallest containing subspace and $\mathcal{I}$ a minimal index set for $\mathcal{S}$. Then, $\langle \tilde{x}_i, s\rangle = 0$ for all $i \in \mathcal{I}$ and $s \in \mathcal{S}$.*

Proof. Suppose $\langle \tilde{x}_i, w\rangle \neq 0$ for some $i \in \mathcal{I}$ and $w \in \mathcal{K}$. Since $w \in \mathcal{K}$, $\tilde{X}w \succeq 0$ and it must be that $\langle \tilde{x}_i, w\rangle > 0$. Let $z \in \mathcal{S}^\perp$ be arbitrary and define $w' = z + \alpha w$, $\alpha > 0$. By taking α to be sufficiently large, we obtain

$$\langle \tilde{x}_i, w'\rangle = \langle \tilde{x}_i, z\rangle + \alpha \langle \tilde{x}_i, w\rangle > 0,$$

Since $w' \notin \mathcal{S} \supseteq \mathcal{K}$, we must have

$$\tilde{X}_{\mathcal{I}\setminus i}w' \not\succeq 0 \implies \tilde{X}_{\mathcal{I}\setminus i}z \not\succeq 0,$$

where we have used $Xw \succeq 0$. Moreover, this holds for all $z \in \mathcal{S}^\perp$, which implies that $\mathcal{K}_{\mathcal{I}\setminus i} \subseteq \mathcal{S}$ and $\mathcal{I}$ cannot be minimal for $\mathcal{S}$. We conclude $\langle \tilde{x}_i, w\rangle = 0$ for all $i \in \mathcal{I}$ and $w \in \mathcal{K}$ by contradiction.

Since $\mathcal{S}$ is the affine hull of $\mathcal{K}$, we have for every $s \in \mathcal{S}$ and $i \in \mathcal{I}$ the following:

$$\langle \tilde{x}_i, s\rangle = \left\langle \tilde{x}_i, \sum_{j=1}^k \alpha_j y_j \right\rangle = \sum_{j=1}^k \alpha_j \langle \tilde{x}_i, y_j\rangle = 0,$$

since $y_j \in \mathcal{K}$. □

Similarly, if any constraint is tight at a relative interior point, then that constraint must be orthogonal to the cone.

Lemma B.5. *Let $\tilde{X} \in \mathbb{R}^{m\times d}$, $\mathcal{K} = \left\{w : \tilde{X}w \succeq 0\right\}$, and $\bar{w}$ be a relative interior point of $\mathcal{K}$. If $\langle \tilde{x}_j, \bar{w}\rangle = 0$ for any $j \in [m]$, then $\tilde{x}_j$ is orthogonal to $\mathcal{K}$.*

Proof. Suppose $\langle \tilde{x}_j, \bar{w}\rangle = 0$ for some $j \in [m]$. If there exists $w \in \mathcal{K}$ such that $\langle \tilde{x}_j, \bar{w}\rangle > 0$, then $\langle \tilde{x}_j, \bar{w} + w\rangle > 0$ and $\bar{w} + w \in \mathcal{K}$, which contradicts the assumption $\bar{w}$ is a relative interior point. Since every $w \in \mathcal{K}$ satisfies $\langle \tilde{x}_j, w\rangle \geq 0$, we conclude $\langle \tilde{x}_j, w\rangle = 0$ for all such w. □

Lemma B.4 is key to our analysis because it implies that the half-space constraints which force $\mathcal{K}$ to lie in a subspace don't "cut into" that subspace. In particular, it means that we can choose to enforce membership in $\mathcal{H}_{x_i}$ or $\mathcal{H}_{-x_i}$ without changing the inclusion. We show now that there exists a choice of signed half-spaces for which the intersection is non-singular.

Lemma B.6. *Let $x_1, \dots x_m$ be a collection of vectors in $\mathbb{R}^d$ and $X \in \mathbb{R}^{m \times d}$ the matrix formed by stacking these vectors. Then there exists a diagonal matrix $\tilde{D}$, where $\tilde{D}_{jj} \in \{-1, 1\}$, such that*

$$\mathit{aff}(\left\{w : \tilde{D}Xw \succeq 0\right\}) = \mathbb{R}^d.$$

Proof. We proceed by induction. Let $\tilde{\mathcal{D}}_{11} = 1$ and $\mathcal{K}_1 = \mathcal{H}_{x_1} := \{w : \langle x_1, w\rangle \geq 0\}$. Clearly $\text{aff}(\mathcal{K}_1) = \mathbb{R}^d$ since it is a half-space.

Now, let $t < m$ and assume that $\text{aff}(\mathcal{K}_t) = \mathbb{R}^d$. Consider

$$A_{t+1} := \mathcal{K}_t \cap \mathcal{H}_{x_{t+1}}.$$

If $\mathcal{S} := \text{aff}(A_{t+1}) = \mathbb{R}^d$, then the inductive hypothesis holds at $\mathcal{K}_{t+1} = A_{t+1}$ and we can choose $\tilde{D}_{t+1,t+1} = 1$. Otherwise, x_{t+1} must be part of a minimal index set $\mathcal{I} \subseteq [t+1]$ such that $\left\{w : \tilde{D}_{\mathcal{I}} X_{\mathcal{I}} w \succeq 0\right\} \subseteq \mathcal{S}$. Lemma B.4 now implies that x_{t+1} is orthogonal to $\mathcal{S}$. Let $w \in \mathcal{K}_t \cap \mathcal{S}^\perp$ (which is non-empty by the inductive hypothesis) and observe that

$$\langle w, x_{t+1}\rangle < 0,$$

must hold, otherwise $w \in A_{t+1}$. We deduce $\langle w, x_{t+1}\rangle \leq 0$ for every $w \in \mathcal{K}_t$ and thus

$$\mathcal{K}_t \cap \mathcal{H}_{-x_{t+1}} = \mathcal{K}_t,$$

which is full-dimensional by the inductive hypothesis. Taking $\mathcal{K}_{t+1} = \mathcal{K}_t$ and $\mathcal{D}_{t+1,t+1} = -1$ completes the case.

The desired result follows by induction.

□

We now use Lemma B.6 to show that every singular cone is contained in a non-singular cone.

Proposition 3.2. *Suppose $\mathcal{K}_i$ is singular for $D_i \in \mathcal{D}_X$. Then $\exists D_j \in \mathcal{D}_X$ such that $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$ and $\mathcal{K}_i \subset \mathcal{K}_j$.*

Proof. For simplicity, we drop the index i and work with $\mathcal{K} = \{w : (2D - I)Xw \succeq 0\}$. To ease the notation, we also write $\tilde{D} = (2D - I)$ and $\tilde{X} = \tilde{D}X$. Let $\mathcal{S} = \text{aff}(\mathcal{K})$ be the smallest subspace containing $\mathcal{K}$, $\mathcal{O}$ be the set of all indices j such that x_j is orthogonal to $\mathcal{S}$, and $\mathcal{U} = [n] \setminus \mathcal{O}$. Lemma B.6 implies that there exists an alternative activation pattern $\tilde{D}'$ such that, $\tilde{D}'_{\mathcal{U}} = \tilde{D}_{\mathcal{U}}$, and if $\mathcal{K}' = \left\{w : \tilde{D}'Xw \succeq 0\right\}$, then

$$\mathcal{K}'_{\mathcal{O}} := \left\{w : \tilde{D}'_{\mathcal{O}} X_{\mathcal{O}} w \succeq 0\right\} \quad \text{satisfies} \quad \text{aff}\left(\mathcal{K}'_{\mathcal{O}}\right) = \mathbb{R}^d.$$

Since the vectors indexed by $\mathcal{O}$ are orthogonal to $\mathcal{S}$, they are also orthogonal to every $w \in \mathcal{K}$, implying $\mathcal{K} \subset \mathcal{K}'$. In other words, the change of activation signs preserves inclusion of $\mathcal{K}$.

Let us show that $\mathcal{K}'$ contains an interior point. Let $\bar{w}$ be a relative interior point of $\mathcal{K}$ and suppose that $\langle \tilde{x}_j, \bar{w}\rangle = 0$ for some $j \in \mathcal{U}$. Lemma B.5 implies $\tilde{x}_j$ is orthogonal to $\mathcal{K}$. But, then $j \in \mathcal{O}$, which is a contradiction. We conclude $\tilde{X}_{\mathcal{U}} \bar{w} \succ 0$.

Let $\bar{y}$ be an interior point of $\mathcal{K}'_{\mathcal{O}}$, which exists because $\text{aff}(\mathcal{K}'_{\mathcal{O}}) = \mathbb{R}^d$. The point $\bar{z} = \bar{y} + \alpha\bar{w}$, $\alpha > 0$ satisfies

$$\tilde{D}'_{\mathcal{O}} X_{\mathcal{O}} \bar{z} = \tilde{D}'_{\mathcal{O}} X_{\mathcal{O}} \bar{y} \succ 0,$$

since $\bar{w}$ is orthogonal to the rows of $X_{\mathcal{O}}$. Similarly, by taking α to be sufficiently large, we have

$$\tilde{D}'_{\mathcal{U}} X_{\mathcal{U}} \bar{z} \succ 0.$$

We have shown that $\bar{z}$ is an interior point of $\mathcal{K}'$ and Lemma B.1 now implies $\mathcal{K}' - \mathcal{K}' = \mathbb{R}^d$. □

Theorem 3.3. *Let $\lambda = 0$. For every training set (X, y), there exists $\tilde{\mathcal{D}} \subseteq \mathcal{D}_X$ such that the sub-sampled C-GReLU and C-ReLU problems are both equivalent to the full C-ReLU problem with $\tilde{\mathcal{D}} = \mathcal{D}_X$.*

Proof. Let u^* be a solution to the full C-GReLU problem and p^* the optimal value. For every $D_i \in \mathcal{D}_X$, we either have $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$, or not. If the former condition holds, take $r_i^* = u_i^*$. Otherwise, invoke Proposition 3.2 to obtain $D_j \in \mathcal{D}_X$ such that $\mathcal{K}_i \subset \mathcal{K}_j$ and $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$. By the construction in the proof of Proposition 3.2, if $[D_j]_{kk} \neq [D_i]_{kk}$, then x_k is orthogonal to $\mathcal{K}_i$ and we deduce

$$D_j X u_i^* = D_i X u_i^*.$$

We may therefore merge the two neurons as $r_j^* = u_j^* + u_i^*$ and update $\tilde{\mathcal{D}}' = \tilde{\mathcal{D}} \setminus D_i$ without changing the (optimal) value of the C-GReLU program. In this way, we obtain a sub-sampled C-GReLU problem with activation patterns $\tilde{\mathcal{D}}$, for which

$$\begin{aligned} p^* &= \mathcal{L}\left(\sum_{D_i \in \mathcal{D}_X} D_i X u_i^*, y\right) \\ &= \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X r_i^*, y\right). \end{aligned}$$

Since $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$ for every $D_i \in \tilde{\mathcal{D}}$, we may decompose $r_i^* = v_i' - w_i'$, such that $v_i', w_i' \in \mathcal{K}_i$. In this way, we obtain a feasible input (v', w') to C-ReLU such that

$$\begin{aligned} p^* &= \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X r_i^*, y\right) \\ &= \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i' - w_i', y\right) \\ &\geq \min_{v,w}\left[\mathcal{L}\left(\sum_{D_i \in \mathcal{D}_X} D_i X\left(v_i - w_i\right), y\right) \;:\; v_i, w_i \in \mathcal{K}_i\right] \\ &:= d^*, \end{aligned}$$

To establish the reverse inequality, let (v^*, w^*) be a solution to the full C-ReLU problem and repeat the neuron-merging process described above to obtain a sub-sampled C-ReLU problem with identical objective value and activation patterns $\tilde{\mathcal{D}}$. For every $D_i \in \tilde{\mathcal{D}}$, let $u_i' = r_i^* - s_i^*$ (where r_i^* and s_i^* are the weights obtained from after merging neurons) to obtain

$$\begin{aligned} p^* &= \mathcal{L}\left(\sum_{D_i \in \mathcal{D}_X} D_i X u_i', y\right) \\ &\geq \min_u \mathcal{L}\left(\sum_{D_i \in \mathcal{D}_X} D_i X u_i', y\right) \\ &= \min_u \mathcal{L}\left(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i', y\right) := d^*. \end{aligned}$$

We conclude that the two sub-sampled optimization problems are equivalent and achieve the same global minimum as the full C-ReLU problem.

□

B.2. Approximating ReLU by Cone Decomposition: Proofs

Proposition 3.4. *Suppose X is full row-rank. If $\mathcal{I} = \{i \in [n] : \langle \tilde{x}_i, u\rangle < 0\}$, then for every $u \in \mathbb{R}^d$,*

$$u = (u + w) - w, \textit{where } w = -\tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} u,$$

is a valid decomposition onto $\mathcal{K} - \mathcal{K}$ satisfying,

$$\|u + w\|_2 + \|w\|_2 \leq 2\|u\|_2.$$

Proof. Let's show that the decomposition is valid. Setting $v = u + w$, we obtain $v - w = u$ by construction. For notational ease, let $\mathcal{J} = [n] \setminus \mathcal{I}$. It holds that

$$\tilde{X}_{\mathcal{I}} w = -\tilde{X}_{\mathcal{I}} \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} u = -\tilde{X}_{\mathcal{I}} u > 0.$$

Moreover, since X is full row-rank and $\mathcal{J} \cap \mathcal{I} = \emptyset$, we have

$$\tilde{X}\tilde{X}^{\dagger} = \tilde{X}\tilde{X}^{\top}\left(\tilde{X}\tilde{X}^{\top}\right)^{-1} = I,$$

which implies that $\tilde{X}_{\mathcal{J}} \tilde{X}_{\mathcal{I}}^{\dagger} = 0$. We deduce

$$\tilde{X}_{\mathcal{J}} w = -\tilde{X}_{\mathcal{J}} \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} u = 0.$$

Moreover, we also have

$$\tilde{X}_{\mathcal{I}} v = \tilde{X}_{\mathcal{I}} u - \tilde{X}_{\mathcal{I}} u = 0,$$

and

$$\tilde{X}_{\mathcal{J}} v = \tilde{X}_{\mathcal{J}} u + 0 \geq 0,$$

by definition of $\mathcal{J}$. We conclude $w, v \in \mathcal{K}$.

To show the approximation result, start from

$$\begin{aligned} \|w\|_2 &= \left\| \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} u \right\|_2 \\ &\leq \left\| \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} \right\|_2 \|u\|_2 \\ &= \|u\|_2 . \end{aligned}$$

Triangle inequality now gives,

$$\begin{aligned} \|v\|_2 &= \left\| u - \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}} u \right\|_2 \\ &\leq \left\| (I - \tilde{X}_{\mathcal{I}}^{\dagger} \tilde{X}_{\mathcal{I}}) \right\|_2 \|u\|_2 \\ &= \|u\|_2 , \end{aligned}$$

and summing these two inequalities gives the result. □

Proposition 3.5. *There does not exist a decomposition $u = v - w$, where $v, w \in \mathcal{K}$, such that*

$$\|v\|_2 + \|w\|_2 \leq C \|u\| ,$$

holds for an absolute constant C.

Proof. Consider the data matrix

$$X = \begin{bmatrix} 1 & \alpha \\ -1 & \alpha \end{bmatrix} \tag{22}$$

The cone corresponding to positive activations for both examples is $\mathcal{K} = \left\{ x \in \mathbb{R}^d : x_2 \geq -x_1/\alpha, x_2 \geq x_1/\alpha \right\}$. Consider decomposing the vector $u = [2, 0]$ onto $\mathcal{K} - \mathcal{K}$. Clearly $u \notin \mathcal{K}$; by inspection, we see that the minimum norm decomposition is given by $v = [1, 1/\alpha]$, and $w = [-1, 1/\alpha]$. Taking $\alpha \to 0$, we find $\|v\|_2 = \|w\|_2 \to \infty$. □

Proposition 3.6. *For every $u \in \mathbb{R}^d$, if $(\bar{v}, \bar{w})$ is a solution to the cone-decomposition program (5) with $R(v, w) = \|v\|_2 + \|w\|_2$, then there exists $\mathcal{J} \subseteq [n]$ such that*

$$\|\bar{v}\|_2 + \|\bar{w}\|_2 \leq \left(1 + 2\kappa(\tilde{X}_{\mathcal{J}})\right) \|u\|_2 .$$

Proof. First, we re-parameterize the problem: $u = v - w$ implies $u + w = v$, giving the equivalent program

$$\min_{w \in \mathcal{K}} \|w\|_2 + \|u + w\|_2^2 . \tag{23}$$

In order to character the solution, we re-write the constraints into a single system of linear inequalities as follows:

$$\begin{aligned}\mathcal{F} &= \left\{ w : \tilde{X}w \succeq 0, \tilde{X}w \succeq -\tilde{X}u \right\} \\ &= \left\{ w : \tilde{X}w \succeq 0, \tilde{X}w \succeq b \right\},\end{aligned}$$

where we have introduced $b = -\tilde{X}u$. It is possible to combine these inequalities by taking the element-wise maximum as follows:

$$\begin{aligned}\mathcal{F} &= \left\{ w : \tilde{X}w \succeq \max(0, b) \right\} \\ &= \left\{ w : \tilde{X}w \succeq (b)_+ \right\}.\end{aligned}$$

Let $\bar{w} \in \mathcal{F}$ be a optimal point for the reparameterized program. Relaxing the objective using triangle inequality gives,

$$\begin{aligned}\|\bar{w}\|_2 + \|u + \bar{w}\|_2 &= \min_{w \in \mathcal{F}} \|w\|_2 + \|u + w\|_2 \\ &\leq \min_{w \in \mathcal{F}} 2\|w\|_2 + \|u\|_2 .\end{aligned}$$

Let w' be a solution to the relaxation. The KKT conditions imply there exists a submatrix $\tilde{X}_{\mathcal{J}}$ for which the inequality constraints are tight:

$$\tilde{X}_{\mathcal{J}} w' = (b_{\mathcal{J}})_+ .$$

The set of vectors satisfying this equality is $\left\{ \left[\tilde{X}_{\mathcal{J}}\right]^\dagger (b_{\mathcal{J}})_+ + z : z \in \text{null}(X_{\mathcal{J}}) \right\}$. Choosing $z \neq 0$ can only increase the value of the objective, from which we deduce $w' = \left[\tilde{X}_{\mathcal{J}}\right]^\dagger (b_{\mathcal{J}})_+$. We obtain

$$\begin{aligned}\|\bar{w}\|_2 + \|\bar{v}\|_2 &\leq 2\|w'\|_2 + \|u\|_2 \\ &= 2\left\| \left[\tilde{X}_{\mathcal{J}}\right]^\dagger (b_{\mathcal{J}})_+ \right\| + \|u\|_2 \\ &\leq \frac{2}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})} \left\|(b_{\mathcal{J}})_+\right\|_2 + \|u\|_2 \\ &\leq \frac{2}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})} \|b_{\mathcal{J}}\|_2 + \|u\|_2 \\ &= \frac{2}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})} \left\|\tilde{X}_{\mathcal{J}} u\right\|_2 + \|u\|_2 \\ &\leq \left(1 + 2\frac{\sigma_{\max}(\tilde{X}_{\mathcal{J}})}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})}\right) \|u\|_2 .\end{aligned}$$

□

Theorem 3.7. *Let $\lambda \geq 0$ and let p^* be the optimal value of the full C-ReLU problem with training set (X, y). There exists $\mathcal{J} \subseteq [n]$ and sub-sampled C-GReLU problem with minimizer u^* and optimal value d^* satisfying,*

$$d^* \leq p^* \leq d^* + 2\lambda\kappa(\tilde{X}_{\mathcal{J}}) \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2 .$$

Proof. The proof is straightforward given our existing results. Let u^* be the solution to the full (potentially regularized) C-GReLU problem. For every $D_i \in \mathcal{D}_X$, we either have $\mathcal{K}_i - \mathcal{K}_i = \mathbb{R}^d$, or not. If the former condition holds, take $r_i^* = u_i^*$.

Otherwise, invoke Proposition 3.2 to obtain $D_j \in \mathcal{D}_X$ such that $\mathcal{K}_i \subset \mathcal{K}_j$ and $\mathcal{K}_j - \mathcal{K}_j = \mathbb{R}^d$. By the construction in the proof of Proposition 3.2, if $[D_j]_{kk} \neq [D_i]_{kk}$, then x_k is orthogonal to $\mathcal{K}_i$ and we deduce

$$D_j X u_i^* = D_i X u_i^*.$$

We may therefore merge the two neurons as $r_j^* = u_j^* + u_i^*$ and update $\tilde{\mathcal{D}}' = \tilde{\mathcal{D}} \setminus D_i$ without changing the loss component of the C-GReLU program. Furthermore, since $\left\|r_j^*\right\| \leq \left\|u_j^*\right\|_2 + \|u_i^*\|_2$, merging these neurons can only decrease the regularization term. [3] In this way, we obtain a sub-sampled C-GReLU problem with activation patterns $\tilde{\mathcal{D}}$ and optimal value d^*.

Let (v^*, u^*) be the optimal solution to full C-ReLU problem. Applying Proposition 3.6 for each $D_i \in \tilde{\mathcal{D}}$ gives decompositions $u_i^* = v_i' - u_i'$ such that

$$\begin{aligned}
d^* &= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i^*, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2 \\
&\leq \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i^* - w_i^*, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i^* - w_i^*\|_2 \\
&\leq \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i^* - w_i^*, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i^*\|_2 + \|w_i^*\|_2 \\
&= p^* \\
&\leq \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i' - w_i', y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i'\|_2 + \|w_i'\|_2 \\
&= \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i^*, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i'\|_2 + \|w_i'\|_2 \\
&\leq \mathcal{L}\Big(\sum_{D_i \in \tilde{\mathcal{D}}} D_i X u_i^*, y\Big) + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} 2\frac{\sigma_{\max}(\tilde{X}_{\mathcal{J}})}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})} \|u_i^*\|_2 \\
&= p^* + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} 2\frac{\sigma_{\max}(\tilde{X}_{\mathcal{J}})}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})} \|u_i^*\|_2 .
\end{aligned}$$

We have abused notation here and omitted the dependence on i in $\tilde{X}_{\mathcal{J}} = (2D_i - I)\, X_{\mathcal{J}_i}$. However, observe that $2D_i - I$ is orthonormal so that $\sigma_{\max}(\tilde{X}_{\mathcal{I}}) = \sigma_{\max}(X_{\mathcal{I}})$ and $\sigma_{\min}(\tilde{X}_{\mathcal{I}}) = \sigma_{\min}(X_{\mathcal{I}})$ for all $\mathcal{I}$. Maximizing over $\mathcal{I} \subseteq [n]$ now gives a fixed subset $\mathcal{J}$ for which the claimed bound holds. □

[3] In fact, we know that one of u_j^*, u_i^* is zero or they are collinear

C. Efficient Global Optimization: Proofs

Theorem 4.1. *Let (W_1^*, w_2^*) be the minimum-norm global minimizer of the NC-GReLU problem with gates $\mathcal{G}$. Then, we can compute an ϵ-optimal point $(W_{1\epsilon}, w_{2\epsilon})$ in iterations*

$$T \leq \left(2\epsilon^{-1}\lambda_{max}\left(M^\top M\right)\sum_{D_i\in\tilde{\mathcal{D}}}\|W_{1i}^* w_{2i}^*\|_2^2\right)^{1/2}.$$

Proof. Applying the solution mapping from the proof of Theorem 2.2 (see Appendix A) we find that taking

$$v_i' = W_{1i} * w_{2i},$$

for each $D_i \in \mathcal{D}$ yields a global minimizer of C-ReLU. Now we apply the iteration complexity of FISTA (Beck & Teboulle, 2009, Theorem 4.4) to obtain an ϵ-optimal solution in

$$\begin{aligned} T &\leq \frac{\left(2\lambda_{\max}\left(M^\top M\right)\sum_{D_i\in\tilde{\mathcal{D}}}\|v_i'\|_2^2\right)^{1/2}}{\epsilon^{1/2}} \\ &= \frac{\left(2\lambda_{\max}\left(M^\top M\right)\sum_{D_i\in\tilde{\mathcal{D}}}\|W_{1i}^* w_{2i}^*\|_2^2\right)^{1/2}}{\epsilon^{1/2}}, \end{aligned}$$

iterations. Note that we have used the fact that $\lambda_{\max}(M^\top M)$ is the Lipschitz smoothness constant of the squared-error loss.

□

Proposition 4.2. *Suppose $\tilde{w}$ is a minimizer of (9) and let $\tilde{v} = u + \tilde{w}$. If X is full row-rank, then*

$$\|(\tilde{X}\tilde{w})_-\|_2 + \|(\tilde{X}\tilde{v})_-\|_2 \leq \frac{2\rho}{\sigma_{min}(\tilde{X})}.$$

Furthermore, if $\rho > 0$, then the norm bound in Proposition 3.6 also holds for the approximate solution $(\tilde{v}, \tilde{w})$.

Alternatively, suppose X is not full row-rank. As $\rho_k \to 0$, every convergent subsequence of $(\tilde{v}_k, \tilde{w}_k)$ is a feasible cone decomposition. Moreover, at least one such sequence exists.

Proof. First-order optimality conditions for $\tilde{w}$ imply

$$-\tilde{X}^\top\left(b - \tilde{X}\tilde{w}\right)_+ \in \rho\cdot\partial\left\|\tilde{w}\right\|_2.$$

Noting that every vector in $\partial\left\|\tilde{w}\right\|_2$ has norm at most 1, we deduce

$$\begin{aligned} \left\|\tilde{X}^\top\left(b - \tilde{X}\tilde{w}\right)_+\right\|_2 &\leq \rho \\ \implies \left\|\left(b - \tilde{X}\tilde{w}\right)_+\right\|_2 &\leq \frac{\rho}{\sigma_{\min}(\tilde{X})} \\ \iff \left\|\left(\max\left\{-\tilde{X}u, 0\right\} - \tilde{X}\tilde{w}\right)_+\right\|_2 &\leq \frac{\rho}{\sigma_{\min}(\tilde{X})}. \end{aligned}$$

since $\tilde{X}$ is full row-rank and by definition of b. Using the fact that only positive elements contribute to the norm, we obtain the following two inequalities:

$$\begin{aligned} \left\|\left(\tilde{X}\tilde{w}\right)_-\right\|_2 &\leq \frac{\rho}{\sigma_{\min}(\tilde{X})} \\ \left\|\left(\tilde{X}(u+\tilde{w})\right)_-\right\|_2 &\leq \frac{\rho}{\sigma_{\min}(\tilde{X})}, \end{aligned}$$

Recalling $\tilde{v} = u + \tilde{w}$ and summing gives the first result.

If $\rho > 0$, then it is easy to observe (i.e. by arguing via contradiction) that $\tilde{w}$ must have smaller norm than any w' in a feasible decomposition (v', w'). Thus, it must also have smaller norm than $\bar{w}$ from the SOCP cone decomposition:

$$\|\tilde{w}\|_2 \leq \|\bar{w}\|_2 .$$

Since the proof of Proposition 3.6 relies on controlling only the norm of $\bar{w}$, the conclusion of that theorem also applies to $(\tilde{v}, \tilde{w})$.

Finally, suppose X is not full row-rank. For $\rho > 0$, Equation (9) is equivalent to solving

$$\textbf{CD-A}: \min_w g(w, \rho) := \frac{1}{2}\|(b - \tilde{X}w)_+\|_2^2 + \rho P(w) \quad \text{s.t.} \|w\|_2 \leq \|\bar{w}\|_2 , \tag{24}$$

where $\bar{w}$ is the norm of the minimum-norm (with respect to w only) solution to the cone decomposition problem. This is a minimization problem with compact constraint set; since g is continuous in both w and ρ, we may apply Berge's maximum theorem (Ausubel & Deneckere, 1993) to obtain find

$$g^*(\rho) = \min_w \{g(w, \rho) : \|w\|_2 \leq \|\bar{w}\|_2\},$$

is continuous. Since the cone decomposition is realizable at $\rho = 0$, $g^*(0) = 0$ and any sequence ρ_k converging to 0 satisfies $\lim_k g^*(\lambda_k) = 0$.

Let $\tilde{w}_k$ be the sequence of minimizers associated with ρ_k. Since $\tilde{w}_k$ is bounded, it has at least one convergent subsequence. Let $\tilde{w}_0$ be the associated limit point. Since g is continuous in w and ρ, we find

$$g(w_0, 0) = \lim_k g(\tilde{w}_k, \rho_k) = \lim_k g^*(\rho_k) = 0,$$

which shows that $\tilde{w}_0$ is a feasible decomposition. This completes the proof. □

Proposition C.1. *Suppose $\tilde{w}$ is a minimizer of (9) and let $\tilde{v} = u + \tilde{w}$. There exists $\mathcal{J} \subseteq [n]$ such that*

$$\|(\tilde{X}\tilde{w})_-\|_2 + \|(\tilde{X}\tilde{v})_-\|_2 \leq \frac{2\rho}{\sigma_{min}(\tilde{X}_{\mathcal{J}})},$$

where $\sigma_{min}(\tilde{X}_{calJ})$ is the minimum (possibly zero) singular value of the sub-matrix $\tilde{X}_{\mathcal{J}}$.

Proof. First-order optimality conditions for $\tilde{w}$ imply

$$-\tilde{X}^\top \left(b - \tilde{X}\tilde{w}\right)_+ \in \rho \cdot \partial \|\tilde{w}\|_2 .$$

Let $\mathcal{J} = \{i \in [n] : b_i - \langle \tilde{x}_i, \tilde{w} \rangle > 0\}$ and define B be a diagonal matrix such that $B_{jj} = 1$ if $j \in \mathcal{J}$ and 0 otherwise. In this notation, the optimality conditions can be written as

$$\begin{aligned} & -\tilde{X}^\top B \left(b - \tilde{X}\tilde{w}\right) \in \rho \cdot \partial \|\tilde{w}\|_2 \\ \iff & -(B\tilde{X}^\top) \left(b - \tilde{X}\tilde{w}\right) \in \rho \cdot \partial \|\tilde{w}\|_2 . \end{aligned}$$

Noting that every vector in $\partial \|\tilde{w}\|_2$ has norm at most 1, we deduce

$$\begin{aligned} & \left\|(B\tilde{X})^\top \left(b - \tilde{X}\tilde{w}\right)\right\|_2 \leq \rho \\ \iff & \left\|\tilde{X}_{\mathcal{J}}^\top \left(b - \tilde{X}\tilde{w}\right)_{\mathcal{J}}\right\|_2 \leq \rho \\ \implies & \left\|\left(b - \tilde{X}\tilde{w}\right)_{\mathcal{J}}\right\|_2 \leq \frac{\rho}{\sigma_{\min}(\tilde{X}_{\mathcal{J}})}. \end{aligned}$$

Recalling the definition of $\mathcal{J}$ and proceeding as in the proof of Proposition 4.2 gives the desired result. □

Remark C.2. The bound given in Proposition C.1 may be vacuous when $\tilde{X}_{\mathcal{J}}$ is not full row-rank since it concerns the minimum singular value, rather than the minimum *non-zero* singular value. In this respect, it is unlike the other results given so which have relied on the minimum non-zero singular value. However, it is worth reporting since this bound may be non-vacuous even when $\tilde{X}$ is not full row-rank and only the asymptotic result in Proposition 4.2 applies.

Theorem C.3. *Let (γ^*, ζ^*) be the minimum-norm maximizer of the Lagrange dual of Problem 2. Assume $\delta > 0$ is fixed and at each iteration Equation (11) is carried out exactly. Then, the AL method computes an ϵ-optimal estimate $(\gamma_\epsilon, \zeta_\epsilon)$ in*

$$T \leq \frac{\|\gamma^*\|_2^2 + \|\zeta^*\|_2^2}{\delta \epsilon},$$

Proof. Let d be the Lagrange dual function associated the Problem 2. We will show that the desired iteration complexity follows from standard results in the optimization literature.

Firstly, it is well-known that if the primal objective is a proper, closed, convex function, then one iteration of the AL method with penalty strength $\delta > 0$ is equivalent to the following proximal-point step on the dual problem:

$$(\gamma_{k+1}, \zeta_{k+1}) = \underset{\gamma \geq 0, \zeta \geq 0}{\arg\max} \left\{ d(\gamma, \zeta) - \frac{1}{2\delta} \left[\|\gamma - \gamma_k\|^2 \|\zeta - \zeta_k\|^2 \right] \right\}.$$

See Bertsekas (1997, Section 5.4.6) for a proof of this fact.

Invoking Güler (1991, Theorem 2.1) implies that the AL method attains the following convergence rate for the dual parameters:

$$d(\gamma^*, \zeta^*) - d(\gamma_k, \zeta_k) \leq \frac{\|\gamma^* - \gamma_0\|_2^2 + \|\zeta^* - \gamma_0\|_2^2}{\delta k}. \tag{25}$$

Choosing $\gamma_0 = \zeta_0 = 0$ and re-arranging this equation gives the desired iteration complexity. □

C.1. Data Normalization

Recall that the proximal gradient update has the form,

$$\begin{aligned} u_{k+1} &= \underset{x}{\arg\min} \left\{ f(u_k) + \langle \nabla f(u_k), x - u_k \rangle + \frac{1}{2\eta_k} \|x - u_k\|_2^2 + g(x) \right\} \\ &= \underset{x}{\arg\min} \left\{ \frac{1}{2\eta_k} \|x - (u_k - \eta_k \nabla f(u_k))\|_2^2 + g(x) \right\}. \end{aligned}$$

Taking $g(x)$ to be the group ℓ_1 penalty, we have

$$u_{k+1} = \underset{x}{\arg\min} \left\{ \frac{1}{2\eta_k} \|x - (u_k - \eta_k \nabla f(u_k))\|_2^2 + \lambda \sum_{g \in \mathcal{G}} \|x_g\|_2 \right\},$$

where $\mathcal{G}$ is the set of group indices. Letting $u^+ = u_k - \eta_k \nabla f(u_k)$, the update takes the form (see, e.g. Sra et al. (2012, Section 2.3)),

$$[u_{k+1}]_g = \left(1 - \frac{\lambda}{\|u_g^+\|_2} \right)_+ u_g^+,$$

which establishes our claim that the proximal step (7) is a thresholding operator.

Thresholding operators are sensitive to rounding and other forms of numerical error. Indeed, it is not hard to see that using a perturbed gradient $\hat{\nabla} f(u_k) = \nabla f(u_k) + \epsilon$ can lead to groups dropping out of the model (or staying in the model) when they should remain non-zero. Thus, it is important to reduce numerical error as much as possible by improving the condition of other operations, like computing $\nabla f(u_k)$. We can use data normalization to partially achieve this goal.

In the remainder of this section, we restrict ourselves to the C-GReLU problem with squared loss,

$$\min_v \frac{1}{2} \left\| \sum_{D_i \in \tilde{\mathcal{D}}} D_i X v_i - y \right\|_2^2 + \lambda \sum_{D_i \in \tilde{\mathcal{D}}} \|v_i\|_2.$$

The Hessian of the smooth component of this problem is $\nabla^2 f(v) = M^\top M$, where M is the "expanded" data matrix $M = \left[D_1 X \; D_2 X \; \ldots D_{|\tilde{\mathcal{D}}|} X\right]$. Let $h \in \mathbb{R}^d, h_i = \|X_{\cdot,i}\|_2$ and $H = \text{diag}(h)$. That is, H is a diagonal matrix with the column-norms of X along the diagonal. Finally, define the column-normalized version of X to be $N = H^{-1}X$.

It is not hard to see that the diagonal elements of $N^\top N$ are 1 by construction. Applying a trace bound, we have

$$\lambda_{\max}(N^\top N) \leq \text{trace}\left(N^\top N\right) = d. \tag{26}$$

Now, consider the normalized version of the expanded data matrix, $\tilde{N} = [D_1 N \; D_2 N \ldots]$. Recalling each D_i is a diagonal matrix whose elements are either 0 or 1, we have

$$N^\top D_i^\top D_i N = N^\top D_i N^\top \preceq N^\top N,$$

for each $D_i \in \tilde{\mathcal{D}}$ and the diagonal elements of this matrix are bounded by 1. We conclude that

$$\lambda_{\max}(\tilde{N}^\top \tilde{N}) \leq \text{trace}\left(\tilde{N}^\top \tilde{N}\right) \leq d \cdot |\tilde{\mathcal{D}}|.$$

This establishes the claim in Section 4.1.1 that data normalization can be used to upper-bound the maximum eigenvalues of the Hessian.

Moving on to computation of the gradient, note that the Hessian $\tilde{N}^\top \tilde{N}$ will be low-rank as long as $|\tilde{\mathcal{D}}| * d > n$. In fact, this is nearly always the case since we typically choose $\tilde{\mathcal{D}}$ to be as large as possible. Thus, although the condition number of $\nabla^2 f(v)$ is not well-defined, it is possible to reduce the maximum expansion entailed by the Hessian via column normalization. Observing $\nabla f(v) = \nabla^2 f(v) v - \tilde{N}^\top y$, we may expect conditioning of the gradient computation to improve. Finally, since the C-GReLU is a linear model, transforming the weights as $v_i' = H^{-1} v_i$ after optimization can be used to project the model back into the original data space, ensuring that data normalization has no effects outside of optimization.

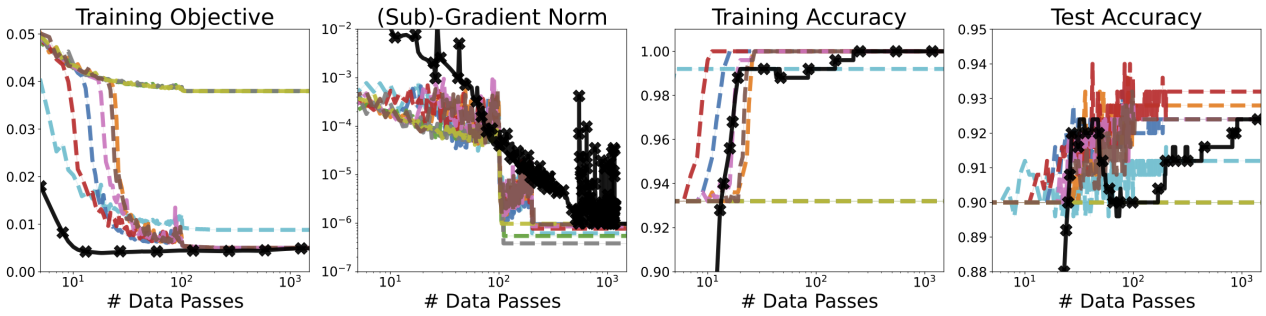

Figure 6. Expanded version of Figure 1 showing the square of the training gradient norm and test accuracy in addition to training objective and training accuracy, as reported in the main paper. Every run of SGD clearly converges to an approximate stationary point before terminating. The "bumps" in gradient norm for the AL method are caused by the dual updates, which increase the norm of the minim-norm subgradient of the augmented Lagrangian ($\mathcal{L}_\delta$) with respect to the primal parameters.

D. Additional Experiments and Experimental Details

Now we provide additional experimental results which were omitted from the main paper due to space constraints. We also give all necessary details to replicate our experiments.

D.1. Synthetic Classification

In this section, we provide additional details and results for the synthetic classification problem shown in Figure 1.

Experimental Details: As mentioned in the main text, we generate the dataset by sampling $X \sim \mathcal{N}(0, \Sigma)$ and then taking $y = \text{sign}(h_{W_1,w_2}(X))$, where h_{W_1,w_2} is a two-layer ReLU network with $m = 100$ and random Gaussian weights. We create 250 training examples and 250 test examples with $d = 50$ in this fashion. The covariance matrix Σ is generated by sampling a random orthonormal matrix of eigenvectors and $d - 2$ eigenvalues from the interval $[1, 10]$. We then append 10 and 1 to this list and form Σ from the diagonalization; this guarantees that the condition number of Σ is exactly 10. Before optimization, we unitize the columns of the feature matrix (see Appendix C.1) to be consistent without our other experiments.

For the non-convex optimization problem, we use the standard PyTorch initialization and a step-size of 10. This step-size gave the fastest convergence out of a grid of $\{20, 10, 5, 1, 0.5, 0.1, 0.01\}$. The mini-batch size is 25 examples (10% of the dataset) and the maximum number of epochs is 1000. We consider SGD to have converged when the gradient norm (as computed by PyTorch) is less than 10^{-3}. We change the global seed at each of the ten different runs, which ensures that both the initialization and mini-batch order/composition are different. For our AL method, we randomly sample 100 "diversity" arrangements, which we augmented will activation patterns generated by SGD while optimizing the non-convex model. Unlike SGD, the randomness in 10 runs of the AL method is due only to (i) sampling of the diversity set and (ii) the sign patterns from the SGD run. Note that we are careful to use exactly the same runs of SGD as described above when using the active set method to compute $\tilde{\mathcal{D}}$. We use the standard parameters as given in Appendix G for the remainder of the AL method's settings.

Additional Results: Figure 6 shows the convergence behavior of our AL and SGD.As in the main paper, we omit all runs of the AL method but one because they are nearly identical. One run of SGD diverges, while nine runs converge to stationary points as measured by the convergence criterion. Of these, four converge to local minima with sub-optimal objective values; these models also do not have 100% accuracy on the training set despite the problem being realizable. These sub-optimal local minimal also give worse test accuracy than the model found by the AL method.

D.2. Large-Scale Comparison

This section gives concrete experimental details for the large-scale comparison of optimization performance presented in Figure 4. We also present the *same* experimental results with different thresholds for success.

D.2.1. Experimental Details

We first provide details required to reproduce Figure 4.

Data: We generate the performance profile using 73 datasets taken from the UCI machine learning repository and six individual regularization parameters for each dataset. See Appendix H for details. This created a set of 438 optimization problems on which we tested the optimization algorithms. Post-optimization, we omit all problems for which a degenerate solution (ie. all weights are zero) is optimal.

Models: We generated $\tilde{\mathcal{D}}$ for the C-ReLU and C-GReLU problems by sampling 5000 and 2500 generating vectors from $z_i \sim \mathcal{N}(0, \mathbf{I})$, respectively, and then computing $D_i = \text{diag}(Xz_i > 0)$. The zero matrix was removed and duplicate patterns were filtered out. The convex formulations were extended to multi-class classification problems as described in Appendix A.1. To ensure the convex and non-convex problems have the same optimal values, we use the vector-out formulation of the NC-ReLU problem given in Equation (16).

We use the exact same activation patterns for NC-GReLU and C-ReLU. We approximately match the NC-GReLU and C-ReLU model spaces by choosing

$$m = \sum_{D_i \in \tilde{\mathcal{D}}} \left| \{v_i^* : v_i^* \neq 0\} \cup \{w_i^* : w_i^* \neq 0\} \right|.$$

Recall from Theorem 2.1 that this choice ensures the model space for C-ReLU is a strict subset of that for NC-ReLU. Thus, our results can only favor the non-convex formulations.

Optimizers: For models with gated ReLU activations, we compare R-FISTA with default parameters (see Appendix G) to Adam, SGD, and MOSEK.

We use a mini-batch size of $\frac{n}{10}$ for Adam and SGD and perform a grid-search over the following set of step-sizes:

$$\eta \in \{10, 5, 1, 0.5, 0.1, 0.01, 0.001\}.$$

For each optimization problem, we choose the step-size which gives the smallest final training objective. We also use a decay schedule that halves the step-size every 100 epochs; experimentally, this "step" schedule worked much better than classical schedules of the form $\eta_k = \eta_0 * t^{-r}$, $r > 0.5$. To control for stochasticity, Adam and SGD are run with three independent random seeds and only the best execution is reported. MOSEK is run with the default configuration using CVXPY as an interface (Diamond & Boyd, 2016; Agrawal et al., 2018); we only use MOSEK on the convex reformulation.

The same experimental procedure is used for Adam and SGD on models with ReLU activations, We use our AL method with standard parameters (Appendix G) to solve the convex reformulation and MOSEK is again used with standard parameters to solve the convex reformulation.

Hardware and Timing: R-FISTA, our AL method, Adam, and SGD are run on GPU compute nodes with one GeForce RTX 2080Ti graphics card, two AMD 7502P CPUs, and 8 GB of RAM. Note that the GPUs themselves have 11 GB of GPU RAM.MOSEK cannot be run on GPUs, so instead these experiments are executed on CPU nodes with 32 GB of RAM and 4 AMD EPYC 7502 CPUs, each of which have 32 cores. In practice, we observed extremely small variance when timing identical runs. As such, we do not average times over multiple runs.

Determining Successes: We use the (sub)-optimality gap $F(x_k) - F(x^*)$ to determine if optimization is successful. In particular, the relative optimality gap can be checked as

$$\Delta_k := \frac{F(x_k) - F(x^*)}{F(x^*)} \leq r_{\text{gap}},$$

for some threshold r_{gap}. Figure 4 reports results for $r_{\text{gap}} = 1$. For fairness, we provide figures generated from the same experimental data with different choices of r_{gap} in the next sub-section. Finally, runs which exceed their available memory and crash are considered failures, as are problems which take more than 15 minutes. In practice, this is only applicable for MOSEK, which scales poorly in both memory and time.

D.2.2. Additional Results

Alternative Success Thresholds: Choosing the threshold for the relative optimality gap Δ_k is subjective and can potentially favor some methods over others. In this section, we show that alternative values of r_{gap} preserve the ordering of methods from Figure 4. In particular, tightening the threshold shows that our convex solvers are not only faster than the non-convex baselines, but also solve the optimization problems to greater accuracy.

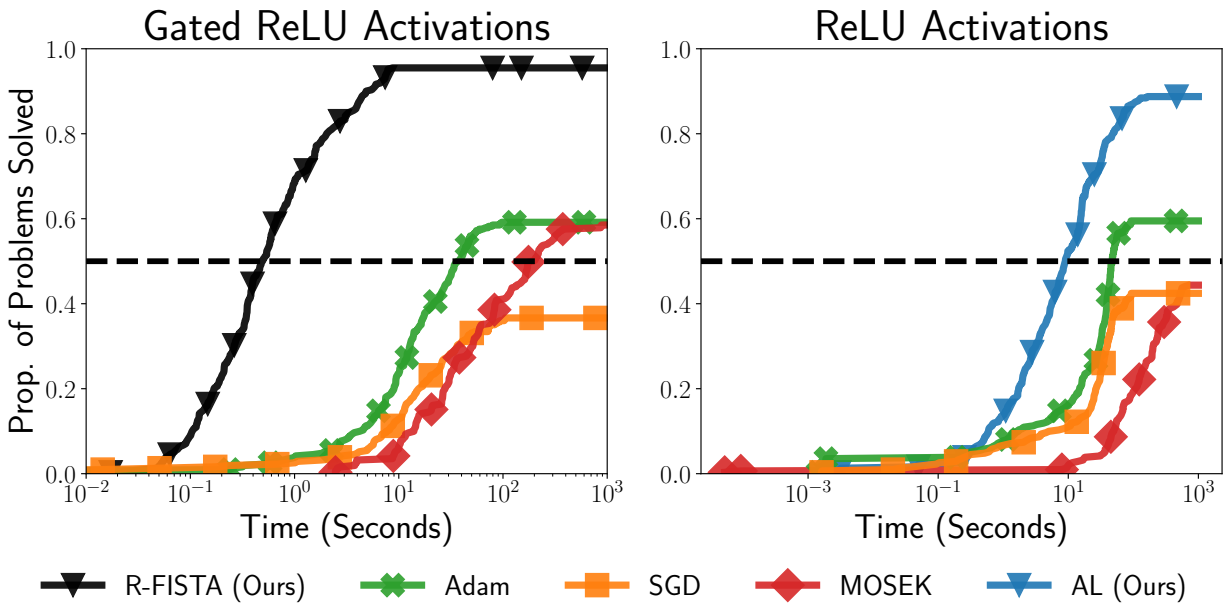

Figure 7. Alternative version of Figure 4 with success threshold set to be $r_{\text{gap}} = 0.5$. Recall that a problem is considered solved when $(F(x_k) - F(x^*))/F(x^*) \leq r_{\text{gap}}$, where $F(x^*)$ is the smallest objective value found by any method. We find that tightening the success threshold (as compared to Figure 4) only improves the performance of our convex solvers relative to Adam and SGD.

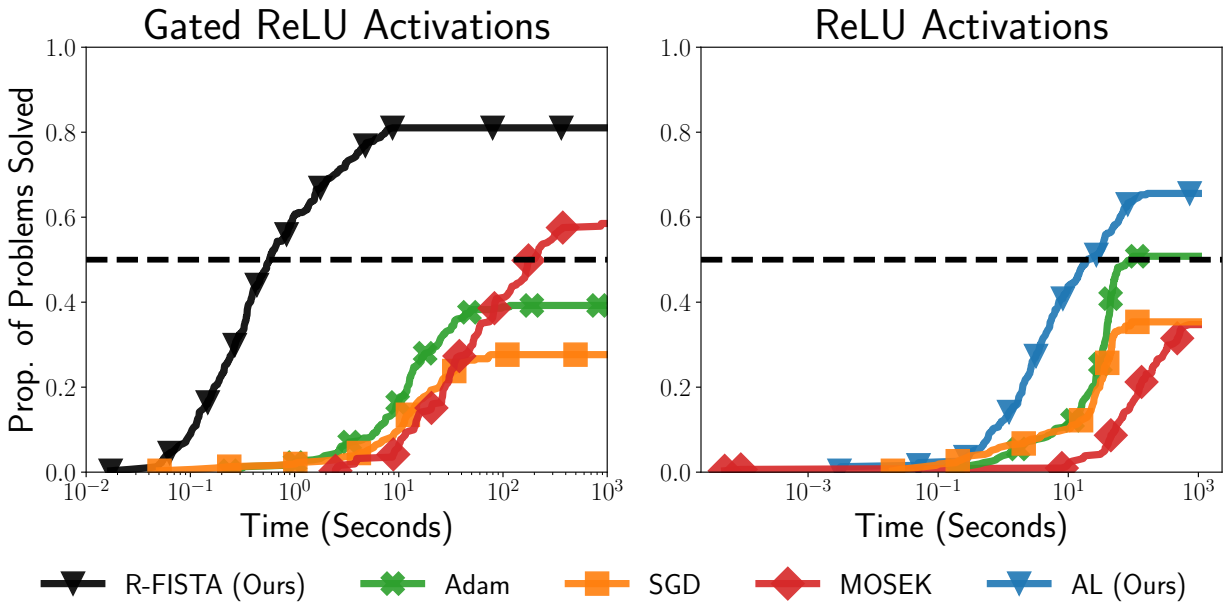

Figure 8. Alternative version of Figure 4 with success threshold set to be $r_{\text{gap}} = 0.1$. Performance of all optimization methods decreases at this threshold, with the notable exception of MOSEK for the C-GReLU problem. However, the relative ordering of R-FISTA, AL, Adam, and SGD remains unchanged. In other words, our optimizers applied to the convex reformulations still out-perform the non-convex baselines.

Table 5. Approximating the C-ReLU problem with cone decompositions. We compare the solution to C-GReLU (FISTA) with cone-decomposition by solving the min-norm program (CD-SOCP), the approximate cone decomposition (CD-A), and directly solving C-ReLU using the AL method. We report the norm of each model to demonstrate the "blow-up" effect of the cone decomposition.

	R-FISTA			**CD-SOCP**			**CD-A**			**AL**		
Dataset	Acc.	Time	Norm	Acc.	Time	Norm	Acc.	Time	Norm	Acc.	Time	Norm
breast-cancer	68.4	0.84	1.06×10^{2}	68.4	12.14	3.73×10^{3}	68.4	3.18	4.76×10^{3}	66.7	6.17	5.84×10^{1}
congressional	64.4	0.89	4.20×10^{1}	64.4	80.69	1.90×10^{3}	97.3	5.26	3.5×10^{3}	69.0	18.32	3.48×10^{1}
sonar	87.8	0.45	2.25×10^{1}	90.2	23.13	1.28×10^{2}	64.4	1.09	1.83×10^{2}	87.8	4.34	2.33×10^{1}
credit	82.6	0.44	7.32×10^{1}	83.3	63.46	3.97×10^{3}	84.1	5.1	4.76×10^{3}	84.1	6.46	5.11×10^{1}
cylinder	75.5	1.18	9.64×10^{1}	77.5	79.78	2.09×10^{3}	75.5	2.68	2.61×10^{3}	75.5	15.18	1.11×10^{2}
ecoli	71.6	0.07	1.73×10^{1}	71.6	149.7	5.82×10^{2}	70.1	0.36	1.13×10^{2}	70.1	3.38	1.68×10^{1}
energy-y1	86.3	0.12	2.90×10^{1}	86.3	134.55	2.34×10^{3}	86.3	2.01	1.34×10^{3}	83.7	5.05	2.89×10^{1}
glass	64.3	0.13	2.00×10^{1}	64.3	68.76	4.05×10^{2}	64.3	0.69	2.31×10^{2}	61.9	3.0	1.72×10^{1}
cleveland	51.7	0.13	1.97×10^{1}	51.7	109.4	2.52×10^{2}	51.7	0.6	2.18×10^{2}	50.0	1.82	1.77×10^{1}
hungarian	86.2	0.88	8.01×10^{1}	86.2	17.15	2.75×10^{3}	86.2	4.09	3.59×10^{3}	84.5	7.01	5.09×10^{1}
heart-va	35.0	0.16	2.44×10^{1}	37.5	67.11	4.79×10^{2}	37.5	0.9	4.73×10^{2}	37.5	1.86	1.58×10^{1}
hepatitis	80.6	1.06	2.96×10^{1}	80.6	10.57	2.63×10^{2}	80.6	1.97	3.4×10^{2}	74.2	8.56	5.53×10^{1}
horse-colic	85.0	1.0	5.79×10^{1}	86.7	29.42	9.45×10^{2}	86.7	4.54	1.36×10^{3}	90.0	9.75	8.92×10^{1}
ionosphere	90.0	1.15	4.74×10^{1}	90.0	44.12	1.51×10^{3}	90.0	5.76	2.05×10^{3}	90.0	15.91	6.42×10^{1}
mammograph	78.1	0.24	4.13×10^{1}	78.1	33.17	8.11×10^{3}	78.1	5.18	3.02×10^{3}	79.2	5.14	3.70×10^{1}
monks-2	60.6	1.95	1.02×10^{2}	60.6	5.81	7.36×10^{3}	57.6	6.61	9.43×10^{3}	45.5	18.31	8.13×10^{1}
monks-3	87.5	1.6	5.59×10^{1}	87.5	3.89	3.44×10^{3}	87.5	6.49	4.49×10^{3}	95.8	31.84	8.60×10^{1}
oocytes	78.6	0.98	1.14×10^{2}	79.1	136.3	1.19×10^{4}	78.0	5.67	1.3×10^{4}	74.2	81.68	8.54×10^{1}
parkinsons	92.3	1.9	4.70×10^{1}	92.3	16.03	1.06×10^{3}	92.3	4.84	1.45×10^{3}	89.7	19.04	6.69×10^{1}
pima	73.2	0.36	7.88×10^{1}	73.2	37.68	8.09×10^{3}	73.2	5.07	7.45×10^{3}	75.8	4.72	4.03×10^{1}
planning	63.9	1.33	8.94×10^{1}	63.9	10.54	2.16×10^{3}	63.9	3.07	3.05×10^{3}	58.3	9.74	1.09×10^{2}
seeds	95.2	0.25	1.91×10^{1}	95.2	26.71	4.83×10^{2}	95.2	1.99	4.17×10^{2}	95.2	5.16	1.82×10^{1}
australian	64.5	0.69	1.58×10^{2}	63.8	55.59	1.06×10^{4}	64.5	5.46	1.21×10^{4}	65.2	4.8	2.74×10^{1}
statlog-heart	81.5	0.81	6.76×10^{1}	81.5	15.57	1.48×10^{3}	81.5	2.5	2.02×10^{3}	85.2	7.36	6.11×10^{1}
teaching	40.0	0.24	3.71×10^{1}	40.0	12.18	1.10×10^{3}	40.0	1.75	1.14×10^{3}	33.3	2.05	2.21×10^{1}
tic-tac-toe	97.9	0.45	1.70×10^{2}	97.9	60.36	6.86×10^{3}	97.9	4.38	7.6×10^{3}	93.7	14.25	1.57×10^{2}
vertebral-col.	88.7	0.68	7.41×10^{1}	88.7	9.64	7.76×10^{3}	88.7	5.54	6.73×10^{3}	90.3	8.23	4.74×10^{1}
wine	100	0.25	1.87×10^{1}	100	32.05	1.97×10^{2}	100	0.95	2.42×10^{2}	100	3.67	1.78×10^{1}

Figures 7 and 8 present the same experimental results as in Figure 4 with $r_{\text{gap}} = 0.5$ and $r_{\text{gap}} = 0.1$, respectively. They should be compared against the threshold value of $r_{\text{gap}} = 1$ used in the main paper. These figures show that the relative performance of each optimization method remains unchanged as the threshold is decreased, with the notable exception of MOSEK. This is because MOSEK uses a highly accurate, but slow, interior point method.

D.3. Cone Decompositions

Now we provide details and additional results for the cone-decomposition experiments given in Table 1.

Experimental Details: We selected 23 datasets from the UCI repository and fixed the regularization parameter at $\lambda = 0.01$. Note that this parameter is not necessary optimal for each dataset; the purpose of these experiments is to study the effects of using cone-decompositions to approximate the C-ReLU solution with a G-ReLU solution, rather than to obtain optimal test accuracies. We randomly sampled 1000 activation patterns for the C-ReLU and C-GReLU models, removing duplicates and the zero pattern as necessary. Note that we report the median results from five individual runs with re-sampled activation patterns to control for variance in the procedure.

For multi-class datasets, the convex formulations were extended as described in Appendix A.1. We used the standard parameters for R-FISTA and the AL method as given in Appendix G, while the min-norm decomposition programs (CD-SOCP) was solved with MOSEK using the default parameters. For CD-A, we set $\lambda = 10^{-10}$ and Equation (9) with R-FISTA using the default parameters. We terminate the optimization procedure when the min-norm subgradient has squared-norm less than or equal to 10^{-10}. R-FISTA and the AL method were run on GPU compute nodes with one GeForce RTX 2080Ti graphics card, and four AMD 7502P CPUs, and 32 GB of RAM. The cone decompositions were solved on identical nodes with four AMD 7502P CPUs with 32 GB of RAM.

Additional Results: Table 5 provides the full set of results on all 23 datasets. It also includes the final group norms of the models, calculated as

$$\sum_{D_i \in \tilde{\mathcal{D}}} \|u_i^*\|_2 \,,$$

for the C-GReLU model and

$$\sum_{D_i \in \tilde{\mathcal{D}}} \|v_i^*\|_2 + \|w_i^*\|_2 \,,$$

for the C-ReLU models. This allows us to quantify the "blow-up" in the model norm from decomposing u_i^* onto $\mathcal{K}_i - \mathcal{K}_i$. In practice, we find that CD-SOCP leads to very large increases in the model norm compared to the FISTA/AL solutions, while CD-A has a less severe effect. However, the increased norms do not appear to affect the test accuracy of the final models. Indeed, CD-SOCP and CD-A perform as well as the solution to the C-GReLU problem given by R-FISTA and are comparable to the AL method's solution. The major downside of exact the cone-decomposition method is the huge increase in time necessary to solve for the decomposition. This is largely because MOSEK is restricted to running on CPU.

D.4. The Role of Acceleration and other Algorithmic Components

This section studies the effects of different algorithmic components on the optimization performance of R-FISTA for the C-GReLU problem. By systematically removing restarts, acceleration, and line-search, we illustrate the importance of these enhancements to the speed and robustness of the optimization procedure.

Figure 9 shows a performance profile comparing R-FISTA, the FISTA algorithm without restarts (FISTA), proximal gradient descent with the line-search described in Section 4.1.1 (PGD-LS), and proximal gradient descent (PGD) with a fixed step-size. We use the same problem set as for Figure 4: 438 individual training problems generated by considering six regularization parameters for 73 datasets taken from the UCI dataset repository. See Appendix H for more details. Note that we do not include problems for which the regularization parameter is overly large and a degenerate model (ie. all zeros) is optimal. A problem is considered solved the minimum norm subgradient has norm less than or equal to 10^{-3}; in practice, we check an identical condition on the gradient norm squared. The C-GReLU model is formed by sampling 5000 activation patterns.

The x-axis shows the number of passes through that dataset that each method performs. This quantity is equivalent to the iteration counter for PGD; for the remaining methods it also includes the number of function evaluations due to back-tracking on the line-search condition. For R-FISTA, FISTA, and PGD-LS, we use the step-size initialization strategy described in the main paper (see Appendix D.5 for experiments studying this rule) with the standard parameters given in Appendix G. For each problem, we use the best fixed step-size for PGD out of the grid $\{10, 1, 0.1, 0.01\}$.

We make the following observations: (i) R-FISTA requires about three-fourths as many data passes as FISTA to solve 80% of problems, which suggests restarts allow greater adaptivity to problem structure; (ii) acceleration is critical to solving problems quickly and PGD-LS performs poorly compared to both R-FISTA and FISTA; (iii) PGD is very slow despite using about $4\times$ more compute than the other methods.

We also report convergence behavior on two randomly selected datasets to illustrate the fine-grained performance of each method. Figures 10 and 11 show the convergence of R-FISTA, FISTA, PGD-LS, and PGD with respect to objective value and subgradient norm (squared) for the `twonorm` and `heart-cleveland` datasets. Results for are shown for the smallest regularization parameter considered and the largest for which the model was not degenerate. We omit step-sizes for which PGD diverged.

D.5. Step-size Update Rules

Now we perform an ablation study on the step-size initialization rule proposed by Liu et al. (2009) and discussed in Section 4.1.1. Throughout this section, we refer to this initialization strategy as quadratic-bound (QB). We compare QB against warm starting as $\eta_k = \eta_{k-1}$ (WS), and forward tracking (FT). As in the previous section, we use a performance profile to summarize results for solving the C-GReLU problem on the same 438 problems as in Figure 4. We use the same backtracking parameter $\beta = 0.8$ for QB, WS, and FT, while we use a forward-tracking parameter of $\alpha = 1.25$ for QB and FT. Note that these are the standard parameters discussed in Appendix G. We use the standard settings for all other parameters of R-FISTA. We sample 5000 random activation patterns just as in the previous section.

Empirically, we find (see Figure 12) that the QB initialization strategy is surprisingly resilient to the choice of threshold parameter, c. Indeed, QB with any $c \in \{10, 5, 2\}$ is more efficient than FT or WS. Surprisingly, FT and WS have similar

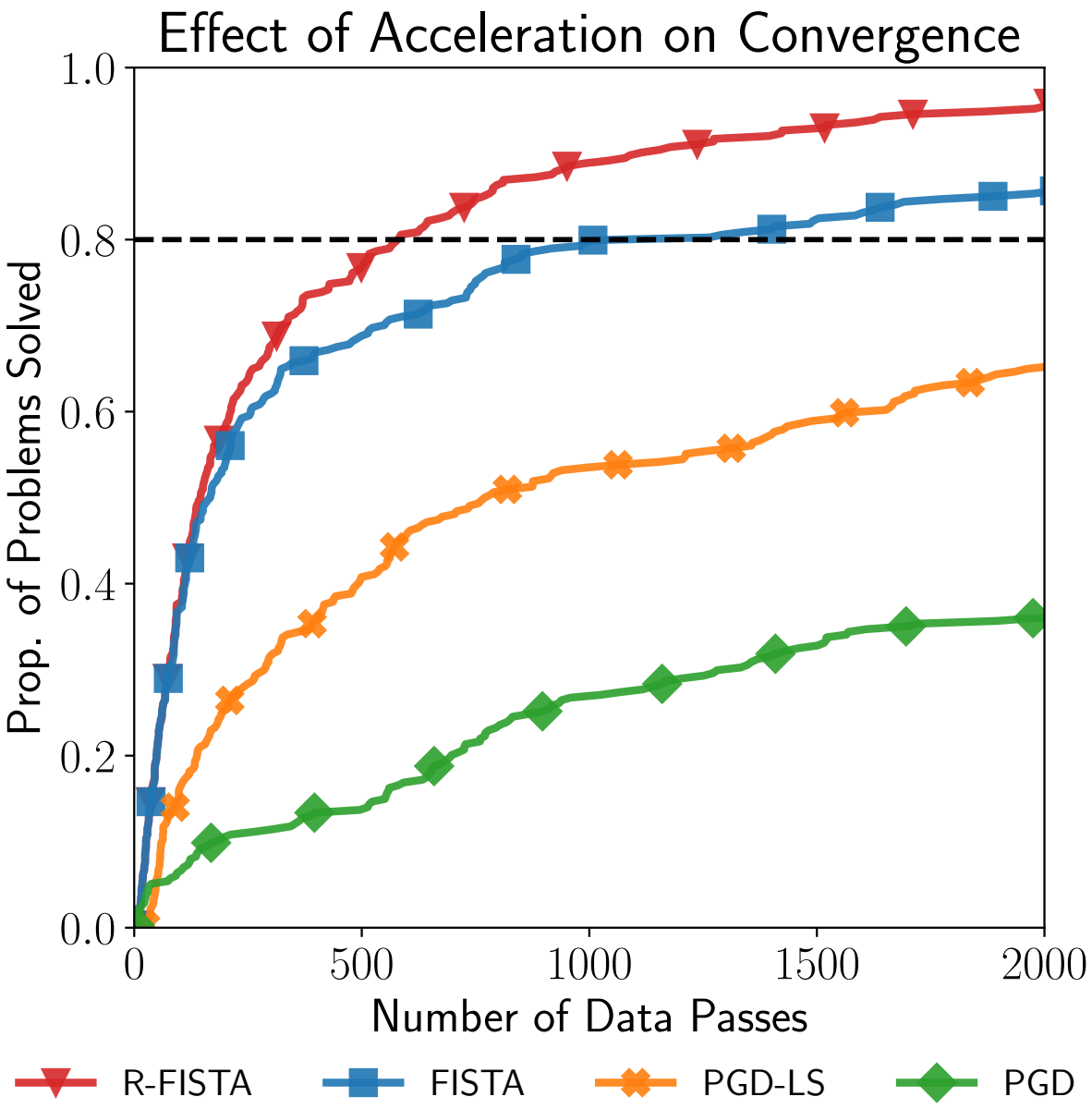

Figure 9. Performance profile comparing R-FISTA, FISTA without restarts, proximal gradient descent with line-search (PGD-LS) and proximal gradient descent with a fixed step-size (PGD) for solving C-GReLU on 73 datasets from the UCI repository. For PGD, we report results for the best step-size chosen by grid-search individually for each problem. R-FISTA solves a higher proportion of problems in fewer passes through the dataset.

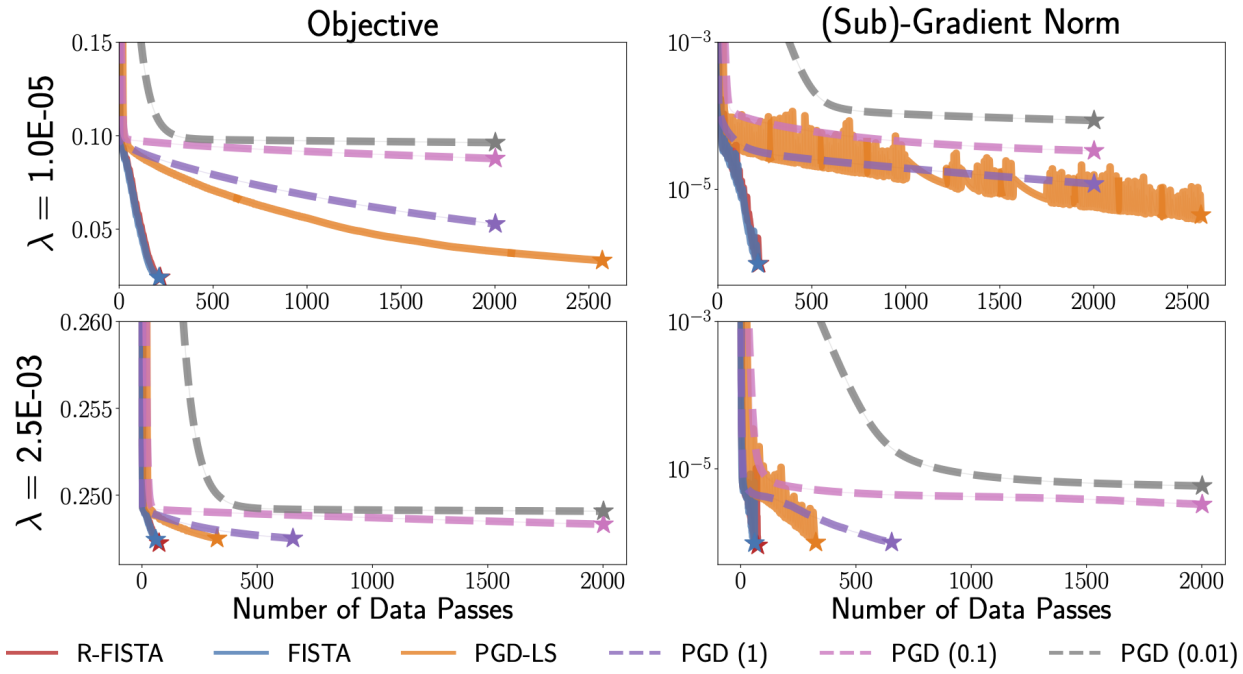

Figure 10. Convergence comparison for R-FISTA, FISTA without restarts (FISTA), proximal gradient descent with line-search (PGD-LS) and proximal gradient descent (PGD) with several fixed step-sizes (reported in parenthesis) on the `twonorm` dataset. PGD stalls while the accelerated methods converge very quickly to an approximate stationary point.

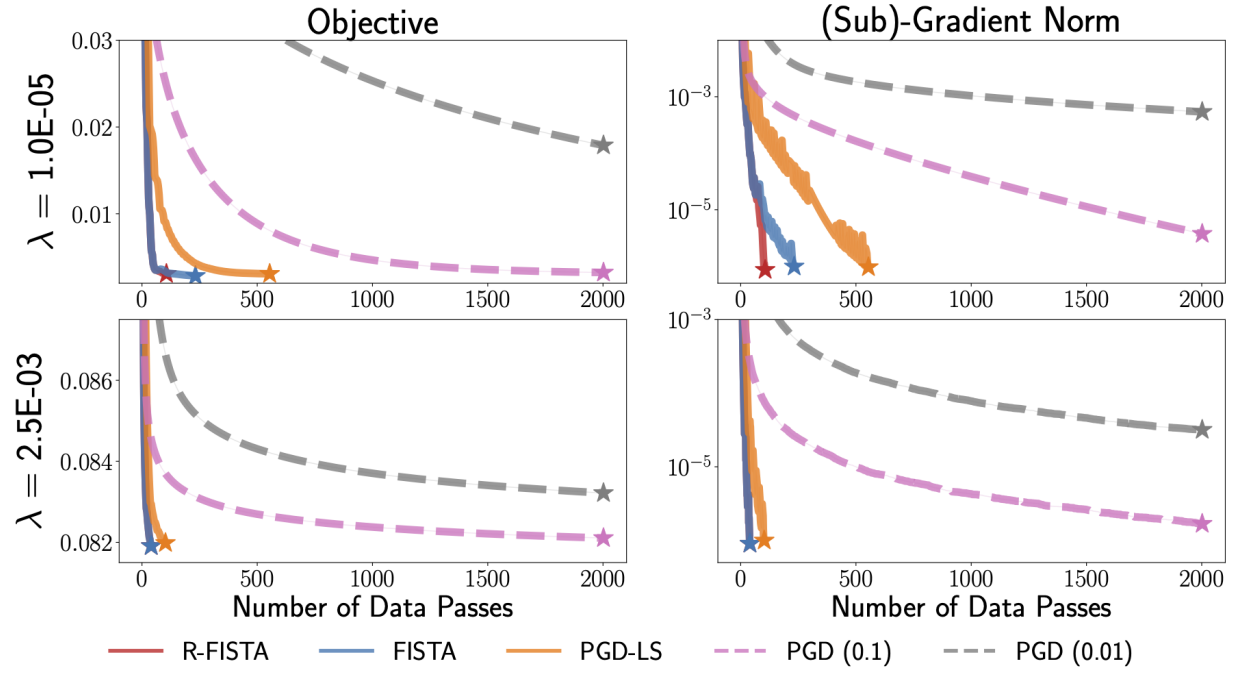

Figure 11. Convergence comparison for R-FISTA, FISTA without restarts (FISTA), proximal gradient descent with line-search (PGD-LS) and proximal gradient descent (PGD) with several fixed step-sizes (reported in parenthesis) on the `heart-cleveland` dataset. The performance of R-FISTA and FISTA is identical when $\lambda = 2.5 \times 10^{-3}$. In contrast, restarting allows R-FISTA to converge in around half as many iterations as FISTA for the smoother problem with $\lambda = 1 \times 10^{-5}$.

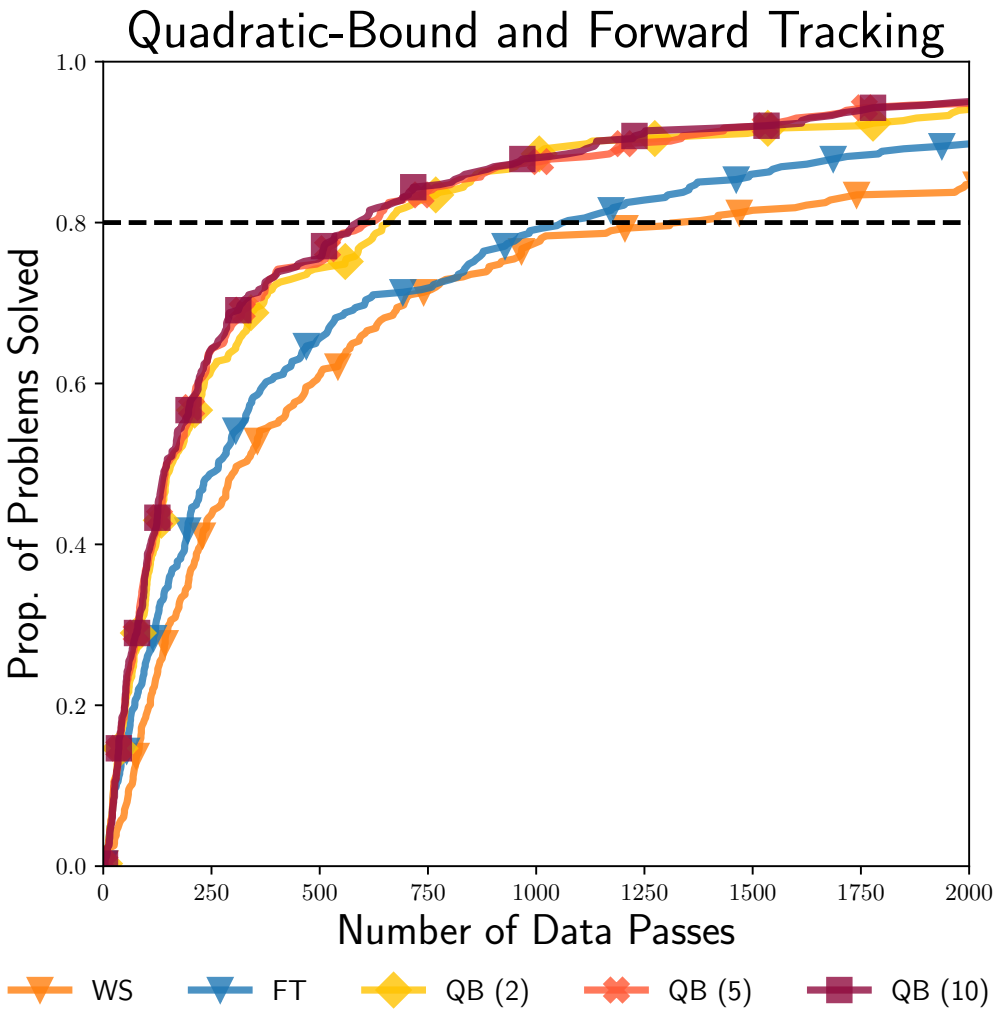

Figure 12. Performance profile comparing R-FISTA, FISTA with different step-size initialization rules. We compare checking the quadratic bound for tightness (QB) with a variety of choices for the threshold parameter c against warm starting as $\eta_k = \eta_{k-1}$ (WS), and forward tracking (FT). As before, we generate the profile by solving the C-GReLU problem on 73 datasets from the UCI repository. QB is robust to the choice of c and outperforms both WS and FT. WS and FT have similar performance despite very different behavior.

performance despite their substantially different convergence behavior (see Figures 13 and 14). This is primarily because we measure progress in total data passes, which includes the unnecessary backtracking performed by R-FISTA with the FT update.

D.6. The Windowing Heuristic

Recall that the key hyper-parameter for our AL method is the penalty strength, denoted δ. Here we verify the effectiveness of the windowing heuristic for selecting δ as proposed in Section 4.3.1. Experimentally, the rule performs nearly as well as the best fixed value of δ across a wide range of datasets and avoids the catastrophic failures which can occur when δ is miss-specified.

We initialize our AL method with $\delta_0 \in \left\{1, 10, 10^2, 10^3, 10^4\right\}$ and compare tuning δ using the windowing heuristic against keeping δ fixed throughout optimization. All other parameters are identical and constant for the two approaches (see Appendix G for specifics). To evaluate speed and robustness, we use another performance profile on the 438 problems generated from the UCI datasets as detailed in Appendix H. In this case, a problem is considered "solved" when the minimum-norm subgradient of the augmented Lagrangian is smaller than 10^{-3} and the norm of the constraint gaps is also

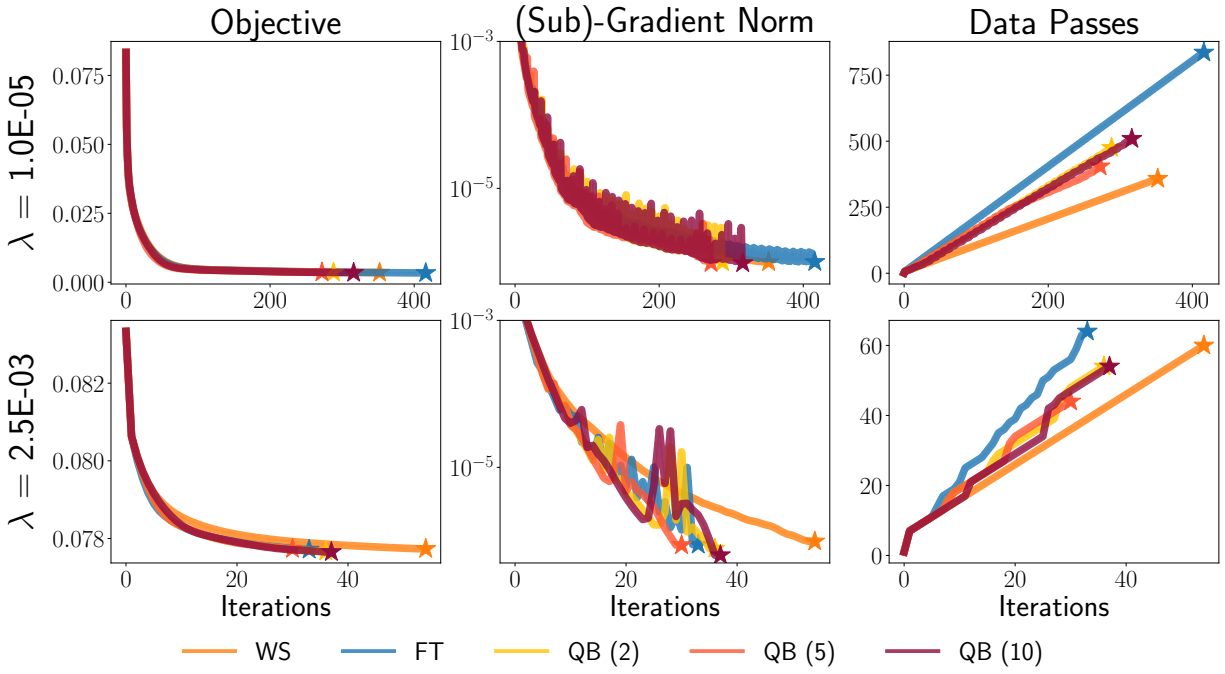

Figure 13. Convergence comparison for R-FISTA with different step-size initialization rules on the `glass` dataset. We compare warm-starting (WS) and forward-tracking (FT) against the initialization proposed by Liu et al. (2009) (QB) for several fixed thresholds (reported in parentheses). QB has similar convergence performance to FT without requiring as many passes through the training set and is resilient to the choice of threshold.

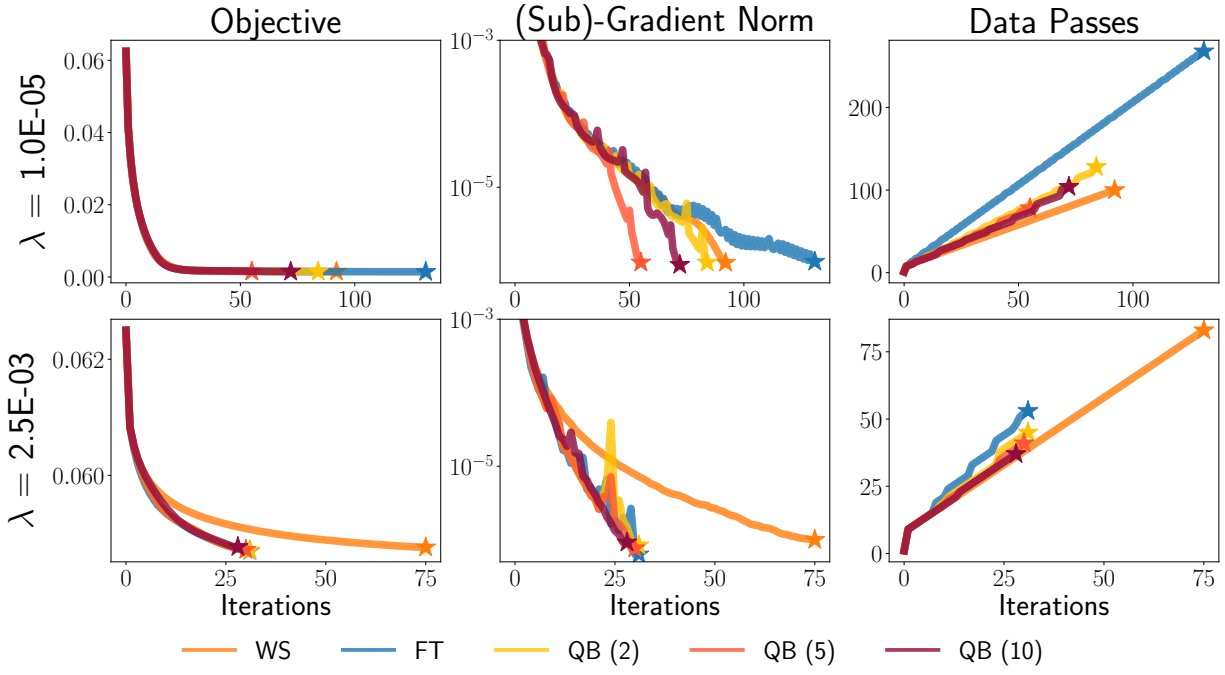

Figure 14. Convergence comparison for R-FISTA with different step-size initialization rules on the `flags` dataset. See Figure 13 for additional details.

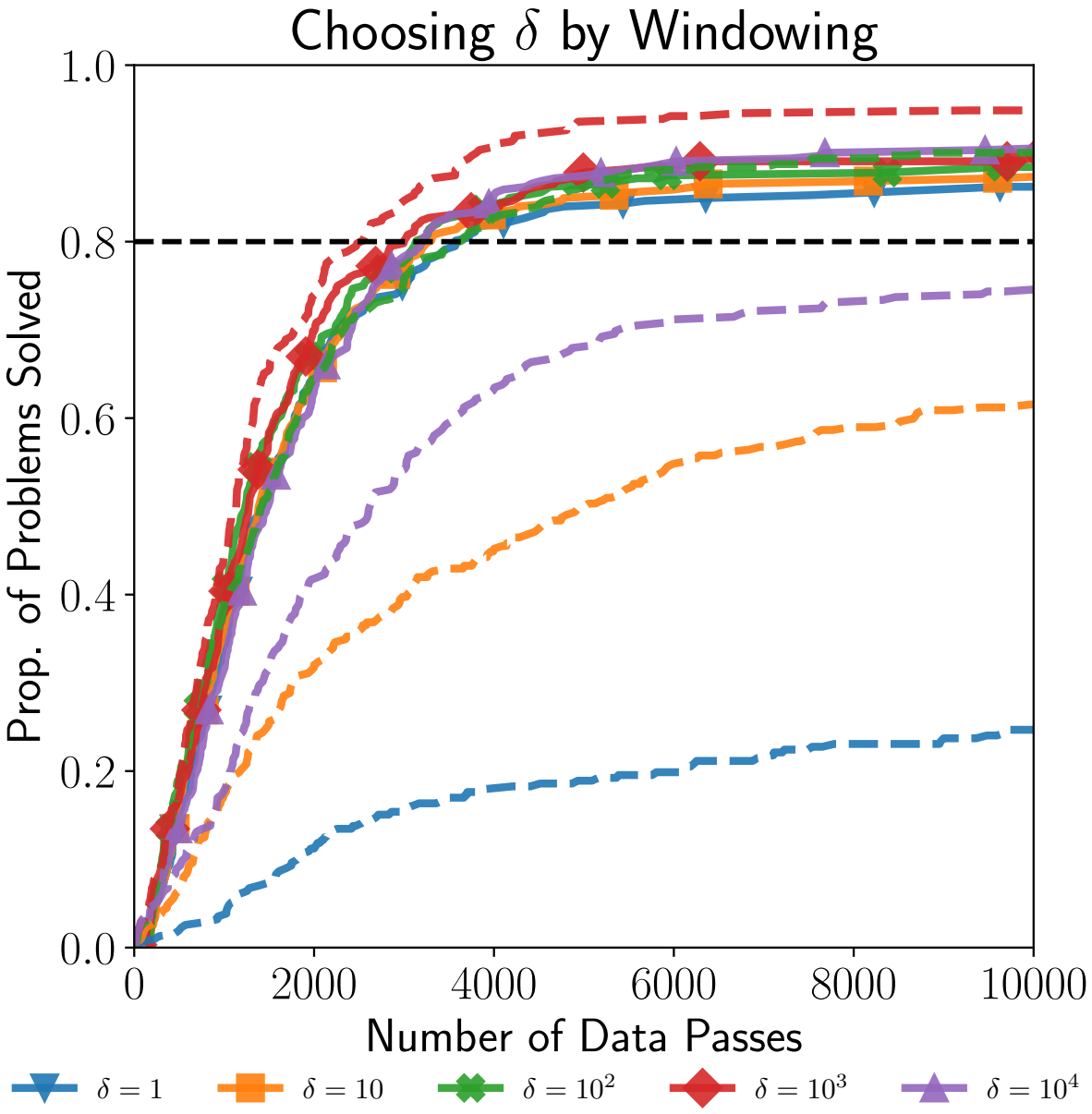

Figure 15. Performance profile comparing our AL method with (solid lines with markers) and without (dashed lines) the windowing heuristic for setting the penalty strength. We consider a wide range of initial δ values and generate the profile by solving the C-ReLU problem on 73 datasets from the UCI repository with 6 different regularization parameters for each dataset. The windowing heuristic performs nearly as well as the best fixed δ and without a noticeable computational overhead. In contrast, extreme values of δ can cause the "fixed" approach to fail on approximately 40% of problems.

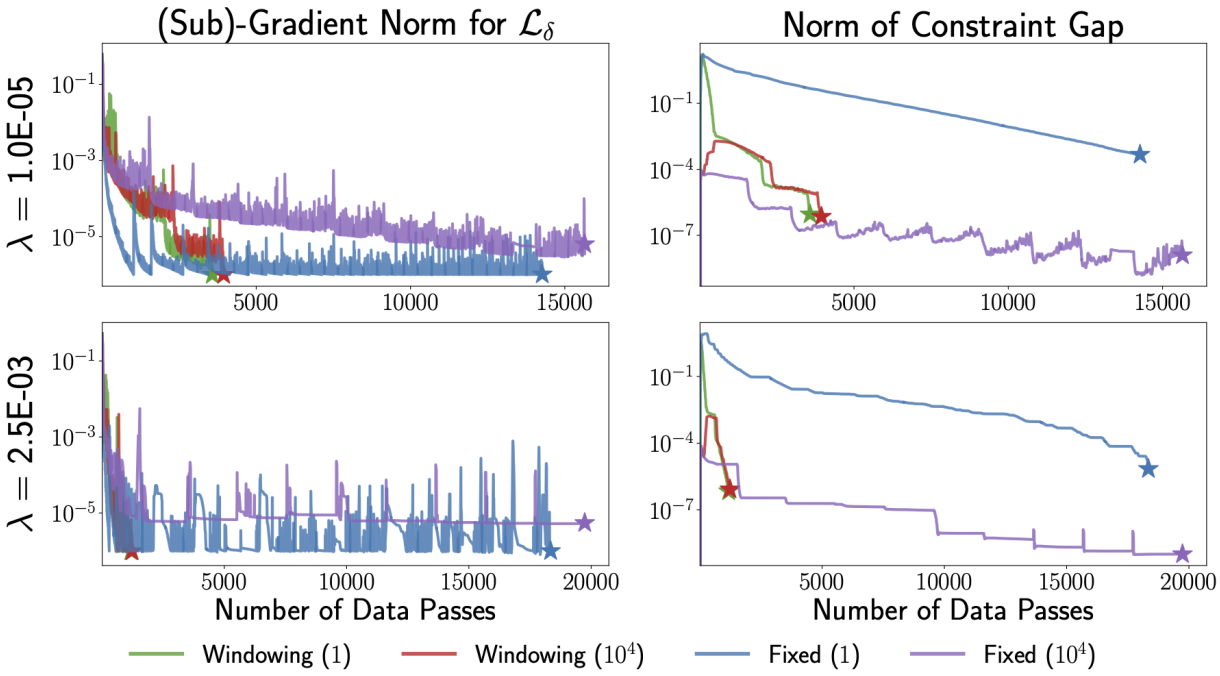

Figure 16. Convergence comparison for our AL method with with and without the windowing heuristic on the `monks-2` dataset. We show two extreme values of δ (shown in parentheses) to illustrate failure models the AL method without our heuristic. Roughly, each "bump" in the subgradient norm (squared) of $\mathcal{L}_\delta$ corresponds to one prox-point iteration on the dual parameters (i.e. an AL update). Observe that when δ is small, R-FISTA solves sub-problem (11) quickly, but the AL method cannot find a feasible solution without an extreme number of dual updates. In contrast, R-FISTA never solves (11) to the first-order tolerance (10^{-6}) when δ is very large. In this case, dual updates are triggered only by a limit on the number of R-FISTA iterations for solving the sub-problem. See Appendix G. The windowing heuristic corrects both failure modes.

less than 10^{-3}. This isn't equivalent to terminating when the Lagrangian function is approximately stationary, but we found the rule to work well in practice. We use 500 randomly sampled activation patterns for the C-ReLU model.

Figure 15 plots the result, with dotted lines for the AL method with fixed δ and solid lines with markers for methods using the windowing heuristic. Empirically, the windowing heuristic is nearly effective as the best fixed δ and avoids the complete failure of AL methods with fixed, poorly specified penalty parameters (e.g. $\delta = 1$ or $\delta = 10^4$). Moreover, this is achieved at almost no overhead in terms of total data passes required for convergence. Finally, we observe that fixing $\delta = 10^3$ works very well across all problems; this is likely because the problems are carefully normalized before optimization to ensure they are on the same scale. Specifically, the columns of the data matrix for each problem are unitized (Appendix C.1), and the augmented Lagrangian $\mathcal{L}_\delta$ is normalized by $n * k$, where k is the number of classes.

We also provide convergence plots on two randomly selected datasets to better illustrate the failures modes of the AL method with miss-specified penalty strength. Figures 16 and 17 and show detailed results for the `monks-2` and `ilpd-indian-liver` datasets. When δ is too small, the AL method easily solves subproblem (11), but struggles to make progress on the constraint gaps. Intuitively, the step-size for the dual proximal-point algorithm is too small and a very large number of iterations is required to make progress on the dual problem. Conversely, the augmented Lagrangian $\mathcal{L}_\delta$ is poorly conditioned when δ is overly large and R-FISTA struggles to solve the primal sub-problem to the necessary tolerance. The windowing heuristic corrects for both pathologies by ensuring the initial constraint gap is in a "normal" regime that balances penalizing constraint violations and conditioning of the subproblem. This behavior is particularly noticeable for `monks-2`, where the windowing heuristic adjusts δ to shrink the constraint gap ($\delta = 1$) or relax the optimization problem ($\delta = 10^4$).

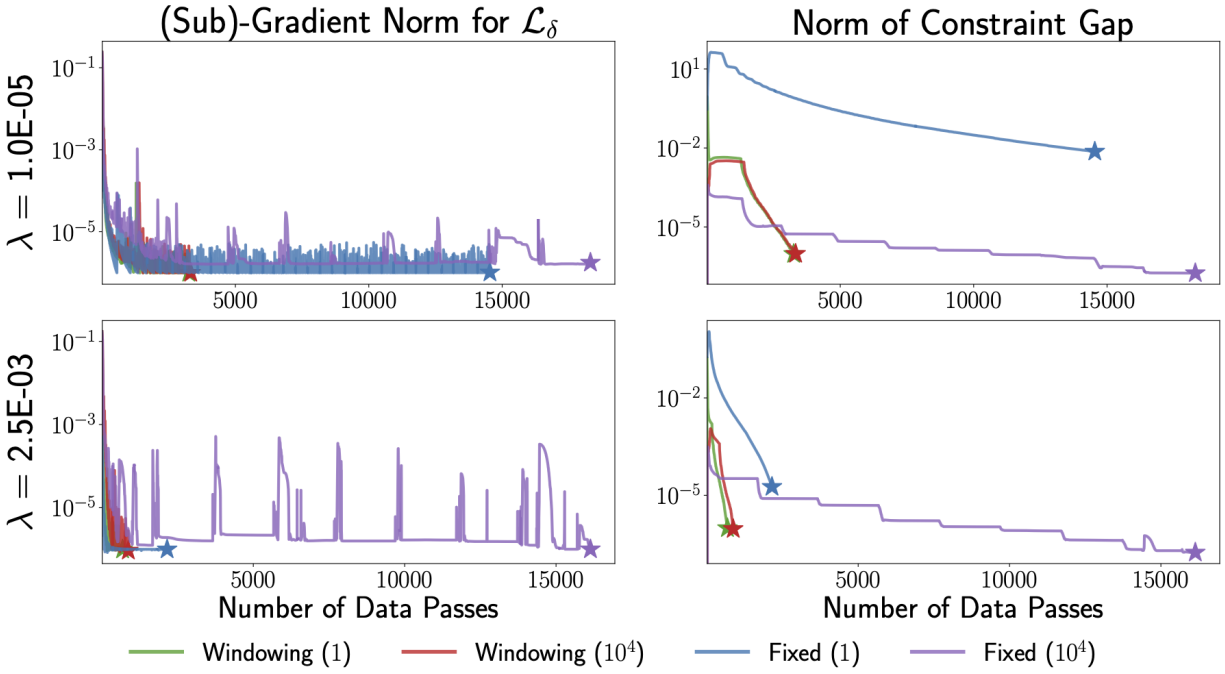

Figure 17. Convergence comparison for our AL method with with and without the windowing heuristic on the `ilpd-indian-liver` dataset. Penalty parameters δ are reported in parenthesis.

E. Sensitivity and Regularization

This section presents additional ablations studying the sensitivity of the C-ReLU and C-GReLU problems to the selection of the sub-sampled activation patterns, $\tilde{\mathcal{D}}$, and the regularization strength, λ.

Experimental Details: We randomly select 10 datasets from our set of 73 filtered UCI datasets (see Appendix H). For each dataset, we considered thirty individual regularization parameters on log-scale grid over the interval $[1 \times 10^{-6}, 1]$. To form the convex formulations, we computed $\tilde{\mathcal{D}}$ by sampling 10, 100, or 1000 generating vectors from $\mathcal{N}(0, \mathbf{I})$. We repeated the sampling procedure with 10 different random seeds, giving a final total of 60 (30 C-ReLU and 30 C-GReLU) optimization problems for each dataset. These problems were then solved using R-FISTA and our AL method with the default parameters (see Appendix G).

Additional Results: Figures 18 and 19 present results for the C-GReLU and C-ReLU problems, respectively. Similar to Figure 5, a U-shaped bias-variance trade-off is visible as the regularization strength is increased. This trend is especially noticeable for the `monks-3` and `statlog-heart` datasets. Variance introduced by sampling $\tilde{\mathcal{D}}$ is only significant for `heart-va`.

E.1. UCI Classification

This section gives experimental details and additional results for the experiments evaluating generalization performance of the convex reformulations.

Experimental Details: We selected 37 binary classification datasets from our filtered collection of 73 datasets; see Appendix H for details how the 73 datasets were obtained.

We used the default parameters for each of the convex solvers as described in Appendix G, except that a tighter convergence tolerance of 10^{-7} was used for terminating our methods. R-FISTA was limited to 2000 iterations. For the gated ReLU problems (both C-GReLU and NC-GReLU) we sampled the same set of 5000 activation patterns for both the convex reformulation and the original non-convex model. We used 2500 activation patterns for the C-ReLU problem.

For each dataset-method pair, we performed five-fold cross validation on the training set to select hyper-parameters. We considered two hyper-parameters for our methods: regularization strength,

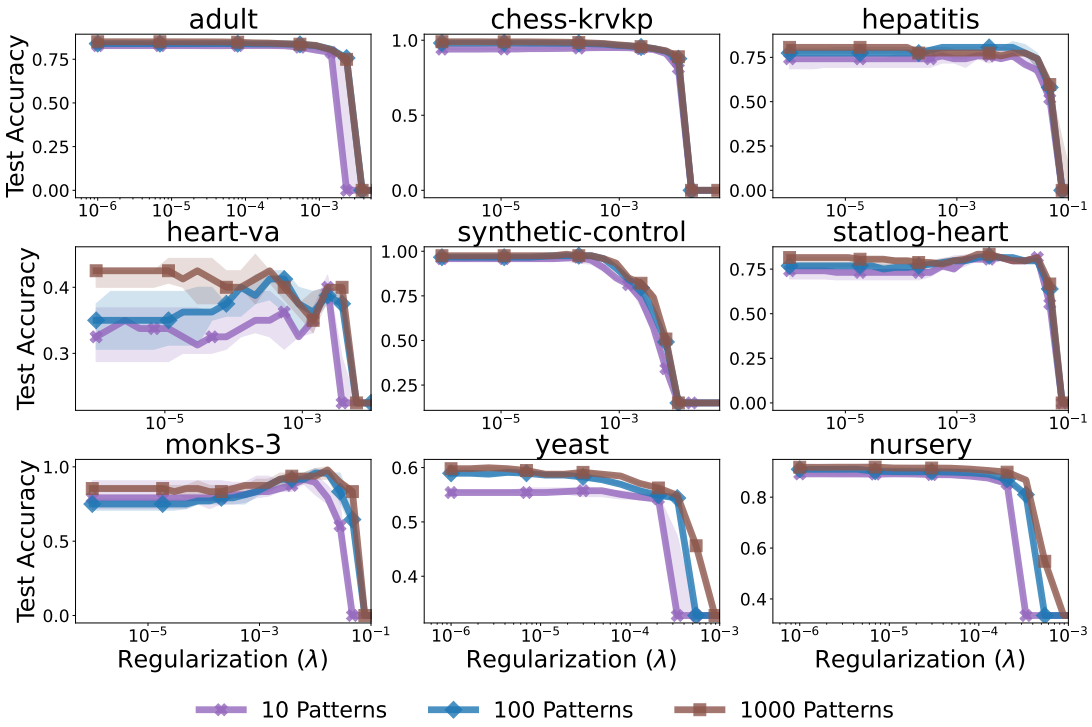

Figure 18. Effect of sampling activation patterns on test accuracy for neural networks trained using the **C-GReLU** problem on nine different UCI datasets. We consider a grid of regularization parameters and plot median (solid line) and first and third quartiles (shaded region) over 10 random samplings of $\tilde{\mathcal{D}}$, where $|\tilde{\mathcal{D}}|$ is limited to 10, 100, or 1000 patterns.

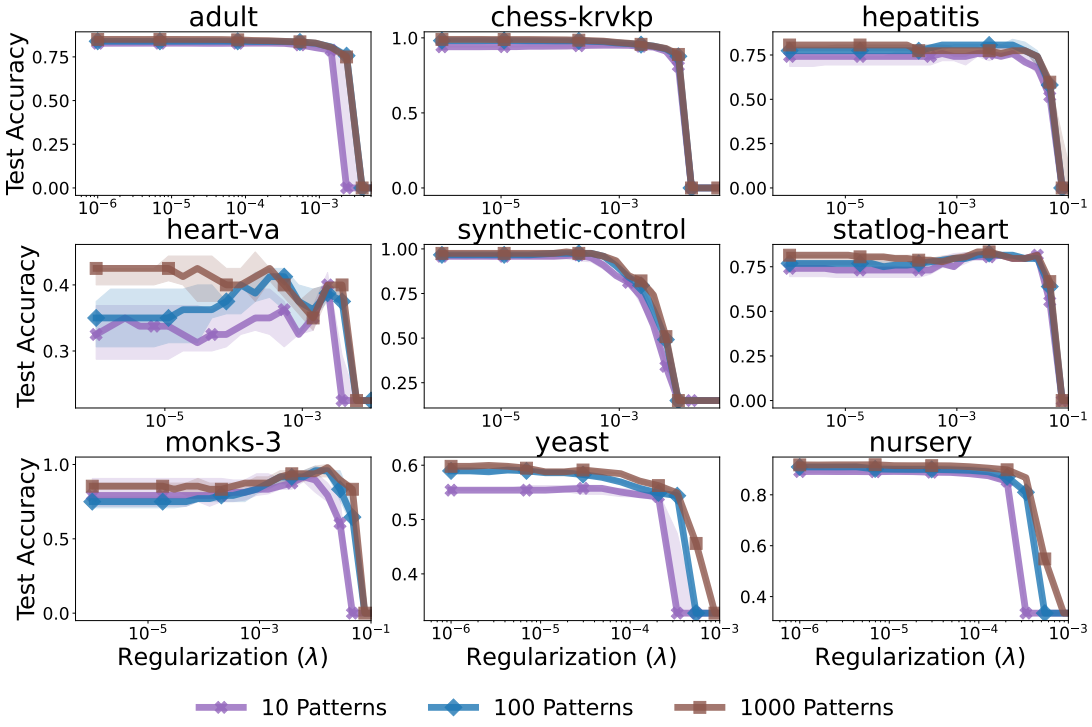

Figure 19. Effect of sampling activation patterns on test accuracy for neural networks trained using the **C-ReLU** problem on nine different UCI datasets. See Figure 18 for details.

Table 6. Additional results comparing our convex solvers against random forests (RF), SVMs with a linear kernel (Linear) and SVMs with an RBF kernel (RBF). We report test accuracies on a further 19 UCI datasets. Combined, C-GReLU and C-ReLU obtain the best test accuracy on 10 datasets. Out of the baselines considered, RBF SVMs are the most competitive with our approach, attaining or tying for best accuracy on 9 datasets.

Dataset	C-GReLU	C-ReLU	RF	SVM	RBF
breast-cancer	73.7	70.2	**75.4**	68.4	68.4
congressional	66.7	65.5	65.5	**67.8**	**67.8**
credit-approval	84.1	84.1	**85.5**	**85.5**	84.8
echocardiogram	80.8	76.9	**88.5**	84.6	84.6
haberman-survival	67.2	**75.4**	70.5	70.5	70.5
hepatitis	80.6	80.6	**83.9**	77.4	77.4
horse-colic	88.3	86.7	**93.3**	90.0	**93.3**
ionosphere	90.0	91.4	91.4	85.7	**97.1**
molec-biol	76.2	**81.0**	76.2	**81.0**	66.7
monks-2	**69.7**	**69.7**	54.5	57.6	**69.7**
monks-3	**95.8**	**95.8**	**95.8**	87.5	91.7
musk-2	99.7	**99.8**	97.3	95.1	99.6
parkinsons	97.4	97.4	84.6	89.7	**100**
pittsburg	**80.0**	75.0	75.0	**80.0**	**80.0**
ringnorm	97.4	83.0	95.1	76.9	**98.2**
spect	46.7	40.0	53.3	**66.7**	**66.7**
statlog-austr.	65.9	**66.7**	62.3	65.2	65.2
statlog-heart	81.5	**85.2**	83.3	81.5	81.5
twonorm	97.6	**97.7**	97.2	97.4	97.4
vertebral-col.	**91.9**	**91.9**	87.1	**91.9**	90.3

and the proportion of examples active in each local model (ie. the number of non-zeros in each D_i matrix). For regularization strength, we optimized over a logarithmic grid with values $\{1 \times 10^{-8}, 3.59 \times 10^{-8}, 1.29 \times 10^{-7}, 4.64 \times 10^{-7}, 1.67 \times 10^{-6}, 5.99 \times 10^{-6}, 2.15 \times 10^{-5}, 7.74 \times 10^{-5}, 2.78 \times 10^{-4}, 1.0 \times 10^{-3}\}$. For the proportion of active examples, we considered (1) setting the bias term for each neuron to enforce 50% of examples to be active or (2) setting the bias to 0 and allowing the proportion to be random.

For the baselines, we used the implementations available from the `scikit-learn` package. We optimized each random forest classifier with respect to the depth of the random trees in the ensemble ($\{2, 4, 10, 25, 50\}$) and over the number of trees in the ensemble ($\{5, 10, 100, 1000\}$). We used the standard soft-margin SVM with and chose regularization parameter from the range $\{1. \times 10^{-5}, 1. \times 10^{-4}, 1. \times 10^{-3}, 1. \times 10^{-2}, 1. \times 10^{-1}, 1\}$ for linear SVMs and $\{1 \times 10^{-5}, 1.78 \times 10^{-4}, 3.16 \times 10^{-3}, 5.62 \times 10^{-2}, 1\}$ for SVMs with an RBF kernel. For RBF SVMs, the RBF bandwidth was optimized over the grid $\{1 \times 10^{-4}, 1.58 \times 10^{-3}, 2.51 \times 10^{-2}, 3.98 \times 10^{-1}, 6.31, 100.0\}$. To obtain final test accuracies, we re-trained each method on the full training set. For our methods, we report the best test accuracy out of five random restarts.

Additional Results: Table 6 reports test results for 19 of the 37 datasets, while Table 2 in the main paper presents results for the remaining 18 datasets. Overall, we find that two-layer neural networks trained using our convex solvers generally perform better than the baseline methods.

E.2. Non-Convex Solvers

This section gives experimental details and additional results for experiments comparing the generalization performance of our convex reformulations to neural networks trained by optimizing the non-convex loss with stochastic gradient methods.

Experimental Details: We selected 20 datasets randomly from our filtered collection of 73 datasets; see Appendix H for details how the 73 datasets were obtained.

We used the default parameters for each of the convex solvers as described in Appendix G. R-FISTA was limited to 2000 iterations, while SGD and Adam were limited to 2000 epochs. For SGD and Adam, considered step-sizes from the following grid: $\{10, 5, 1, 0.5, 0.1, 0.01, 0.001\}$. We used a "step" decrease schedule for the step-sizes, dividing them by 2 every 100 epochs, which we found to work much better than the classical Robbins-Monro schedule (Robbins & Monro, 1951). We considered the following grid of ten regularization parameters: $\{1 \times 10^{-6}, 2.78 \times 10^{-6}, 7.74 \times 10^{-6}, 2.15 \times 10^{-5}, 5.99 \times$

Table 7. Test accuracies for convex and non-convex formulations of the Gated ReLU training problem on a subset of 20 datasets selected from the UCI dataset repository Results are shown as `median (first-quartile/third-quartile)` for each method.

Dataset	C-GReLU	NC-GReLU (Adam)	NC-GReLU (SGD)
magic	86.9 (86.8/87.0)	82.9 (82.9/83.1)	82.1 (82.1/82.2)
statlog-heart	79.6 (79.6/79.6)	85.2 (83.3/85.2)	83.3 (83.3/83.3)
mushroom	100.0 (100/100)	97.6 (97.6/97.9)	96.9 (96.9/96.9)
vertebral-column	87.1 (83.9/87.1)	90.3 (90.3/91.9)	90.3 (90.3/90.3)
cardiotocography	90.1 (89.9/90.4)	85.6 (85.6/85.9)	85.2 (85.2/85.4)
abalone	63.8 (63.7/64.1)	58.7 (58.6/58.7)	58.1 (58.1/58.1)
annealing	90.6 (90.6/91.2)	86.2 (86.2/86.8)	86.2 (85.5/86.2)
car	89.9 (89.9/90.1)	83.8 (83.8/84.1)	83.2 (82.9/83.2)
bank	89.8 (89.7/89.9)	89.9 (89.9/90.0)	89.8 (89.8/90.0)
breast-cancer	68.4 (68.4/68.4)	68.4 (68.4/70.2)	70.2 (70.2/70.2)
page-blocks	96.8 (96.8/96.9)	92.1 (92.0/92.1)	92.4 (92.3/92.4)
contrac	45.9 (45.6/46.3)	53.1 (53.1/53.1)	53.4 (53.1/53.7)
congressional-voting	63.2 (63.2/63.2)	64.4 (64.4/64.4)	66.7 (66.7/66.7)
spambase	93.4 (93.2/93.4)	91.6 (91.6/91.6)	91.2 (91.2/91.3)
synthetic-control	97.5 (97.5/97.5)	98.3 (98.3/98.3)	97.5 (97.5/98.3)
musk-1	93.7 (91.6/93.7)	93.7 (93.7/93.7)	94.7 (92.6/94.7)
ringnorm	69.8 (69.5/69.9)	77.0 (77.0/77.0)	77.2 (77.1/77.2)
ecoli	82.1 (82.1/82.1)	79.1 (79.1/80.6)	4.5 (3.0/43.3)
monks-2	69.7 (66.7/69.7)	66.7 (66.7/66.7)	60.6 (57.6/63.6)
hill-valley	62.0 (59.5/66.1)	57.0 (55.4/57.9)	58.7 (58.7/59.5)

$10^{-5}, 1.67 \times 10^{-4}, 4.64 \times 10^{-4}, 1.29 \times 10^{-3} 3.59 \times 10^{-2}, 1.0 \times 10^{-2}\}$. For each method-dataset pair, we performed five-fold cross validation on the training set and selected the best step-size and regularization parameter according to the cross-validated test accuracy.

For the gated ReLU problems (both C-GReLU and NC-GReLU) we sampled the same set of 5000 activation patterns for both the convex reformulation and the original non-convex model. We used 2500 activation patterns for the C-ReLU problem. To ensure a similar model space, we computed at the number of active neurons (e.g. $v_i \neq 0$ or $w_i \neq 0$) at convergence for C-ReLU and then used this as the number of hidden units for $NC - ReLU$ problems. Note that we extend our convex reformulations to multi-class problems using the results in Appendix A.1. Similarly, we use the vector-output variant of the NC-ReLU problem (Eq. 16) for multi-class problems.

After selecting hyper-parameters, we obtain the final test accuracies by re-training on the full training set and testing on a held-out test set. To control for noise in the sampling of gate vectors in the Gated ReLU problems and C-ReLU, we repeat this final testing procedure five times with different random seeds.

Additional Results: Tables 7 and 8 report median test accuracies as well as first and third quartiles for the convex and non-convex formulations with gated ReLU and ReLU activations, respectively. Note that these results are identical to those the provided in the main paper (Table 3) but for the inclusion of variance/distribution information in the form of quartiles.

F. Image Classification

Experimental Details: The MNIST and CIFAR-10 datasets are high-dimensional, with $(n, d) = (60000, 784)$ and $(50000, 3072)$, respectively. As such, we require a large number of neurons for both problems, for which we use $m = 5000$ and $m = 4000$ neurons respectively. Both datasets are normalized column-wise, and squared loss is used as the objective. Activation patterns are generated by sampling u_i from a distribution that samples a 3×3 patch uniformly from the image, then sampling values for that patch from a standard Gaussian distribution, with all other values set to zero. This technique is used for both convex and non-convex architectures. We use the extensions of the C-GReLU and NC-ReLU to multi-class problems as given in Appendix A.1.

For the NC-GReLU experiments, for all optimizers, we consider a learning rate of $1.0, 0.1, 0.01$. We use a momentum parameter of 0.9 for SGD. To improve convergence, the step size was decayed by a factor of 2 every 200 epochs, and the networks were trained for a maximum of 1000 epochs. We use a batch size of 10% of the training data. For the C-GReLU experiments, no R-FISTA optimizer parameters are tuned–we fix the initial step size to 0.1, with quadratic backtracking

Table 8. Test accuracies for convex and non-convex formulations of the ReLU training problem on a subset of 20 datasets selected from the UCI dataset repository Results are shown as `median (first-quartile/third-quartile)` for each method.

Dataset	C-ReLU	NC-ReLU (Adam)	NC-ReLU (SGD)
magic	85.9 (85.8/85.9)	86.9 (86.9/86.9)	86.4 (86.3/86.4)
statlog-heart	83.3 (81.5/83.3)	83.3 (83.3/83.3)	79.6 (79.6/79.6)
mushroom	100.0 (100/100)	100.0 (100/100)	99.9 (99.9/99.9)
vertebral-column	90.3 (88.7/90.3)	90.3 (90.3/90.3)	88.7 (88.7/88.7)
cardiotocography	89.9 (89.9/89.9)	36.5 (22.8/36.5)	88.9 (88.9/88.9)
abalone	66.2 (66.1/66.3)	65.3 (64.9/65.4)	66.1 (66.1/66.1)
annealing	90.6 (89.9/90.6)	93.7 (93.7/93.7)	88.7 (88.1/88.7)
car	87.8 (87.8/87.8)	94.8 (94.8/94.8)	90.1 (90.1/90.1)
bank	89.8 (89.7/89.9)	90.8 (90.8/90.9)	90.5 (90.5/90.5)
breast-cancer	68.4 (66.7/68.4)	64.9 (64.9/64.9)	68.4 (68.4/68.4)
page-blocks	94.0 (94.0/94.0)	97.1 (97.1/97.1)	96.9 (96.9/96.9)
contrac	55.1 (54.1/55.4)	54.4 (54.1/54.4)	53.7 (53.7/53.7)
congressional-voting	63.2 (63.2/65.5)	62.1 (62.1/62.1)	67.8 (67.8/67.8)
spambase	93.3 (93.2/93.4)	93.5 (93.5/93.5)	93.2 (93.2/93.2)
synthetic-control	98.3 (97.5/98.3)	96.7 (96.7/96.7)	96.7 (96.7/96.7)
musk-1	93.7 (93.7/93.7)	96.8 (96.8/96.8)	95.8 (95.8/95.8)
ringnorm	77.0 (76.8/77.0)	77.3 (77.3/77.4)	77.4 (77.3/77.5)
ecoli	80.6 (80.6/80.6)	82.1 (82.1/82.1)	80.6 (80.6/80.6)
monks-2	69.7 (69.7/72.7)	69.7 (66.7/69.7)	72.7 (72.7/75.8)
hill-valley	65.3 (64.5/65.3)	62.8 (62.8/62.8)	55.4 (55.4/55.4)

with $\beta = 0.8$, and forward-tracking with $\alpha = 1.2$ and $c = 5$. For all methods, we consider regularization parameters $\lambda \in [10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}]$, and choose the one with the best accuracy on the validation set, which is chosen to be a random subset of 20% of the training data. All models are trained with an NVIDIA Titan X GPU with 12GB RAM.

For G-ReLU, a value of $\lambda = 10^{-7}$ was chosen for MNIST and $\lambda = 10^{-6}$ for CIFAR-10. For SGD, values of $(\eta, \lambda) = (1.0, 10^{-7})$ were chosen for MNIST and $(\eta, \lambda) = (1.0, 10^{-5})$ for CIFAR-10. For Adam, values of $(\eta, \lambda) = (0.01, 10^{-6})$ were chosen for MNIST and $(\eta, \lambda) = (0.01, 10^{-4})$ for CIFAR-10. For Adagrad, values of $(\eta, \lambda) = (0.01, 10^{-7})$ were chosen for MNIST and $(\eta, \lambda) = (0.01, 10^{-5})$ for CIFAR-10.

G. Default Optimization Parameters

In this section, we report the standard parameter settings for our optimizers. We use these parameters in all experiments unless explicitly stated otherwise. Note that data normalization (Appendix C.1) is applied in all experiments for both the convex and non-convex training problems.

G.1. R-FISTA

We set the backtracking parameter to $\beta = 0.8$ and the forward-tracking parameter to $\alpha = 1.25$. For the step-size initialization strategy, we set the threshold to be $c = 5.0$. We set the first step-size to be $\eta_0 = 1.0$. The restart strategy detailed in Section 4.1.1 is always used unless it is explicitly stated otherwise. Finally, we consider the optimizer to have (approximately) converged when the minimum-norm subgradient has ℓ_2-norm less than or equal to 10^{-3}. In practice we check the equivalent condition on the squared gradient norm with the threshold 10^{-6}. We always initialize the model weights as $v_i = 0$ for each $D_i \in \tilde{\mathcal{D}}$.

G.2. AL Method

We set the initial penalty parameter to be $\delta = 100$ and use the windowing heuristic with $r_u = 10^{-2}$ and $r_l = 10^{-3}$. The dual parameters are initialization at 0, as are the primal parameters. Note that we always warm-start the optimization of the augmented Lagrangian at the solution to the previous iteration's optimization problem. The convergence tolerance when checking for satisfaction of the windowing heuristic is set to be tol $= 10^{-3/2}$. If the constraint gap is larger than r_u, we increase δ as $\delta \leftarrow 2 * \delta$ and repeat the procedure. If $c_{\text{gap}} < r_l$, we set $\delta \leftarrow \delta/2$ and also change the convergence to be tol $\leftarrow$ tol$/2$.

The convergence tolerance for minimization of the augmented Lagrangian once the window heuristic is satisfied is tol $= 10^{-3}$. We consider the AL method to have approximately converged when $c_{\text{gap}} \leq 10^{-3}$ and the minimum norm subgradient of the augmented Lagrangian (with respect to the primal parameters) is also less than 10^{-3}.

To solve Equation (11), we use R-FISTA with the standard configuration as outlined in the previous section. We enforce a maximum of 1000 iterations for the sub-solver, meaning that we execute a step of the AL method after at most 1000 iterations of R-FISTA regardless of the termination tolerances. In general, we permit as many "outer" iterations of the AL method as needed since these do not require gradient computations, but limit the overall optimization procedure to 10000 iterations of R-FISTA.

H. UCI Datasets

We use the binary and multi-class classification datasets from the UCI machine learning repository (Dua & Graff, 2017) as pre-processed by Delgado et al. (2014). Note that we do not use the same training/validation/test procedure as Delgado et al. (2014), since this is known to have test-set leakage. We applied the following selection rules to decide which datasets to retain for our experiments:

- at least 150 examples and 5 features;
- no more than 50000 examples and 10 classes;
- no duplicated datasets with different targets or features.

This left the following 73 datasets from the original collection of 121: `abalone`, `adult`, `annealing`, `bank`, `breast-cancer`, `breast-cancer-wisc-diag`, `car`, `cardiotocography-3clases`, `chess-krvkp`, `congressional-voting`, `conn-bench-sonar-mines-rocks`, `contrac`, `credit-approval`, `cylinder-bands`, `dermatology`, `ecoli`, `energy-y1`, `flags`, `glass`, `heart-cleveland`, `heart-hungarian`, `heart-va`, `hepatitis`, `hill-valley`, `horse-colic`, `ilpd-indian-liver`, `image-segmentation`, `ionosphere`, `led-display`, `low-res-spect`, `magic`, `mammographic`, `molec-biol-splice`, `monks-2`, `monks-3`, `mushroom`, `musk-1`, `musk-2`, `nursery`, `oocytes_merluccius_nucleus_4d`, `oocytes_trisopterus_nucleus_2f`, `optical`, `ozone`, `page-blocks`, `parkinsons`, `pendigits`, `pima`, `planning`, `primary-tumor`, `ringnorm`, `seeds`, `semeion`, `spambase`, `statlog-australian-credit`, `statlog-german-credit`, `statlog-heart`, `statlog-image`, `statlog-landsat`, `statlog-vehicle`, `steel-plates`, `synthetic-control`, `teaching`, `thyroid`, `tic-tac-toe`, `twonorm`, `vertebral-column-2clases`, `wall-following`, `waveform`, `waveform-noise`, `wine`, `wine-quality-red`, `wine-quality-white`, `yeast`

Optimization Performance: For our experiments evaluating optimization performance, we considered all 73 datasets and generated $6 * 73 = 438$ optimization problems by considering the following grid of regularization parameters:

$$\lambda \in \left\{1 \times 10^{-5}, 6.31 \times 10^{-5}, 3.98 \times 10^{-4}, 2.51 \times 10^{-3}, 1.58 \times 10^{-2}, 1.0 \times 10^{-1}\right\}$$

We did a single train/test split for each dataset and report optimization metrics on the training set only. The test was used for heuristic "sanity checks" of the final models.

Model Performance: For our experiments evaluating generalization or test performance of different models, we randomly selected a subset of the filtered UCI datasets. We report the regularization parameters considered for each experiment in the appropriate section.