An efficient nonconvex reformulation of stagewise convex optimization problems

Rudy Bunel[*]
DeepMind
rbunel@google.com

Oliver Hinder[*]
Google Research, University of Pittsburgh
ohinder@pitt.edu

Srinadh Bhojanapalli
Google Research
bsrinadh@google.com

Krishnamurthy (Dj) Dvijotham
DeepMind
dvij@google.com

Abstract

Convex optimization problems with staged structure appear in several contexts, including optimal control, verification of deep neural networks, and isotonic regression. Off-the-shelf solvers can solve these problems but may scale poorly. We develop a nonconvex reformulation designed to exploit this staged structure. Our reformulation has only simple bound constraints, enabling solution via projected gradient methods and their accelerated variants. The method automatically generates a sequence of primal and dual feasible solutions to the original convex problem, making optimality certification easy. We establish theoretical properties of the nonconvex formulation, showing that it is (almost) free of spurious local minima and has the same global optimum as the convex problem. We modify PGD to avoid spurious local minimizers so it always converges to the global minimizer. For neural network verification, our approach obtains small duality gaps in only a few gradient steps. Consequently, it can quickly solve large-scale verification problems faster than both off-the-shelf and specialized solvers.

1 Introduction

This paper studies efficient algorithms for a particular class of stage-wise optimization problems:

$$\underset{(s,z)\in S\times\mathbf{R}^n}{\text{minimize}}\ f(s,z) \tag{1a}$$

$$\text{s.t. } \mu_i(s, z_{1:i-1}) \le z_i \le \eta_i(s, z_{1:i-1}) \qquad \forall i \in \{1,\dots,n\} \tag{1b}$$

where n and m are positive integers, $S \subseteq \mathbf{R}^m$, the function f has domain $S \times \mathbf{R}^n$ and range $\mathbf{R}$, the functions μ_i and η_i have domain $S \times \mathbf{R}^{i-1}$ and range $\mathbf{R}$. Given a vector z, we use the notation $z_{1:i}$ to denote the vector $[z_1, \dots, z_i]$. We let $z_{1:0}$ be a vector of length zero. Throughout the paper we assume that $\eta_1, \dots, \eta_n$ are proper concave functions, f, $\mu_1, \dots, \mu_n$ are proper convex functions, and S is a nonempty convex set.

Problems that fall into this problem class are ubiquitous. They appear in optimal control [1], finite horizon Markov decision processes with cost function controlled by an adversary [2], generalized Isotonic regression [3, 4], and verification of neural networks [5–7]. Details explaining how these problems can be written in the form of (1) are given in Appendix A. Here we briefly outline how neural network verification falls into (1b). Letting s represent the input image and z the activation values, neural networks verification can be written (unconventionally) as

$$\underset{(s,z)\in S\times\mathbf{R}^n}{\text{minimize}}\ f(s,z) \text{ s.t. } z_i = \sigma([s, z_{1:i-1}] \cdot w_i),$$

[*] Equal contribution

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

for (sparse) weight vectors w_i and activation function σ. A convex relaxation is created by picking functions satisfying $\mu_i(s, z_{1:i-1}) \leq \sigma_i([s, z_{1:i-1}] \cdot w_i) \leq \eta_i(s, z_{1:i-1})$ for all s and z feasible to the original problem. Solving these convex relaxations with traditional methods can be time consuming. For example, Salman et al. [8] reports spending 22 CPU years to solve problems of this type in order to evaluate the tightness of their proposed relaxation. Consequently, methods for solving these relaxations faster are valuable.

1.1 Related work

1.1.1 Drawbacks of standard solvers for stagewise convex problems

Standard techniques for solving (1) can be split into two types: first-order methods and second-order methods. These techniques do not exploit this stage-wise structure, and so they face limitations.

First-order methods: Methods such as mirror prox [9], primal-dual hybrid gradient (PDHG) [10], augmented lagrangian methods [11], and subgradient methods [12] have cheap iterations (i.e., a matrix-vector multiply) but may require many iterations to converge. For example,

$$\underset{x}{\text{minimize}} - x_n \quad \text{s.t.} \quad x_1 \in [0, 1], \quad -1 \leq x_i \leq x_{i-1} \quad \forall i \in \{1, \dots, n-1\} \tag{2}$$

is an instance of (1) with optimal solution at $x = \mathbf{1}$. However, this is the type of problem that exhibits the worst-case performance of a first-order method. In particular, one can show (see Appendix B) using the techniques of Nesterov [13, Section 2.1.2] it will take at least $n - 1$ iterations until methods such as PDHG or mirror-prox obtain an iterate with $x_1 > 0$ starting from $x = \mathbf{0}$. Furthermore, existing first-order methods are unable to generate a sequence of primal feasible solutions. This makes constructing duality gaps challenging. We could eliminate these constraints using a projection operator, but in general this will require calling a second-order method at each iteration, making iterations more expensive.

Second-order methods: Methods such as interior point and simplex methods rely on factorizing a linear system, and can suffer from speed and memory problems on large-scale problems if the sparsity pattern is not amenable to factorization. This issue, for example, occurs in the verification of neural networks as dense layers force dense factorizations.

1.1.2 Other nonconvex reformulations of convex problems

Most research on nonconvex reformulations of convex problems is for semi-definite programs [14–16]. In this work, the semi-definite variable is rewritten as the sum of low rank terms, forgoing convexity but avoiding storing the full semi-definite variable. Compared with this line of research our technique is unique for several reasons. Firstly, our primary motivation is speed of convergence and obtaining certificates of optimality, rather than reducing memory or iteration cost. Secondly, the landscape of our nonconvex reformulation is different. For example, it contains spurious local minimizers (as opposed to saddle points) which we avoid via careful algorithm design.

2 A nonconvex reformulation of stagewise convex problems

We now present the main technical contribution of this paper, i.e., a nonconvex reformulation of the stagewise convex problems of the form (1) and an analysis of efficient projected gradient algorithms applied to this formulation.

2.1 Assumptions

We begin by specifying assumptions we make on the objective and constraint functions in (1). Prior to doing so, it will be useful to introduce the notion of a smooth function:

Definition 1. *A function $h : X \to \mathbf{R}$ is smooth if $\nabla h(x)$ exists and is continuous for all $x \in X$; h is L-smooth if $\|\nabla h(x) - \nabla h(x')\|_2 \leq L \|x - x'\|_2, \forall x, x' \in X$.*

Assumption 1. *Assume $f, \eta_1, \dots, \eta_n, \mu_1, \dots, \mu_n$ are smooth functions.*

Remark 1. *If Assumption 1 fails to hold it is may be possible to approximate f, η_i and μ_i by smooth functions [17]. It is also possible one could use a nonsmooth optimization method [18]. However, we leave the study of these approaches to future work.*

Let Π_S denote the projection operator onto the set S. Ideally, the cost of this projection is cheap (e.g., S is formed by simple bound constraints) as we will be running projected gradient descent (PGD) and therefore routinely using projections.

Assumption 2. *Assume S is a bounded set with diameter $D_s = \sup_{s,\hat{s}\in S} \|s - \hat{s}\|_2$. Further assume Z is a bounded set such that for every feasible solution (s, z) to (1) we have $z \in Z$. Define $D_z = \sup_{z,\hat{z}\in Z} \|\hat{z} - z\|_2$.*

We remark that if η and μ are smooth, and S is bounded then there exists a set Z satisfying Assumption 2. The primary reason for Assumption 2 is it will allow us to form lower bounds on the optimal solution to (1). We will also find it useful to be able to readily construct upper bounds, i.e., feasible solutions to (1). This is captured by the following assumption.

Assumption 3. *For all $i \in \{1, \dots, n\}$, if $s \in S$ and $\mu_j(s, z_{1:j-1}) \leq z_j \leq \eta_j(s, z_{1:j-1})$ for $j \in \{1, \dots, i-1\}$ then $\mu_i(s, z_{1:i-1}) \leq \eta_i(s, z_{1:i-1})$.*

Assumption 3 is equivalent to stating that feasible solutions to (1) can be constructed inductively. In particular, given we have a feasible solution to the first $1, \dots, i-1$ constraints we can find a feasible solution for the ith constraint by picking any $z_i \in [\mu_i(s, z_{1:i-1}), \eta_i(s, z_{1:i-1})]$ which must be a nonempty set by Assumption 3. All examples discussed in Appendix A satisfy Assumption 3.

2.2 A nonconvex reformulation

Our idea is to apply PGD to the following nonconvex reformulation of (1),

$$\underset{(s,z,\theta)\in S\times\mathbf{R}^n\times[0,1]^n}{\text{minimize}} \; f(s, z) \tag{3a}$$

$$\text{s.t. } z_i = (1 - \theta_i)\mu_i(s, z_{1:i-1}) + \theta_i\eta_i(s, z_{1:i-1}) \qquad \forall i \in \{1, \dots, n\}. \tag{3b}$$

The basis of this reformulation is that if $\mu_i(s, z_{1:i-1}) \leq z_i \leq \eta_i(s, z_{1:i-1})$ then z_i is a convex combination of $\mu_i(s, z_{1:i-1})$ and $\eta_i(s, z_{1:i-1})$. This reformulation allows us to replace the z variables with θ variables and replaces the constraints (1b) that are difficult to project onto with box constraints. For conciseness we denote (3b) by

$$z \leftarrow \textsc{Forward}(s, \theta).$$

Let us consider an alternative interpretation of (3) that explicitly replaces z with θ. Define $\psi_n(s, z) := f(s, z)$ and recursively define ψ_i for all $i \in \{1, \dots, n\}$ by

$$\psi_{i-1}(s, z_{1:i-1}, \theta_{i:n}) := \psi_i(s, z_{1:i-1}, (1 - \theta_i)\mu_i(s, z_{1:i-1}) + \theta_i\eta_i(s, z_{1:i-1}), \theta_{i+1:n}).$$

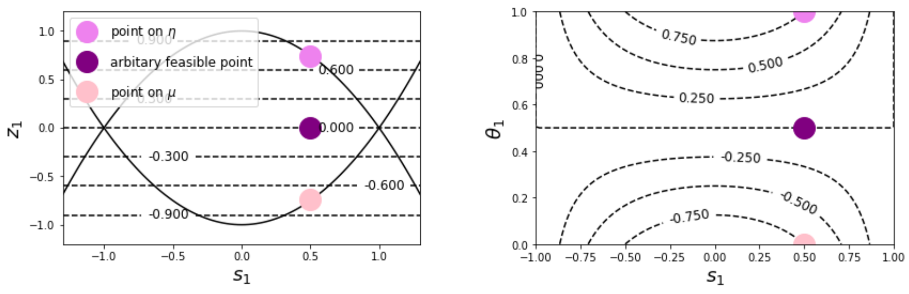
Note that ψ_{i-1} eliminates the variable z_i from ψ_i by replacing it with $(1 - \theta_i)\mu_i(s, z_{1:i-1}) + \theta_i\eta_i(s, z_{1:i-1})$. Using this notation, the reformulation (3) is equivalent to:

$$\underset{(s,\theta)\in S\times[0,1]^n}{\text{minimize}} \; \psi_0(s, \theta). \tag{4}$$

For intuition consider the following example

$$S := [-1, 1], \quad f(s_1, z_1) := z_1, \quad \eta_1(s_1) := 1 - s_1^2, \quad \mu_1(s_1) := s_1^2 - 1. \tag{5}$$

In Figure 1 we plot this example. Consider an arbitrary feasible point, e.g., $z_1 = 0.0$, $s_1 = 0.5$ and note that point can be written as a convex combination of a point on η and a point on μ. The nonconvex reformulation does this explicitly with box constraints replacing nonlinear constraints.



Plot of original convex problem   Plot of nonconvex reformulation $\psi_0(s_1, \theta_1)$

Figure 1: Comparison between original problem and reformulation.

The function ψ_0 is smooth (since it is the composition of smooth functions), and its gradient is computable by backpropagation, i.e., $\boldsymbol{\nabla}\psi_n = \boldsymbol{\nabla} f$ and for $i = n, \dots, 1$,

$$\boldsymbol{\nabla}_s \psi_{i-1} = \boldsymbol{\nabla}_s \psi_i + \frac{\partial \psi_i}{\partial z_i} \left(\theta_i \boldsymbol{\nabla}_s \eta_i + (1 - \theta_i) \boldsymbol{\nabla}_s \mu_i \right) \tag{6a}$$

$$\frac{\partial \psi_{i-1}}{\partial z_j} = \frac{\partial \psi_i}{\partial z_j} + \frac{\partial \psi_i}{\partial z_i} \left(\theta_i \frac{\partial \eta_i}{\partial z_j} + (1 - \theta_i) \frac{\partial \mu_i}{\partial z_j} \right) \quad \forall j \in \{1, \dots, i-1\} \tag{6b}$$

$$\frac{\partial \psi_0}{\partial \theta_i} = \frac{\partial \psi_i}{\partial \theta_i} = \frac{\partial \psi_i}{\partial z_i} \frac{\partial z_i}{\partial \theta_i} = \frac{\partial \psi_i}{\partial z_i} (\eta_i - \mu_i) \tag{6c}$$

where we denote $f = f(s, z)$, $\psi_i = \psi_i(s, z_{1:i-1}, \theta_{i:n})$, $\eta_i = \eta_i(s, z_{1:i-1})$, and $\mu_i = \mu_i(s, z_{1:i-1})$; this abuse of notation, where we assume these functions are evaluated at (s, z, θ) unless specified otherwise, will continue throughout the paper for the purposes of brevity. The subscript on $\boldsymbol{\nabla}$ specifies the arguments the derivative is with respect to, if it is left blank then we take the derivatives with respect to all arguments. Therefore, one can apply PGD, or other related descent algorithm to minimize ψ_0. Moreover, the cost of computing the gradient via backpropagation is cheap (dominated by the cost of evaluating $\boldsymbol{\nabla} f$, $\boldsymbol{\nabla}\eta$, and $\boldsymbol{\nabla}\mu$). However, since ψ_0 is nonconvex, it is unclear whether a gradient based approach will find the global optimum.

We show that this is indeed the case in the following subsections: In section 2.3, we show that global minima are preserved under the nonconvex reformulation. In section 2.4, show that *nondegenerate* local optima are global optima and that projected gradient descent converges quickly to these. In section 2.5, we show how to modify projected gradient descent to avoid convergence to degenerate local optima and ensure convergence to a global optimum.

2.3 Nonconvex reformulation is equivalent to original convex problem

Before arguing that the local minimizers of (3) are equal to the global minimizers of (1), it is important to confirm that the global minimizers are equivalent. Indeed, Theorem 1 confirms this.

Theorem 1. *Any feasible solution to* (1) *corresponds to a feasible solution for* (3) *with the same objective value. Furthermore, if $\mu_i \leq \eta_i$ for all $i \in \{1, \dots, n\}$ and (s, z) feasible to* (3)*, then any feasible solution to* (3) *corresponds to a feasible solution for* (1) *with the same objective value. In which case, the global optimum of* (3) *is same as the global optimum of* (1)*.*

Proof. Consider any feasible solution (s, z) to (1). By setting $\theta_i = \frac{z_i - \mu_i}{\eta_i - \mu_i}$ (any $\theta_i \in [0, 1]$ suffices if $\mu_i = \eta_i$) we obtain a feasible solution to (3). On the other hand, if $\mu_i \leq \eta_i$ then (3b) and $\theta_i \in [0, 1]$ implies $\mu_i \leq z_i \leq \eta_i$. □

A sufficient condition for the premise of Theorem 1 to hold is Assumption 3. As Figure 2 shows, if Assumption 3 fails then the nonconvex reformulation can generate infeasible solutions to the original convex optimization problem (1b). Consider the example given by (5) except with $S := [-1.5, 1.5]$ instead of $S := [-1, 1]$. The set of feasible solutions to (1) is enclosed by the two curves. At $s_1 = 1.2$ and $\theta = 1$, $\mu(s_1) > \eta(s_1)$, which is infeasible.

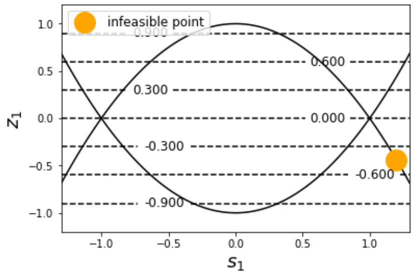


Figure 2: Infeasible: Assumption 3 fails.

2.4 Analysis of nondegenerate local optima

This section is devoted to proving that under a nondegeneracy assumption, the first-order stationary points of (3) are global minimizers. Degeneracy issues arise when $\eta_i = \mu_i$. In this situation, if θ_i changes, then z will remain the same, and therefore from the perspective of the convex formulation, the solution is the same. However, from the perspective of the function ψ_0 there is an important difference. In particular, as θ_i changes the gradient of ψ_0 changes. Consequently, certain values of θ_i may generate spurious local minimizers. Recall example (5), i.e., $S := [-1, 1]$, $f(s_1, z_1) := z_1$, $\eta_1(s_1) := 1 - s_1^2$ and $\mu_1(s_1) := s_1^2 - 1$. In this instance,

$$\psi_0 = \theta_1(1 - s_1^2) + (1 - \theta_1)(s_1^2 - 1) = (1 - 2\theta_1)(s_1^2 - 1), \quad \frac{\partial \psi_0}{\partial s_1} = (1 - 2\theta_1)(2s_1 - 1).$$

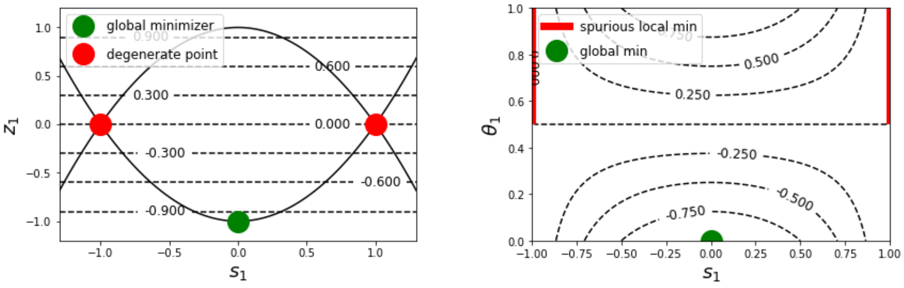
As illustrated in Figure 3, the global minimizer is $s_1 = 0, \theta = 0 \Rightarrow z_1 = -1$. If $s_1 \pm 1$ then for all $\theta_1 \in [0, 1]$ we have $z_1 = 0$. Moreover, the points $s_1 \pm 1$, $\theta_1 \in (0.5, 1]$ are spurious local minimizers.

To see this, note for all $\theta_1 \in [0.5, 1]$, and $s_1 \in S$ that $\psi_0(s_1, \theta_1) \geq 0 = \psi_0(1, \theta_1)$. In contrast, the points $s_1 \pm 1$, $\theta \in [0, 0.5)$ are *not* local minimizers, because for $s_1 \pm 1$ and $\theta_1 \in [0, 0.5)$ we have $\frac{\partial \psi_0}{\partial s_1} > 0$ implying that gradient descent steps move away from the boundary. We conclude that if $\mu_i = \eta_i$ *certain* values of θ_i could be spurious local minimizers. We emphasize the word *certain* because, as Section 2.5 details, there is always a value of θ_i that enables escape.

The nondegeneracy assumption we make is that for some $\gamma \geq 0$ the set

$$\mathcal{K}_\gamma(s, \theta) := \Bigg\{ i \in \{1, \dots, n\} : \quad z = \text{Forward}(s, \theta), \\ \eta_i - \mu_i \leq \gamma, \quad \theta_i \left(\frac{\partial \psi_i}{\partial z_i} \right)^+ + (1 - \theta_i) \left(\frac{\partial \psi_i}{\partial z_i} \right)^- > 0 \Bigg\}$$

is empty, where $(\cdot)^+ := \max\{\cdot, 0\}$ and $(\cdot)^- := \min\{\cdot, 0\}$. If the set $\mathcal{K}_0(s, \theta)$ is non-empty then any coordinate $i \in \mathcal{K}_0(s, \theta)$ could be causing a spurious local minimizer. Values of γ strictly greater than zero ensures that we do not get arbitrarily close to a degenerate point. We will show this nondegeneracy assumption guarantees that first-order stationary points are global minimizers.



Plot of original convex problem Plot of nonconvex reformulation $\psi_0(s_1, \theta_1)$

Figure 3: Example of degeneracy causing spurious local minimizers when $s_1 \pm 1$.

While our nondegeneracy assumption holds it will suffice to run PGD which is defined as

$$(s^+, \theta^+) \leftarrow (s, \theta) + \underset{d \in \mathcal{D}(s, \theta)}{\operatorname{argmin}} \, \nabla \psi_0 \cdot d + \frac{L}{2} \|d\|_2^2,$$

where $\mathcal{D}(s, \theta) := \{d : (s, \theta) + d \in S \times [0, 1]^n\}$ is the set of feasible search directions and L is the smoothness of ψ_0 (see Definition 1). A useful fact is that PGD satisfies $\psi_0(s^+, \theta^+) \leq \psi_0(s, \theta) - \delta_L(s, \theta)$ for

$$\delta_L(s, \theta) := -\underset{d \in \mathcal{D}(s, \theta)}{\text{minimize}} \, \nabla \psi_0 \cdot d + \frac{L}{2} \|d\|_2^2.$$

See [19, Lemma 2.3.] for a proof. In other words, $\delta_L(s, \theta)$ represents the minimum progress of PGD. Once again for brevity we will denote $\delta_L(s, \theta)$ by δ_L. Note that if δ_L is zero then we are at a first-order stationary point of ψ_0. The remainder of this section focuses on proving that δ_L provides an upper bound on the optimality gap. To form this bound we use Lagrangian duality. In particular, the Lagrangian of (1) is:

$$\mathcal{L}(s, z, y) := f + \sum_{i=1}^n (y_i^+ \mu_i - y_i^- \eta_i - y_i z_i)$$

where $y_i^+ = \max\{y_i, 0\}$, and $y_i^- = \max\{-y_i, 0\}$. We will denote $\mathcal{L}(s, z, y)$ by $\mathcal{L}$. Define,

$$\Delta(s, \theta) := \sum_{i=1}^n (y_i z_i - y_i^+ \mu_i + y_i^- \eta_i) + \sup_{(\hat{s}, \hat{z}) \in S \times Z} \nabla_{s,z} \mathcal{L} \cdot (\hat{s} - s, \hat{z} - z) \tag{7}$$

with $z = \textsc{Forward}(s, \theta)$ and $y_i = \frac{\partial \psi_i}{\partial z_i}$. If (s, z) is feasible to (1) we conclude $\Delta(s, \theta)$ is a valid duality gap, i.e., provides global guarantees, because by duality, convexity and (7),

$$f_* \geq \inf_{(\hat{s}, \hat{z}) \in S \times Z} \mathcal{L}(\hat{s}, \hat{z}, y) \geq \mathcal{L} + \inf_{(\hat{s}, \hat{z}) \in S \times Z} \nabla_{s,z} \mathcal{L} \cdot (\hat{s} - s, \hat{z} - z) = f - \Delta(s, \theta). \tag{8}$$

To compute $\Delta(s,\theta)$, one needs to be able to efficiently minimize a linear function over the set Z. For this reason, one should choose Z to have a simple form (i.e., bound constraints).

Assumption 4. *There exists a constant $c > 0$ such that $\|\eta - \mu\|_2 + D_s\|\nabla_s\eta - \nabla_s\mu\|_2 + D_z\|\nabla_z\eta - \nabla_z\mu\|_2 \le c$ for all (s,z) that are feasible to* (1b).

In Assumption 4, observe that $\nabla_s\eta - \nabla_s\mu$ and $\nabla_z\eta - \nabla_z\mu$ are matrices so $\|\cdot\|_2$ is the spectral norm. Also, note that Assumption 1 and 2 imply that Assumption 4 must hold. However, we add Assumption 4 because it makes the constant c explicit.

Lemma 1 (Nondegenerate first-order stationary points are optimal)**.** *Suppose Assumption 2 and 4 hold. Suppose also that $\mathcal{K}_\gamma(s,\theta) = \emptyset$ with $\gamma \in (0,\infty)$, and that $\delta_L \le L/2$. Then $\Delta(s,\theta)^2 \le L\left(D_s\sqrt{2} + 2\gamma^{-1}c\right)^2\delta_L$.*

In the nondegenerate case (i.e., $\mathcal{K}_\gamma(s,\theta) = \emptyset$), δ_L upper bounds $\Delta(s,\theta)$. In particular, as Lemma 1 demonstrates small progress by gradient steps implies small duality gaps. The proof of Lemma 1 appears in Section C.1 and is technical. The core part of the proof of Lemma 1 is bounding $\theta_i y_i^+ + (1-\theta_i)y_i^-$ for $y_i = \frac{\partial\psi_i}{\partial z_i}$ in terms of γ^{-1} and δ_L. When $\theta_i y_i^+ + (1-\theta_i)y_i^- \approx 0$ one can show that $\Delta(s,\theta) \approx \sup_{\hat{s}\in S}\nabla_s\mathcal{L}\cdot(\hat{s}-s) \approx \sup_{\hat{s}\in S}\nabla_s\psi\cdot(\hat{s}-s) \le D_s\sqrt{2L\delta_L}$.

2.4.1 Analysis of projected gradient descent

Lemma 1 provides the tool we need to prove the convergence of PGD in the nondegenerate case. The algorithm we analyze (Algorithm 1) includes termination checks for optimality. Furthermore, the PGD steps can be replaced by any algorithm that makes at least as much function value reduction as PGD would make in the worst-case. For example, gradient descent with a backtracking line search and an Armijo rule [20, Chapter 3], or a safeguarded accelerated scheme [21] would suffice.

Algorithm 1 Local search algorithm for minimizing ψ_0 in the nondegenerate case.

```
function SIMPLE-PSI-MINIMIZATION($s^1, \theta^1, \epsilon$)
    Suppose $\psi_0$ is $L$-smooth. Note $L \in (0,\infty)$ need not be known.
    for $k = 1, \dots, \infty$ do
        Termination checks:
        if $\Delta(s^k, \theta^k) \le \epsilon$ then
            Found an $\epsilon$-optimal solution:
            return $(s^k, \theta^k)$
        end if
        Reduce the function at least as much as PGD would:
        $(s^{k+1}, \theta^{k+1}) \in \{(s,\theta) : \psi_0(s,\theta) \le \psi_0(s^k,\theta^k) - \delta_L(s^k,\theta^k)\}$
    end for
end function
```

Theorem 2 (PGD converges to global minimizer under nondegeneracy assumption)**.** *Suppose Assumption 2, 3 and 4 hold. Suppose ψ_0 is L-smooth, $\epsilon, \gamma \in (0,\infty)$, $(s^1,\theta^1) \in S \times [0,1]^n$, and $\mathcal{K}_\gamma(s^k,\theta^k) = \emptyset$ for all iterates of the algorithm* SIMPLE-PSI-MINIMIZATION*(s^1,θ^1,ϵ). Then, the algorithm terminates after at most*

$$1 + \frac{2\Delta(s^1,\theta^1)}{L} + \frac{L\left(D_s\sqrt{2} + 2c\gamma^{-1}\right)^2}{\epsilon} \quad \textit{iterations.}$$

See Section C.2 for a proof of Theorem 2. The proof of Theorem 2 directly utilizes Lemma 1 using standard techniques, almost identical to the proof of convergence for gradient descent in the convex setting [13, Theorem 2.1.13].

Remark 2. *It is worth discussing the premise in Theorem 2 that ψ_0 is L-smooth. The composition of smooth functions is smooth, implying ψ_0 is smooth. Moreover, since $S \times [0,1]^n$ is a bounded set we deduce that ψ_0 is L-smooth for some $L > 0$. Therefore the premise that ψ_0 is L-smooth is valid. However, the value of L could be extremely large, for example, if $\eta_i(s, z_{1:i-1}) = \mu_i(s, z_{1:i-1}) = 2z_{i-1}$ for $i > 1$, $\eta_1(s) = \mu_1(s) = s_1$, and $f(s,z) = \frac{1}{2}z_n^2$ then $\psi_0(s,\theta) = \frac{1}{2}(2^n s)^2$ and $L = 4^n$. Note this occurs despite the fact that each component function is well-behaved (i.e., η_i, μ_i, f are* 1*-smooth and* 2*-Lipschitz with respect to the Euclidean norm).*

Remark 3. *Consider* (2)*, the hard example for standard first-order methods. Note that starting from the origin (i.e., $x_1 = 0$, $\theta = \mathbf{0}$), then for sufficiently large step size PGD on ψ_0 will take exactly one iteration to find the optimal solution ($x_1 = 1, \theta = \mathbf{1}$).*

Remark 4. *Suppose that we are solving a neural network verification problem (Section 3 and A.2). Then this approach is strongly related to adversarial attack heuristics. In particular, freezing $\theta = \mathbf{0}$ in* Simple-Psi-Minimization *yields a typical gradient based attack on the network [22].*

2.5 Analysis of degenerate local optima

Section 2.4 proved convergence of PGD to the global minimizer under a nondegeneracy assumption (i.e., $\mathcal{K}_\gamma(s^k, \theta^k) = \emptyset$). This section develops a variant of PGD that requires no degeneracy assumptions but still converges to the global minimizer.

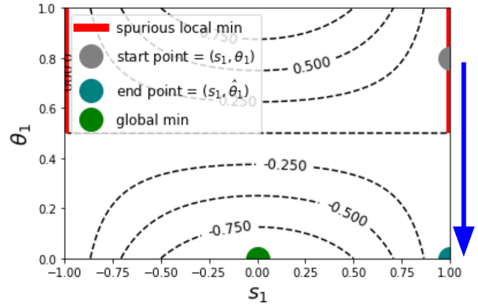
2.5.1 Escaping exact local minimizers

Our main result, presented in Section 2.5.2, proves convergence under minimal assumptions. The key to the result is developing an algorithm for escaping basins of local minimizers. However, the algorithm and analysis is very technical. To give intuition for it this section considers the easier case of escaping *exact* local minimizers (Lemma 2).

The high level idea is illustrated in Figure 4b. Recall from Figure 3 that if we are at a spurious local minimizer then the set $\mathcal{K}_\gamma(s, \theta)$ must be nonempty. In particular, in this instance the set $\mathcal{K}_0(s, \theta) = \{1\}$ is nonempty. In this setting, θ_1 corresponds to an edge that we can move along where $\psi_0(s, \theta)$ is constant. Escape-Exact-Local-Min(s, θ) moves us along this edge from (s, θ) to $(s, \hat{\theta})$ at which $\mathcal{K}_0(s, \hat{\theta})$ is empty and therefore we have escaped the local minimizer.

```
function Escape-Exact-Local-Min$(s, \theta)$
    $z = $ Forward$(s, \theta)$, $\hat{\theta} \leftarrow \text{copy}(\theta)$
    for $i = n, \dots, 1$ do
        if $i \in \mathcal{K}_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n})$ then
            $\hat{\theta}_i = \begin{cases} 0 & \frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i} > 0 \\ 1 & \frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i} < 0 \end{cases}$
        end if
    end for
    return $(s, \hat{\theta})$
end function
```

(a) Algorithm



(b) The high level idea of the algorithm is shown by the blue arrow.

Figure 4: An algorithm for escaping exact local minimizers

Lemma 2 (Escaping exact local minimizers)**.** *Suppose that Assumption 1 holds and let $(s, \hat{\theta}) =$* Escape-Exact-Local-Min(s, θ)*. Then* Forward$(s, \theta) =$ Forward$(s, \hat{\theta})$*, and $\mathcal{K}_0(s, \hat{\theta}) = \emptyset$.*

Proof. By the definition of $\mathcal{K}_0$, if $i \in \mathcal{K}_0(s, \theta)$ then $\eta_i = \mu_i$. Therefore Forward$(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) =$ Forward$(s, \theta_{1:i}, \hat{\theta}_{i+1:n})$, and by induction Forward$(s, \theta) =$ Forward$(s, \hat{\theta})$.

Next, we show that $i \notin \mathcal{K}_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n})$. If $i \notin \mathcal{K}_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n})$ then $\theta_i = \hat{\theta}_i$ so the result trivially holds. On the other hand, if $i \in \mathcal{K}_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n})$ then by definition of $\hat{\theta}_i$,

$$\hat{\theta}_i \left(\frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i} \right)^+ + (1 - \hat{\theta}_i) \left(\frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i} \right)^- = 0 \tag{9}$$

which implies $i \notin \mathcal{K}_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n})$. Further note that Forward$(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) =$ Forward$(s, \theta_{1:i}, \hat{\theta}_{i+1:n})$ implies if $j \leq i$ and $j \notin \mathcal{K}_0(s, \theta_{1:j-1}, \hat{\theta}_{j:n})$ then $j \notin \mathcal{K}_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n})$. By induction we deduce $\mathcal{K}_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) \subseteq \{1, \dots, i-1\}$ and hence $\mathcal{K}_0(s, \hat{\theta})$ is empty. □

A critical feature of Escape-Exact-Local-Min(s, θ) is that we work backwards (i.e., $i = n, \dots, 1$ rather than $i = 1, \dots, n$). This is critical because if we work forwards instead of backwards then (9) would become

$$\hat{\theta}_i \left(\frac{\partial \psi_i(s, z_{1:i}, \theta_{i+1:n})}{\partial z_i} \right)^+ + (1 - \hat{\theta}_i) \left(\frac{\partial \psi_i(s, z_{1:i}, \theta_{i+1:n})}{\partial z_i} \right)^- = 0$$

which, due to the replacement of θ with $\hat{\theta}$ inside ψ_i, is insufficient to establish $\mathcal{K}_0(s, \hat{\theta})$ is empty.

Finally, we remark that $g_i := \frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i}$ can be computed via the recursion

$$g_i \leftarrow \frac{\partial f}{\partial z_i} + \sum_{j=i+1}^{n} g_j \left(\hat{\theta}_j \frac{\partial \eta_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \mu_j}{\partial z_i} \right),$$

and therefore calling ESCAPE-EXACT-LOCAL-MIN takes the same time as computing $\nabla_\theta \psi_0$.

2.5.2 Escaping the basin of a local minimizer

If we modify SIMPLE-PSI-MINIMIZATION to run ESCAPE-EXACT-LOCAL-MIN(s, θ) whenever the set $\mathcal{K}_0(s^k, \theta^k)$ is nonempty then we would escape exact local minimizers. However, that does not exclude the possibility of asymptotically converging to a local minimizer. Therefore we need a method that will escape the basin of a local minimizer. In particular, we must be able to change the value of the θ_i variables with $i \in \mathcal{K}_\gamma(s, \theta)$ for $\gamma > 0$. This, however, introduces technical complications because if $\eta_i > \mu_i$ then as we change θ_i the value of $z_{i:n}$ could change.

Due to these technical complications we defer the algorithm and analysis to Appendix D, and informally state the main result here. The proof of Theorem 3 appears in Appendix D.1. The discussion given in Remark 2 also applies to Theorem 3 and means that the constant C could be large.

Theorem 3. *Suppose that Assumptions 1, 2, and 3 hold. Then there exists an algorithm obtaining an ϵ-duality gap after $C\epsilon^{-3} + 1$ computations of $\nabla \psi_0$ where C is a problem dependent constant.*

3 Experiments

We evaluate our method on robustness verification of models trained on CIFAR10 [23]. We benchmark on three sizes of networks trained with adversarial training [24]. The tiny network has two fully connected layers, with 100 units in the hidden layer. The small network has two convolutional layers, and two fully connected layers, with a total of 8308 hidden units. The medium network has four convolutional layers followed by three fully connected layers, with a total of 46912 hidden units.

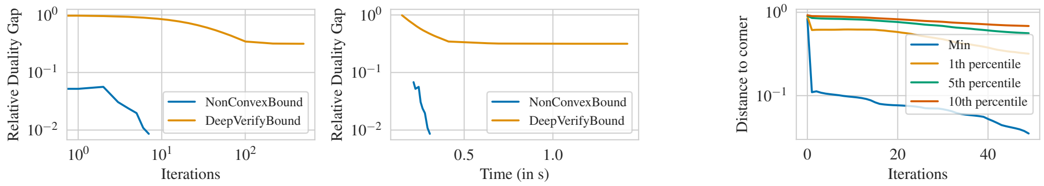
Verification of these networks is relaxed to a stage-wise convex problem (Appendix A.2). We compare three strategies for solving this relaxation: (i) NonConvex, our nonconvex reformulation using SIMPLE-PSI-MINIMIZATION augmented with momentum and backtracking linesearch, (ii) DeepVerify [6] (DV) that performs Lagrangian relaxation on the bound computation problem, (iii) a direct encoding of the relaxation into CVXPY [28], with SCS [26] and ECOS [27] backends[2]. We

ReLU Activation	Average Bound			Runtime (ms)		
	Tiny	Small	Medium	Tiny	Small	Medium
IBP [25]	17.0	743	2.4e+6	5.5	3.1	3.3
DeepVerify [6]	13.7	544	1.6e+6	349	711	1.1e+3
NonConvex (Ours)	5.68	434.9	1.5e+6	91.2	177	175
CVXPY (SCS) [26]	5.64	-	-	1.7e+5	-	-
CVXPY (ECOS) [27]	5.64	-	-	4.3e+4	-	-

SoftPlus Activation	Average Bound			Runtime (ms)		
	Tiny	Small	Medium	Tiny	Small	Medium
IBP [25]	18.3	6.5e+3	2.0e+9	4	2.5	3.3
DeepVerify [6]	13.7	5.1e+3	1.5e+9	414	855	1.7e+3
NonConvex (Ours)	5.97	3.93e+3	1.3e+9	7.8	65	214
CVXPY (SCS) [26]	5.97	-	-	2.9e+5	-	-

Table 1: **Benchmark** For each model, we report the average bound achieved on the adversarial objective and the average runtime in milliseconds to obtain it, over the CIFAR-10 test set. IBP [25] does not perform any optimization so it has an extremely small runtime but the bounds it generates are much weaker. The off-the-shelf solvers are significantly slower than the first-order methods DeepVerify and NonConvex and were not feasible to run beyond the tiny network.

[2] We also ran tests on an internal primal-dual hybrid gradient implementation. It was not remotely competitive (failing to converge after 100,000 iterations on trialed instances) so we did not include it in the results.



(a) Evolution of the relative duality gap as a function of time or number of iteration, for the NonConvex and DeepVerify Solver.

(b) Distribution of distance to potentially degenerate points.

Figure 5: Evaluation on the Medium-sized network with SoftPlus activation function

terminate (i) after 50 iteration or when the relative duality gap is less than 10^{-2}, (ii) after 500 iterations or when its dual value is larger than the final value of NonConvex (NC) (details in Appendix F).

Table 1 shows that, compared with the specialized first-order method DV, our method is faster by a factor between 3 and 50 depending on the network architecture, and always produces tighter bounds. As the two methods solve problems that have the same optimal value, we hypothesize that the discrepancy is because the Lagrangian relaxation of DV contains an inner-maximization problem that makes its objective extremely non-smooth, slowing convergence.

In most problems, DV reaches the imposed iterations limit before convergence. This is quantified in Table 2 where we show that beyond the tiny network, DV does not reach a small enough dual gap to achieve early stopping. On the other hand, we observe that for NC, the scale of the network does not significantly impact the required number of iterations. Figure 5a shows an example of the evolution of the computed bound, where we can see that the objective of DV plateaus, while NC converges in few iterations. Since the time per iteration for both methods is roughly the same, our runtime is lower.

After a single iteration, the duality gap achieved by our method is considerably smaller. The variables of DV exist on an unbounded feasible domain and appropriate initial values are therefore difficult to estimate, leading to large initial duality gap. Our method does not suffer from this problem, as all our variables are constrained between 0 and 1, and we can therefore initialize them all to 0.5, which empirically gives us good performance.

Nondegeneracy in practice. In Section 2.4, we described a simple version of our algorithm under the assumption that the algorithm does not enter a degenerate region. In the context of Neural Network verification, due to the structure of the problem, the only possibility for a small gap between $\eta_i - \mu_i$ is at the boundary of the feasible domain of the convex hull relaxation of activation. Even points close to the corner are not necessarily degenerate as they also need to satisfy a condition on the gradients. Throughout optimization, we measure $\frac{\min\{z_i - l_i, u_i - z_i\}}{u_i - l_i}$ where l_i and u_i are lower and upper bounds on z_i (corresponding to the corners), as shown in Figure 5b. We can observe that this value is strictly positive for all i which means we are not entering the degenerate region. This explains why, for these problems, SIMPLE-PSI-MINIMIZATION was able to converge to good solutions.

	Early stopping %		**Avg iteration count**	
	DV	NC	DV	NC
Tiny ReLU	37%	73%	384	18
Small ReLU	0%	97%	500	9
Medium ReLU	63%	100%	284	5
Tiny SoftPlus	14 %	100%	467	4
Small SoftPlus	0 %	100%	500	7
Medium SoftPlus	0 %	59%	500	25

Table 2: Proportion of bound computations on CIFAR-10 where the algorithm converges within the iteration budget, and average number of iterations.

Conclusion: We have developed a novel algorithm for a class of stage-wise convex optimization problems. Our experiments showed that our algorithm is efficient at solving standard relaxations of neural network verification problems. We believe that these results will generalize to stronger relaxations [29], as well as other stage-wise convex problems such as those arising in optimal control and generalized isotonic regression.

Broader Impact

Our work leads to new scalable algorithms for verifying properties of neural networks and solve certain kinds of structured regression problems. On the positive side, these can have an impact in terms of better methods to evaluate the reliability and trustworthiness of state of the art deep learning systems, thereby catching any unseen failure modes and preventing undesirable consequences of deep learning models. On the negative sign, the algorithms are agnostic to the type of properties being verified and may facilitate abuses by allowing attackers to verify that their attacks can reliabily induces specific failure modes in a deep learning model. Further, any applications of these techniques is reliant on carefully designing desirable specifications or properties of a deep learning model - if this is not done carefully, even systems that are verifiable with these algorithms may exhibit undesirable behavior (arising from bias in the data or the specification).

Acknowledgments and Disclosure of Funding

We thank Miles Lubin for establishing the connections between the authors and helpful feedback on the paper. We'd also like to thank Ross Anderson, Christian Tjandraatmadja, and Juan Pablo Vielma for helpful discussions.

References

[1] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[2] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 536–543, 2003.

[3] David Gamarnik and Julia Gaudio. Sparse high-dimensional isotonic regression. In *Advances in Neural Information Processing Systems*, pages 12852–12862, 2019.

[4] Ronny Luss, Saharon Rosset, Moni Shahar, et al. Efficient regularized isotonic regression with application to gene–gene interaction search. *The Annals of Applied Statistics*, 6(1):253–283, 2012.

[5] Rudy R Bunel, Ilker Turkaslan, Philip Torr, Pushmeet Kohli, and Pawan K Mudigonda. A unified view of piecewise linear neural network verification. *Advances in Neural Information Processing Systems*, 2018.

[6] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. *UAI*, 2018.

[7] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ICML*, 2018.

[8] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems*, 2019.

[9] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[11] Andrew R Conn, Nick Gould, and Ph L Toint. Numerical experiments with the LANCELOT package (release a) for large-scale nonlinear optimization. *Mathematical Programming*, 73(1): 73, 1996.

[12] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

[13] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

[14] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

[15] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[16] Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[17] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103 (1):127–152, 2005.

[18] Damek Davis and Dmitriy Drusvyatskiy. Stochastic subgradient method converges at the rate $O(k^{-1/4})$ on weakly convex functions. *arXiv preprint arXiv:1802.02988*, 2018.

[19] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[20] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[21] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, pages 379–387, 2015.

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2013.

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

[25] Sven Gowal, Krishnamurthy (Dj) Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. *ICCV*, 2019.

[26] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

[27] Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. *European Control Conference (ECC)*, 2013.

[28] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *JMLR*, 2016.

[29] Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, pages 1–37, 2020.

[30] L Lasdon, S Mitter, and A Waren. The conjugate gradient method for optimal control problems. *IEEE Transactions on Automatic Control*, 12(2):132–138, 1967.

[31] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[32] Rudy Bunel, Alessandro De Palma, Alban Desmaison, Krishnamurthy Dvijotham, Pushmeet Kohli, Philip HS Torr, and M Pawan Kumar. Lagrangian decomposition for neural network verification. *UAI*, 2020.

[33] Moonkyung Ryu, Yinlam Chow, Ross Anderson, Christian Tjandraatmadja, and Craig Boutilier. CAQL: Continuous action Q-learning. *ICLR*, 2020.

[34] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[35] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL): 1–30, 2019.

[36] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018.

[37] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems*, pages 6367–6377, 2018.

[38] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 2, 2018.

[39] Matthew Rosencrantz, Geoffrey Gordon, and Sebastian Thrun. Locating moving entities in indoor environments with teams of mobile robots. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 233–240, 2003.

[40] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[41] Guillaume Obozinski, Gert Lanckriet, Charles Grant, Michael I Jordan, and William Stafford Noble. Consistent probabilistic outputs for protein function prediction. *Genome Biology*, 9(1): S6, 2008.

[42] Michael J Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

[43] Junyu Zhang, Mingyi Hong, and Shuzhong Zhang. On lower iteration complexity bounds for the saddle point problems. *arXiv preprint arXiv:1912.07481*, 2019.

[44] Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

[45] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8:231–357, 2015.

[46] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

A Examples of optimization problems with this structure

A.1 Linear quadratic control and extensions

Linear quadratic control problems [1] take the form:

$$\underset{x}{\text{minimize}}\frac{1}{2}\sum_{t=1}^{M} x(t)^T Q(t)x(t) + u(t)^T R(t)u(t) \tag{10a}$$

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{10b}$$

$$x(0) = x_{\text{initial}} \tag{10c}$$

where $A(t), B(t) \in \mathbf{R}^{n\times n}$ are matrices, $Q(t), R(t) \in \mathbf{R}^{n\times n}$ are symmetric positive definite matrices, and $x(t) \in \mathbf{R}^n$ represents the system state, $u(t) \in \mathbf{R}^n$ the input, the initial system state is $x_{\text{initial}} \in \mathbf{R}^n$, and the positive integer M is the number of time steps. This problem can be solved by dynamic programming using $O(Mn^3)$.

Another approach [30] is to reformulate by eliminating the x variables by forward propagation, thereby rewriting the problem as

$$\underset{u}{\text{minimize}}\ h(u). \tag{11}$$

The gradient of (11) can be computed by backpropagation which takes time proportional to the total number of non-zeros in $A(t)$, $B(t)$, $Q(t)$, and $R(t)$. One can therefore solve this problem using gradient descent and due to the lower iteration cost, potentially find an approximate minimizer faster than using dynamic programming. The function $h(u)$ is a convex quadratic, which implies gradient descent finds the global minimizer. It is worth noting that applying our nonconvex reformulation to (10) by letting $s \leftarrow u$ and $z \leftarrow x$ yields $h(u) = \psi_0(u, \theta)$. Therefore, for linear quadratic control our approach and the approach of Lasdon et al. [30] are essentially equivalent.

However, one benefit of our approach is that we can tackle a wider range of problems than these classical methods. For example, we can support more complex dynamics where $x(t+1)$ is wedged between a convex and concave function.

A.2 Verification of neural networks robustness to adversarial attacks

To provide guarantees on the behaviour of neural networks, there has been a surge of interest in verifying that the output classification of a trained model remains stable when the input is slightly perturbed (adversarially) [5–7]. In particular, consider an input s_0 with label c^*. We wish to show there exists no adversarial example close to s_0 such that the network outputs $c \neq c^*$. Define S as the restriction of the input domain over which we want to perform verification. In the context of robustness to adversarial attacks, this would typically correspond to $S = \{s \mid \|s - s_0\|_\infty \leq \epsilon\}$. The set of feasible activation values for a feedfoward neural network for any input $s \in S$ satisfy,

$$s \in S \tag{12a}$$

$$z_{1:j(1)} = \sigma(W_0 s + b_0) \tag{12b}$$

$$z_{j(k)+1:j(k+1)} = \sigma(W_k z_{j(k-1)+1:j(k)} + b_k), \quad \forall k \in \{1, \dots, K-1\}, \tag{12c}$$

where k represents layers, $j(0), \dots, j(K)$ partitions the vector z such that $z_{j(k-1)+1:j(k)}$ is the activation values for layer k, W_k is the weight matrix for the kth layer, and σ represents the activation function (e.g., ReLU). By splitting the matrices W_k into a sequence of vectors w_i and the vectors b_k into a sequence of numbers h_i this can be rewritten in the form

$$s \in S \tag{13a}$$

$$z_i = \sigma([s, z_{1:i-1}] \cdot w_i + h_i) \quad \forall i \in \{1, \dots, n\}. \tag{13b}$$

To make the definitions precise, $w_i = [\mathbf{0}, [W(k)]_{i-j(k-1)}]$ where k is the unique solution to $j(k-1)+1 \leq i \leq j(k)$, and $[W(k)]_{i-j(k-1)}$ denotes the $i - j(k-1)$th row of $W(k)$; $h_i = [b_k]_{i-j(k-1)}$. Note that (13) is more general than (12) as it could capture more than just feedforward networks.

We now describe a procedure to verify the network. Let C be the set of possible output classes from the network and v_c a weight vector for each class $c \in C$. Typically neural networks classify an example according to the rule

$$\underset{c\in C}{\operatorname{argmax}}\, v_c \cdot z.$$

Usually, v_c is a sparse vector with zeros in all entries except those corresponding to the last layer of the network.

Therefore to verify that the network will output class c^* for all inputs in S it suffices to solve

$$\underset{z}{\text{minimize}}\ (v_{c^*} - v_c) \cdot z \quad \text{subject to} \quad (13),$$

for each $c \in C \setminus \{c^*\}$. If the minimum value of each of these subproblems is positive then the network is robust to adversarial perturbations.

Unfortunately, this problem is intractable as the feasible region given by (13) is nonconvex, and moreover the problem is in general NP-hard [31]. However, this does not preclude the possibility of verifying the neural network by forming a convex relaxation of (13). To form this convex relaxation of (13), we need lower and upper bounds on the possible values for each value of $[s, z_{1:i-1}] \cdot w_i$. These bounds can be obtained either by optimization over the partially constructed problem or by simple bound propagation [25]. Let us denote these bounds by l_i and u_i. In the case where σ is a ReLU we define the convex relaxation in the form of (1) with

$$f(s, z) = (v_c - v_{c'}) \cdot z$$

and for all $i \in \{1, \dots, n\}$,

$$\mu_i(s, z_{1:i-1}) = \sigma([s, z_{1:i-1}] \cdot w_i + h_i) = \max\{[s, z_{1:i-1}] \cdot w_i + h_i, 0\}$$

$$\eta_i(s, z_{1:i-1}) = \begin{cases} \frac{u_i}{u_i - l_i}([s, z_{1:i-1}] \cdot w_i + h_i - l_i) & 0 \in [l_i, u_i] \\ [s, z_{1:i-1}] \cdot w_i + h_i & l_i \geq 0 \\ 0 & u_i \leq 0 \end{cases}$$

where μ_i and η_i are depicted in Figure 6. Since ReLU is a convex function, this feasible region is convex. Definition of the constraints corresponding to the convex hull relaxation of different type of non-linearities have been previously published in the literature [8, 32].

Due to the way that the lower and upper bounds are constructed, this convex relaxation satisfies Assumption 3. In particular, if we form the bounds by optimizing over the partially constructed problem, i.e.,

$$l_j = h_j + \underset{(s,z) \in S \times \mathbf{R}^n}{\text{minimize}}\ [s, z_{1:j-1}] \cdot w_j \quad \text{s.t.} \quad \mu_i(s, z_{1:i-1}) \leq z_i \leq \eta_i(s, z_{1:i-1}) \quad \forall i \in \{1, \dots, j-1\}$$

$$u_j = h_j + \underset{(s,z) \in S \times \mathbf{R}^n}{\text{maximize}}\ [s, z_{1:j-1}] \cdot w_j \quad \text{s.t.} \quad \mu_i(s, z_{1:i-1}) \leq z_i \leq \eta_i(s, z_{1:i-1}) \quad \forall i \in \{1, \dots, j-1\}$$

then we can see that given we have a feasible solution $[s, z_{1:j-1}]$ to $\mu_i(s, z_{1:i-1}) \leq z_i \leq \eta_i(s, z_{1:i-1})\ \forall i \in \{1, \dots, j-1\}$ then $[s, z_{1:j-1}] \cdot w_j + h_j \in [l_i, u_i]$ which implies that for $0 \notin [l_i, u_i]$ that $\eta_j(s, z_{1:j-1}) - \mu_j(s, z_{1:j-1}) \geq 0$ by definition and if $0 \in [l_i, u_i]$ then

$$\begin{aligned} \eta_j(s, z_{1:j-1}) - \mu_j(s, z_{1:j-1}) &= \frac{u_i}{u_i - l_i}([s, z_{1:i-1}] \cdot w_i - l_i) - \max\{[s, z_{1:i-1}] \cdot w_i, 0\} \\ &\geq \min\left\{\frac{u_i}{u_i - l_i}(u_i - l_i), \frac{u_i}{u_i - l_i}(l_i - l_i)\right\} \\ &= 0 \end{aligned}$$

as required to establish Assumption 3 (intuition for this can be given by contrasting Figure 2 with Figure 6). By a similar argument, simple bound propagation [25] to compute l_i and u_i will also guarantee Assumption 3 holds. In general, if a relaxation is constructed in an inductive fashion Assumption 3 tends to naturally hold.

There exists tighter linear programming bounds for neural network verification that require an (implicit) exponential number of inequalities [29]. Ignoring smoothness issues, these also can be cast into our framework. Using our approach to solve these linear programs is an interesting avenue for future work. These types of relaxations also have applications in reinforcement learning [33].

Related work on Deep Network verification The convex relaxation was proposed by Ehlers [Ehlers, ATVA 2017] but the high computational cost of solving it with off-the-shelf LP solvers was prohibitive for large instances. A large number of papers such as IBP [34], DeepPoly [35], Crown [36], Neurify [37], LP-relaxed-dual [7] focused on looser relaxations to allow for fast, closed

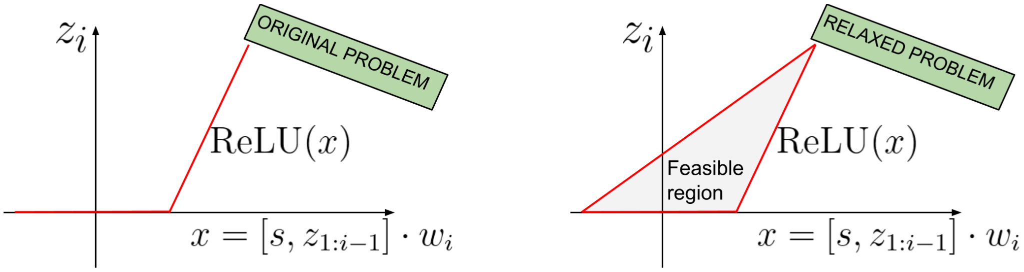


Figure 6: Comparison of original feasible region with convex relaxation when σ is ReLU.

form solutions of the bound computation problem scaling to larger networks. In parallel, work was done to reformulate the optimization problem to allow the use of better algorithms: DeepVerify [38] introduced an unconstrained dual reformulation of the non-convex problem and showed equivalence with the convex relaxation. Proximal [32] performed lagrangian decomposition and used proximal methods to solve the problem faster. The application of our method to network certification follows this research direction of speeding up the computation of network bounds without compromising on tightness.

A.3 Finite horizon Markov decision processes with cost function controlled by an adversary

Consider a finite horizon Markov Decision Process with uncertain rewards. In particular, at each time stage $t \in \{1, \dots, T\}$ we take an action $a \in A$ and move from state $k \in K$ to a state $k' \in K$ with probability $P_{a,k,k',t}$ and earn reward $R_{a,k,t}$. Further suppose the rewards are uncertain but we know $R \in \mathcal{R}$ where $\mathcal{R}$ is a convex set. We wish to know what the optimal policy is given we start in state 1. This can be written as the following optimization problem

$$\min_{R \in \mathcal{R}, v} \quad v_{1,1} \tag{14a}$$

$$\max_{a \in A} \sum_{k' \in K} P_{a,k,k',t} v_{k',t+1} + R_{a,k,t} \leq v_{k,t} \tag{14b}$$

where $v_{k,T} = 0$. McMahan et al. [2] develops a specialized algorithm for this problem. This algorithm is used for robots playing laser tag [39].

We can also cast this problem in our framework. Suppose we have an upper bound on the rewards $R^{\max}_{a,k,t}$ ($R^{\max}_{a,k,t} \geq R_{a,k,t}$ for all $R \in \mathcal{R}$) then applying standard dynamic programming we can compute $v^{\max}_{k,t}$ allowing us to rewrite (14) in the form,

$$\min_{R \in \mathcal{R}, v} \quad v_{1,1}$$

$$\max_{a \in A} \sum_{k' \in K} P_{a,k,k',t} v_{k',t+1} + R_{a,k,t} \leq v_{k,t} \leq v^{\max}_{k,t}.$$

For this problem one can show Assumption 2 and 3 are satisfied. The only issue is that $\max_{a \in A} \sum_{k' \in S} P_{a,k,k',t} v_{k',t+1} + R_{a,k,t}$ is a nonsmooth function. Although, as we mention in Remark 1, this problem is likely surmountable.

A.4 Generalized Isotonic regression

Classic isotonic regression considers the following problem

$$\min_{z} \sum_{i=1}^{n} (z_i - y_i)^2 \quad \text{s.t.} \quad z_j \leq z_i \quad \forall\, j \prec i. \tag{15}$$

This problem has applications in genetics [3, 4], psychology [40], and biology [41]. When $\prec$ is a total ordering then there exists efficient algorithms that use only linear time [42]. However, for the general case developing efficient algorithms is an area of active research [4].

Our approach offers a new way of solving these problems. In particular, if we add the mild assumption that z_i is bounded in $[l, u]$ then (15) reduces to

$$\min_{z} \sum_{i=1}^{n} (z_i - y_i)^2 \quad \text{s.t.} \quad \max\{z_j, l\} \leq z_i \leq u \quad \forall\, j \prec i. \tag{16}$$

Note that if $\prec$ is a partial ordering then there exists a total ordering that is consistent with this partial ordering. We can represent such a total ordering by a permutation $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ where $i \preceq j \Rightarrow \pi(i) \leq \pi(j)$. Therefore, without loss of generality assume that that $i \preceq j \Rightarrow i \leq j$. This allows us to write the problem in the form of (1), as

$$\min_{z} \sum_{i=1}^{n} (z_i - y_i)^2 \quad \text{s.t.} \quad \mu_i(z_{1:i-1}) \leq z_i \leq \eta_i(z_{1:i-1}), \quad \forall i \in \{1, \dots, n\},$$

where $\mu_i(z_{1:i-1}) = \max \left\{ l, \operatorname*{maximum}_{j : j \prec i} z_j \right\}$ and $\eta_i(z_{1:i-1}) = u$. Given $\mu_i(z_{1:i-1}) \leq \eta_i(z_{1:i-1})$ for all $i < j$ then $z_i \leq u$ which implies that $\eta_j(z_{1:j-1}) \leq u$, i.e., Assumption 3 holds.

B Sketch of lower bounds

Here we briefly sketch how to use to show that solving (2) requires at least $n - 1$ iterations of saddle point methods. We only provide a sketch since very similar results are already known [43, 44]. Before reading this section we recommend reading a standard reference on lower bounds (e.g., [45, Section 3.5], [13, Section 2.1.2]) and a standard reference on saddle point methods (e.g., [45, Section 5.2]).

We can reformulate (2) as a saddle point problem as follows

$$\operatorname*{minimize}_{x \in X} \quad \operatorname*{maximize}_{y \in Y} - e_n^T x + y^T A x$$

where e_n is a vector with a one in the nth entry and all other values zero, $X := \{x \in \mathbf{R}^n : x \geq -1, x_1 \in [0, 1]\}$, $Y := \{y \in \mathbf{R}^{n-1} : y \geq 0\}$ and

$$A = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \cdots & & \\ & & & 1 & -1 \end{pmatrix}.$$

After appropriate reindexing the iterates $\{(x^t, y^t)\}_{t=0}^{\infty}$, saddle point algorithms such as primal-dual hybrid gradient [10] and mirror-prox [9] satisfy

$$\begin{aligned} \hat{x}_{t+1} &\in x_t + \text{span}(A^T y_0, \dots, A^T y_t) \\ x_{t+1} &= \Pi_X(\hat{x}_{t+1}) \\ \hat{y}_{t+1} &\in y_t + \text{span}(A x_0, \dots, A x_t) \\ y_{t+1} &= \Pi_Y(\hat{y}_{t+1}) \end{aligned}$$

where Π_X and Π_Y projects onto the set X and Y respectively.

Define $Z_X^t = \{x \in \mathbf{R}^n : x_i = 0, \forall i \in \{1, \dots, n - t\}\}$ and $Z_Y^t = \{y \in \mathbf{R}^{n-1} : y_i = 0, \forall i \in \{1, \dots, n - t - 1\}\}$. Now, if $t \in \{0, \dots n - 2\}$, $x^0, \dots, x^t \in Z_X^t$ and $y^0, \dots, y^t \in Z_Y^t$ then

- $\hat{x}^{t+1} \in x^0 + \text{span}(A^T y^0, \dots, A^T y^t) \subseteq x^t + Z_X^{t+1} = Z_X^{t+1} \Rightarrow x^{t+1} \in Z_X^{t+1}$.
- $\hat{y}^{t+1} \in y^0 + \text{span}(A x^0, \dots, A x^t) \subseteq y^t + Z_Y^{t+1} = Z_Y^{t+1} \Rightarrow y^{t+1} \in Z_Y^{t+1}$

where given a set S and a vector s' we define the addition of them by $S + s' := \{s + s' : s \in S\}$.

Therefore if $x^0 = \mathbf{0} \in Z_X^0$ and $y^0 = \mathbf{0} \in Z_X^0$ then by induction $x_1^t = 0$ for $t < n$.

C Proof results from Section 2.4

We will use the following fact throughout the proofs.

Fact 1. *Let* $(s, \theta) \in S \times [0, 1]^n$ *and* $z = \text{FORWARD}(s, \theta)$. *If Assumption 3 holds then* $\mu_i \leq \eta_i$.

As discussed in Section 2.3, this follows immediately from Theorem 1.

C.1 Proof of Lemma 1

Lemma 3 is a standard result on the relationship between progress made by a gradient step and the gradient of a function.

Lemma 3. *Suppose* $s \in S$. *If Assumption 2 holds then* $\sup_{\hat{s} \in S} \boldsymbol{\nabla}_s \psi_0 \cdot (s - \hat{s}) \leq D_s \sqrt{2 L \delta_L}$.

Proof. Note that for any $\hat{s} \in S$,

$$\delta_L(s,\theta) \geq -\underset{\alpha \in \mathbf{R}}{\text{minimize}}\ \alpha \boldsymbol{\nabla}_s \psi_0 \cdot (\hat{s} - s) + \frac{L}{2}\|\hat{s} - s\|_2^2 \alpha^2 = \frac{(\boldsymbol{\nabla}_s \psi_0 \cdot (s - \hat{s}))^2}{2L\|\hat{s} - s\|_2^2}$$

where the equality uses that $\alpha = \frac{\boldsymbol{\nabla}_s \psi_0 \cdot (s - \hat{s})}{L\|\hat{s} - s\|_2^2}$ minimizes the quadratic. Rearranging yields,

$$\boldsymbol{\nabla}_s \psi_0 \cdot (s - \hat{s}) \leq \|\hat{s} - s\|_2 \sqrt{2L\delta_L(s,\theta)}.$$

□

Lemma 4. *Suppose Assumption 2 holds, then*

$$\sup_{(\hat{s},\hat{z}) \in S \times Z} \boldsymbol{\nabla}_{s,z}\mathcal{L} \cdot (s - \hat{s}, z - \hat{z}) \leq D_s\sqrt{2L\delta_L} + D_s\|\boldsymbol{\nabla}_s\mathcal{L} - \boldsymbol{\nabla}_s\psi\|_2 + D_z\|\boldsymbol{\nabla}_z\mathcal{L}\|_2.$$

Proof. Moreover,

$$\begin{aligned}
&\sup_{(\hat{s},\hat{z}) \in S \times Z} \boldsymbol{\nabla}_{s,z}\mathcal{L} \cdot (s - \hat{s}, z - \hat{z}) \\
&= \sup_{\hat{s} \in S} \boldsymbol{\nabla}_s\mathcal{L} \cdot (s - \hat{s}) + \sup_{\hat{z} \in Z} \boldsymbol{\nabla}_z\mathcal{L} \cdot (z - \hat{z}) \\
&\leq \sup_{\hat{s} \in S} \boldsymbol{\nabla}\psi_s \cdot (s - \hat{s}) + \sup_{\hat{s} \in S}(\boldsymbol{\nabla}_s\mathcal{L} - \boldsymbol{\nabla}_s\psi) \cdot (s - \hat{s}) + \sup_{z \in Z} \boldsymbol{\nabla}_z\mathcal{L} \cdot (z - \hat{z}) \\
&\leq D_s\sqrt{2L\delta_L} + D_s\|\boldsymbol{\nabla}_s\mathcal{L} - \boldsymbol{\nabla}_s\psi\|_2 + D_z\|\boldsymbol{\nabla}_z\mathcal{L}\|_2
\end{aligned}$$

where the final inequality uses $\sup_{\hat{s} \in S} \boldsymbol{\nabla}\psi_s \cdot (s - \hat{s}) \leq D_s\sqrt{2L\delta_L}$ (Lemma 3) and Assumption 2. □

Display (8) establishes that $\Delta(s,\theta)$ is a valid duality gap. Moreover, Lemma 4 shows that to provide an upper bound on $\Delta(s,\theta)$ it will suffice to upper bound

$$\sum_{i=1}^{n}(y_i z_i - y_i^+ \mu_i + y_i^- \eta_i) + D_s\sqrt{2L\delta_L} + D_s\|\boldsymbol{\nabla}_s\mathcal{L} - \boldsymbol{\nabla}_s\psi\|_2 + D_z\|\boldsymbol{\nabla}_z\mathcal{L}\|_2.$$

Lemma 5 is our first step towards bounding these quantities. Before proceeding with Lemma 5 we prove a fact we will find useful.

Fact 2. *Let $\gamma, t, \alpha, \beta \in \mathbf{R}$ then*

$$(\gamma^+\beta - \gamma^-\alpha) - \gamma(t\alpha + (1-t)\beta) = (\beta - \alpha)(\gamma^+ t + \gamma^-(1-t)).$$

Proof. Observe that

$$\begin{aligned}
\gamma^+(\beta - (t\alpha + (1-t)\beta)) &= \gamma^+ t(\beta - \alpha) \\
\gamma^-(-\alpha + (t\alpha + (1-t)\beta)) &= \gamma^-(1-t)(\beta - \alpha),
\end{aligned}$$

adding the two expressions together gives the result. □

Lemma 5. *Suppose Assumption 4 holds. Let $y_i = \frac{\partial \psi_i}{\partial z_i}$ and $r_i = \theta_i y_i^+ + (1 - \theta_i) y_i^-$, then*

$$\|\boldsymbol{\nabla}_s\mathcal{L} - \boldsymbol{\nabla}_s\psi_0\|_2 \leq \|\boldsymbol{\nabla}_s\mu - \boldsymbol{\nabla}_s\eta\|_2\|r\|_2 \tag{17a}$$

$$\|\boldsymbol{\nabla}_z\mathcal{L}\|_2 \leq \|\boldsymbol{\nabla}_z\mu - \boldsymbol{\nabla}_z\eta\|_2\|r\|_2 \tag{17b}$$

$$\sum_{i=1}^{n}(y_i z_i - y_i^+ \mu_i + y_i^- \eta_i) \leq \|\eta - \mu\|_2\|r\|_2. \tag{17c}$$

Proof. Consider the expansion of (6a) and (6b) using $\psi_n = f$:

$$\boldsymbol{\nabla}_s\psi_i = \boldsymbol{\nabla}_s f + \sum_{j=i+1}^{n} \frac{\partial \psi_j}{\partial z_j}\left(\theta_j \boldsymbol{\nabla}_s\eta_j + (1 - \theta_j)\boldsymbol{\nabla}_s\mu_j\right)$$

$$\frac{\partial \psi_i}{\partial z_k} = \frac{\partial f}{\partial z_k} + \sum_{j=i+1}^{n} \frac{\partial \psi_j}{\partial z_j}\left(\theta_j \frac{\partial \eta_j}{\partial z_k} + (1 - \theta_j)\frac{\partial \mu_j}{\partial z_k}\right)$$

for each $i \in \{1, \dots, n\}$ and $k \in \{1, \dots, i\}$. Setting $k = i$ gives

$$\nabla_s \psi_0 = \nabla_s f + \sum_{j=1}^{n} \frac{\partial \psi_j}{\partial z_j} \left(\theta_j \nabla_s \eta_j + (1 - \theta_j) \nabla_s \mu_j \right) \tag{18a}$$

$$\frac{\partial \psi_i}{\partial z_i} = \frac{\partial f}{\partial z_i} + \sum_{j=i+1}^{n} \frac{\partial \psi_j}{\partial z_j} \left(\theta_j \frac{\partial \eta_j}{\partial z_i} + (1 - \theta_j) \frac{\partial \mu_j}{\partial z_i} \right). \tag{18b}$$

Contrast (18) with

$$\nabla_s \mathcal{L} = \nabla_s f + \sum_{j=1}^{n} \left(y_j^+ \nabla_s \mu_j - y_j^- \nabla_s \eta_j \right) \tag{19a}$$

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial f}{\partial z_i} - y_i + \sum_{j=i+1}^{n} \left(y_j^+ \frac{\partial \mu_j}{\partial z_i} - y_j^- \frac{\partial \eta_j}{\partial z_i} \right). \tag{19b}$$

One can see (18) and (19) share a very similar structure which we will exploit. In particular,

$$\begin{aligned} \nabla_s \mathcal{L} - \nabla_s \psi_0 &= \sum_{i=1}^{n} \left(y_j^+ \nabla_s \mu_j - y_j^- \nabla_s \eta_j - y_j \left(\theta_j \nabla_s \eta_j + (1 - \theta_j) \nabla_s \mu_j \right) \right) \\ &= \sum_{j=1}^{n} \left(\nabla_s \mu_j - \nabla_s \eta_j \right) \left(\theta_j y_j^+ + (1 - \theta_j) y_j^- \right). \end{aligned}$$

where the first equality subtracts (18a) from (19a), and the second equality uses Fact 2. We conclude (17a) holds. Similarly,

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial z_i} &= \sum_{j=i+1}^{n} \left(y_j^+ \frac{\partial \mu_j}{\partial z_i} - y_j^- \frac{\partial \eta_j}{\partial z_i} - y_j \left(\theta_j \frac{\partial \eta_j}{\partial z_i} + (1 - \theta_j) \frac{\partial \mu_j}{\partial z_i} \right) \right) \\ &= \sum_{j=i+1}^{n} \left(\frac{\partial \mu_j}{\partial z_i} - \frac{\partial \eta_j}{\partial z_i} \right) \left(\theta_j y_j^+ + (1 - \theta_j) y_j^- \right). \end{aligned}$$

where the first equality substitutes $y_i = \frac{\partial \psi_i}{\partial z_i}$ into (19b) and then subtracts (18b) from (19b), and the second equality uses Fact 2. We conclude (17b) holds. Finally, $z_i - \mu_i = (1 - \theta_i)\mu_i + \theta_i \eta_i - \mu_i = \theta_i(\eta_i - \mu_i)$ and $\eta_i - z_i = (1 - \theta_i)(\eta_i - \mu_i)$ which implies

$$\sum_{i=1}^{n} y_i^+ (z_i - \mu_i) + y_i^- (\eta_i - z_i) = \sum_{i=1}^{n} (\eta_i - \mu_i)(y_i^+ \theta_i + y_i^- (1 - \theta_i)) = \sum_{i=1}^{n} r_i (\eta_i - \mu_i),$$

establishing (17c). □

Lemma 6. *Suppose Assumption 2 holds. Let* $y_i = \frac{\partial \psi_i}{\partial z_i}$ *and* $r_i = \theta_i y_i^+ + (1 - \theta_i) y_i^-$, *then* $\Delta(s, \theta) \leq D_s \sqrt{2L\delta_L} + (\|\eta - \mu\|_2 + D_s \|\nabla_s \mu - \nabla_s \eta\|_2 + D_z \|\nabla_z \mu - \nabla_z \eta\|_2) \|r\|_2$.

Proof. Note that by (8), Lemma 4 and 5,

$$\begin{aligned} \Delta(s, \theta) &\leq \sum_{i=1}^{n} (y_i z_i - y_i^+ \mu_i + y_i^- \eta_i) + D_s \sqrt{2L\delta_L} + D_s \|\nabla_s \mathcal{L} - \nabla_s \psi\|_2 + D_z \|\nabla_z \mathcal{L}\|_2 \\ &\leq \left(\|\eta - \mu\|_2 \|r\|_2 + D_s \left(\sqrt{2L\delta_L} + \|\nabla_s \mu - \nabla_s \eta\|_2 \|r\|_2 \right) + D_z \|\nabla_z \mu - \nabla_z \eta\|_2 \|r\|_2 \right). \end{aligned}$$

□

We will find Lemma 6 useful later in Section D.

Next, define

$$t_{i,L} := -\underset{\theta_i + d_i \in [0,1]}{\text{minimize}} \; \frac{\partial \psi_0}{\partial \theta_i} d_i + \frac{L}{2} d_i^2$$

which represent the guaranteed reduction from a gradient step, contributed by θ_i, assuming the function ψ_0 is L-smooth.

While Lemma 6 represents useful progress in bounding $\Delta(s, \theta)$. We would like our final bound on $\Delta(s, \theta)$ to depend only on δ_L and problem constants. Lemma 7 allows us to do that.

Lemma 7. *Suppose $\eta_i - \mu_i > 0$. Let $y_i = \frac{\partial \psi_i}{\partial z_i}$ and $r_i = \theta_i y_i^+ + (1-\theta_i) y_i^-$, then*

$$r_i \leq \frac{\max\left\{\sqrt{2Lt_{i,L}}, 2t_{i,L}\right\}}{\eta_i - \mu_i}. \tag{20}$$

Proof. Suppose $\frac{\partial \psi_0}{\partial \theta_i} \geq 0$ and $\theta_i \leq \frac{1}{L}\frac{\partial \psi_0}{\partial \theta_i}$ then

$$t_{i,L} \geq \theta_i \left(\frac{\partial \psi_0}{\partial \theta_i} - \frac{L\theta_i}{2} \right) \geq \frac{\theta_i}{2} \frac{\partial \psi_0}{\partial \theta_i}.$$

If $\frac{\partial \psi_0}{\partial \theta_i} \geq 0$ and $\theta_i \geq \frac{1}{L}\frac{\partial \psi_0}{\partial \theta_i}$ then $t_{i,L} \geq \frac{1}{2L}\left(\frac{\partial \psi_0}{\partial \theta_i}\right)^2$. Therefore, if $\frac{\partial \psi_0}{\partial \theta_i} \geq 0$ then

$$t_{i,L} \geq \frac{1}{2} \min\left\{ \frac{1}{L}\left(\frac{\partial \psi_0}{\partial \theta_i}\right)^2, \theta_i \frac{\partial \psi_0}{\partial \theta_i} \right\} \tag{21}$$

which implies that

$$(\eta_i - \mu_i)\frac{\partial \psi_i}{\partial z_i}\theta_i = \frac{\partial \psi_0}{\partial \theta_i}\theta_i \leq \max\left\{\theta_i\sqrt{2Lt_{i,L}}, 2t_{i,L}\right\} \tag{22}$$

where the equality uses (6c) and the inequality rearranges (21). By the same argument, if $\frac{\partial \psi_0}{\partial \theta_i} \leq 0$ then

$$-(\eta_i - \mu_i)\frac{\partial \psi_i}{\partial z_i}(1-\theta_i) \leq \max\left\{(1-\theta_i)\sqrt{2Lt_{i,L}}, 2t_{i,L}\right\}. \tag{23}$$

By (22) and (23) we deduce (20). □

Proof of Lemma 1. Observe that

$$\gamma^2 \|r\|_2^2 \leq \sum_{i=1}^{n} \max\left\{2Lt_{i,L}, 4t_{i,L}^2\right\} \leq \sum_{i=1}^{n} 2Lt_{i,L} + 4t_{i,L}^2 \leq 2L\delta_L + 4\delta_L^2 \leq 4L\delta_L \tag{24}$$

where the first inequality uses (20), the second inequality uses that $\sum_{i=1}^{n} t_{i,L} \leq \delta_L$ and last inequality uses the assumption that $\delta_L \leq L/2$. Combining equation (24), Lemma 6 and Assumption 4 gives the result. □

C.2 Proof of Theorem 2

Rather than directly prove the Theorem 2, we first prove Lemma 8 which is a generic statement on the convergence of algorithms to minimizers. We will find Lemma 8 useful later in Section D.

Lemma 8. *Let $\zeta_1, \zeta_2, \zeta_3 \in (0, \infty)$. Consider a sequence $(s^k, \theta^k)_{k=0}^{\infty}$ satisfying*

$$\left(f(s^k, \theta^k) - f_*\right)^{\zeta_1 + 1} \leq \zeta_2 (f(s^k, \theta^k) - f(s^{k+1}, \theta^{k+1})) \tag{25}$$

for all $\frac{f(s^1, \theta^1) - f_*}{f(s^k, \theta^k) - f(s^{k+1}, \theta^{k+1})} \leq \zeta_3$. Then for $K > \zeta_3$

$$f(s^k, \theta^k) - f_* \leq \left(\frac{\zeta_2}{K - \zeta_3}\right)^{1/\zeta_1}.$$

Proof. Define, $f(s^k, \theta^k) - f_* = v^k$. First consider the case that $\frac{f(s^k, \theta^k) - f(s^{k+1}, \theta^{k+1})}{f(s^1, \theta^1) - f_*} \leq \zeta_3$, then by (25) and $v^k - v^{k+1} = f(s^k, \theta^k) - f(s^{k+1}, \theta^{k+1})$ we deduce

$$\frac{(v^k)^{\zeta_1 + 1}}{\zeta_2} \leq v^k - v^{k+1} \Rightarrow v^{k+1} \leq v^k \left(1 - \frac{(v^k)^{\zeta_1}}{\zeta_2}\right).$$

Dividing both sides by $v^{k+1}(v^k)^{\zeta_1}$ and using that $v^{k+1} \leq v^k$ yields

$$\frac{1}{(v^k)^{\zeta_1}} \leq \frac{v^k}{v^{k+1}} \left(\frac{1}{(v^k)^{\zeta_1 - 1}} - \frac{1}{\zeta_2} \right) \leq \frac{1}{(v^{k+1})^{\zeta_1}} - \frac{1}{\zeta_2}. \tag{26}$$

Furthermore, if $\frac{f(s^k,\theta^k) - f(s^{k+1},\theta^{k+1})}{f(s^1,\theta^1) - f_*} \geq \zeta_3$ then

$$v^{k+1} \leq v^k - \frac{f(s^1,\theta^1) - f_*}{\zeta_3} \tag{27}$$

and this can happen at most ζ_3 times. Therefore if $K > \zeta_3$ then

$$\frac{1}{(v^K)^{\zeta_1}} \geq \frac{1}{(v^K)^{\zeta_1}} - \frac{1}{(v^1)^{\zeta_1}} = \sum_{k=1}^{K-1} \frac{1}{(v^{k+1})^{\zeta_1}} - \frac{1}{(v^k)^{\zeta_1}} \geq \frac{K - \zeta_3}{\zeta_2}$$

where the first inequality uses that $v_1 \geq 0$, and second inequality uses (26). Rearranging gives the result. □

Proof of Theorem 2. Define

$$\begin{aligned} \zeta_1 &= 1 \\ \zeta_2 &= L\left(D_s\sqrt{2} + 2\frac{c}{\gamma}\right)^2 \\ \zeta_3 &= \frac{2\Delta(s^1,\theta^1)}{L}. \end{aligned}$$

Lemma 1 shows that if $\delta_L(s^k,\theta^k) \leq \Delta(s^1,\theta^1)/\zeta_3$ then $\Delta(s^k,\theta^k)^2 \leq \zeta_2 \delta_L(s^k,\theta^k)$. Furthermore, $\delta_L(s^k,\theta^k) \leq \psi_0(s^k,\theta^k) - \psi_0(s^{k+1},\theta^{k+1}) = f(s^k,z^k) - f(s^{k+1},z^{k+1})$ by line 10 of Algorithm 1 and the definition of ψ_0 respectively. Combining these two inequalities yields $(f(s^k,z^k) - f_*)^2 \leq \zeta_2(f(s^k,z^k) - f(s^{k+1},z^{k+1}))$. Applying Lemma 8 to the latter inequality yields the result. □

D Escaping the basin of a local minimizer

For this section, we consider an alternative set to $\mathcal{K}_\gamma$:

$$\mathcal{C}(s,\theta,q) := \left\{ i : \theta_i \left(\frac{\partial \psi_i}{\partial z_i} \right)^+ + (1 - \theta_i) \left(\frac{\partial \psi_i}{\partial z_i} \right)^- > 2q(\eta_i - \mu_i) \right\}$$

this set identifies the indices where the degeneracy could cause a convergence issue. We then modify SIMPLE-PSI-MINIMIZATION (yielding SAFE-PSI-MINIMIZATION) to detect when the set $C(s,\theta,q)$ is empty and take appropriate action, i.e., call FIX-DEGENERACY. FIX-DEGENERACY extends ESCAPE-EXACT-LOCAL-MIN to allow us to escape from the basin of a local minimizer. To see this, note that if (s,θ) is fixed then setting q sufficiently large will cause FIX-DEGENERACY(s,θ,q) to reduce to ESCAPE-EXACT-LOCAL-MIN(s,θ). However, the value of q needed to achieve this could be arbitrarily large.

Algorithm 2 Local search algorithm that will find the global minimizer of ψ_0.

```
function SAFE-PSI-MINIMIZATION($s^1, \theta^1, L, q^1, \epsilon$)
    for $k = 1, \ldots, \infty$ do
        Take corrective action if degeneracy is an issue:
        if $\mathcal{C}(s^k, \theta^k, q^k) \neq \emptyset$ then
            $(s^k, \hat{\theta}^k, \textbf{status}) \leftarrow$ FIX-DEGENERACY$(s^k, \theta^k, q^k)$
            if status $=$ FAILURE then
                $q^{k+1} \leftarrow 10q^k$
            end if
        else
            $\hat{\theta}^k \leftarrow \theta^k$
        end if
        Termination checks:
        if $\Delta(s^k, \hat{\theta}^k) \leq \epsilon$ then
            return $(s^k, \theta^k)$
        end if
        Reduce the function at least as much as PGD would:
        $(s^{k+1}, \theta^{k+1}) \in \{(s, \theta) : \psi_0(s, \theta) \leq \psi_0(s^k, \hat{\theta}^k) - \delta_L(s^k, \hat{\theta}^k)\}$
    end for
end function
```

Algorithm 3 Algorithm for fixing convergence issues in degenerate case

```
function FIX-DEGENERACY($s, \theta, q$)
    $z =$ FORWARD$(s, \theta)$
    $\hat{\theta} \leftarrow \text{copy}(\theta)$
    The minimum reduction in $\psi_0$ if $q \geq Q$:
    $v \leftarrow 0$

    for $i = n, \ldots, 1$ do
        Approximately compute $\frac{\partial \psi_i(s, z_{1:i}, \hat{\theta}_{i+1:n})}{\partial z_i}$:
        $g_i \leftarrow \frac{\partial f}{\partial z_i} + \sum_{j=i+1}^{n} g_j \left( \hat{\theta}_j \frac{\partial \eta_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \mu_j}{\partial z_i} \right)$
        Estimate the distance that z has moved:
        $\omega_i \leftarrow \sum_{j=i+1}^{n} \left| \theta_j - \hat{\theta}_j \right| (\eta_j - \mu_j)$
        $p_i \leftarrow g_i^+ \theta_i + g_i^- (1 - \theta_i)$
        if $p_i > 2q(\omega_i + \eta_i - \mu_i)$ then
            Fix degeneracy in index i:
            $\hat{\theta}_i \leftarrow \begin{cases} 0 & g_i > 0 \\ 1 & g_i < 0 \end{cases}$
            $v \leftarrow v + \frac{1}{2} p_i (\eta_i - \mu_i)$
        end if
    end for

    $\omega_0 \leftarrow \sum_{j=1}^{n} \left| \theta_j - \hat{\theta}_j \right| (\eta_j - \mu_j)$
    if $\psi_0(s, \hat{\theta}) > \psi_0(s, \theta) - v$ then
        return $(s, \theta, \text{FAILURE})$
    else if $\left| g_i - \frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right| \leq q\omega_0, \forall i$ then
        return $(s, \hat{\theta}, \text{SUCCESS})$
    else
        return $(s, \hat{\theta}, \text{FAILURE})$
    end if
end function
```

Remark 5. *With careful implementation the cost of running* FIX-DEGENERACY *is the same as one backpropagation.*

Remark 6. *If $\mathcal{C}(s, \theta, q) = \emptyset$ then $(s, \hat{\theta}, \textbf{status}) \leftarrow$* FIX-DEGENERACY$(s, \theta, q)$ *satisfies $\hat{\theta} = \theta$ and **status** $=$* SUCCESS. *In other words, removing Line 4 and Line 9-11 of* SAFE-PSI-MINIMIZATION *does not change the behaviour of the Algorithm (although it may create unnecessary computation).*

This section introduces two new assumptions (Assumption 5 and 6). We defer justifying these assumptions to Section E where we show that Assumptions 1, 2, and 3 imply that these introduced assumptions hold.

Assumption 5. *Let $z \leftarrow$* FORWARD(s, θ)*,*

$$g_i = \frac{\partial f}{\partial z_i} + \sum_{j=i+1}^{n} g_j \left(\hat{\theta}_j \frac{\partial \eta_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \mu_j}{\partial z_i} \right),$$

and $\hat{z} \leftarrow$ FORWARD$(s, \hat{\theta})$*. Then there exists a constant $Q > 0$ such that*

$$\left| g_i - \frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right| \leq Q \sum_{i=1}^{n} \left| \theta_i - \hat{\theta}_i \right| |\eta_i - \mu_i|.$$

Lemma 9. *Suppose Assumption 3 and 5 holds. If $q \geq Q$ then $(s, \hat{\theta}, \textbf{status}) \leftarrow$* FIX-DEGENERACY$(s, \theta, q)$ *has **status** $=$* SUCCESS.

Proof. By Lines 21 to 27 of FIX-DEGENERACY we can see for **status** $=$ SUCCESS we need both

$$\left| g_i - \frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right| \leq q \omega_0, \forall i \tag{28}$$

and

$$\psi_0(s, \hat{\theta}) \leq \psi_0(s, \theta) - v. \tag{29}$$

We establish each of these in turn. First observe that (28)holds immediately by Assumption 5 and definition of ω_0 in line 20 of FIX-DEGENERACY.

Next we show (29) by bounding each of the right hand side terms in the equality

$$\psi_0(s, \hat{\theta}) - \psi_0(s, \theta) = \sum_{i=1}^{n} \psi_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) - \psi_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n}). \tag{30}$$

Recall the definition of p_i, ω_i, and $\hat{\theta}_i$ from FIX-DEGENERACY. If $p_i > 2q(\omega_i + \eta_i - \mu_i)$ then

$$
\begin{aligned}
&\psi_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) - \psi_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n}) \\
&= \int_{\theta_i}^{\hat{\theta}_i} \frac{\partial \psi_0(s, \theta_{1:i-1}, \gamma, \hat{\theta}_{i+1:n})}{\partial \gamma} \partial \gamma \\
&= \int_{\theta_i}^{\hat{\theta}_i} \frac{\partial \psi_i(s, z_{1:i-1}, t, \hat{\theta}_{i+1:n})}{\partial t} (\eta_i - \mu_i) \partial t && \text{by (6c)} \\
&= \int_{\theta_i}^{\hat{\theta}_i} g_i(\eta_i - \mu_i) \partial t + \int_{\theta_i}^{\hat{\theta}_i} \left(\frac{\partial \psi_i(s, z_{1:i-1}, t, \hat{\theta}_{i+1:n})}{\partial t} - g_i \right) (\eta_i - \mu_i) \partial t \\
&\leq g_i(\eta_i - \mu_i) \int_{\theta_i}^{\hat{\theta}_i} \partial t + Q \left(\sum_{j=i+1}^{n} \left| \theta_j - \hat{\theta}_j \right| (\eta_j - \mu_j) \right) (\eta_i - \mu_i) \left| \int_{\theta_i}^{\hat{\theta}_i} \partial t \right| && \text{by Assumption 5} \\
&= g_i(\eta_i - \mu_i) \int_{\theta_i}^{\hat{\theta}_i} \partial t + Q \omega_i (\eta_i - \mu_i) \left| \int_{\theta_i}^{\hat{\theta}_i} \partial t \right| && \text{by definition of } \omega_i \\
&= g_i(\eta_i - \mu_i)(\hat{\theta}_i - \theta_i) + Q \omega_i (\eta_i - \mu_i) \left| \hat{\theta}_i - \theta_i \right| \\
&\leq g_i(\eta_i - \mu_i)(\hat{\theta}_i - \theta_i) + q \omega_i (\eta_i - \mu_i) \left| \hat{\theta}_i - \theta_i \right| && \text{by } Q \leq q \\
&\leq g_i(\eta_i - \mu_i)(\hat{\theta}_i - \theta_i) + q \omega_i (\eta_i - \mu_i) && \text{as } \theta_i, \hat{\theta}_i \in [0, 1] \\
&= (\eta_i - \mu_i)(q \omega_i - p_i) && \text{by definition of } p_i \text{ and } \hat{\theta}_i \\
&\leq -\frac{p_i(\eta_i - \mu_i)}{2}, && (31)
\end{aligned}
$$

where the last inequality uses $p_i > 2q\omega_i$.

If $p_i \leq 2q(\omega_i + \eta_i - \mu_i)$ then

$$
\psi_0(s, \theta_{1:i-1}, \hat{\theta}_{i:n}) - \psi_0(s, \theta_{1:i}, \hat{\theta}_{i+1:n}) = 0 \tag{32}
$$

Therefore, by (30), (31), (32), and definition of v we establish (29). □

Assumption 6. *Denote $z = \text{FORWARD}(s, \theta)$ and $\hat{z} = \text{FORWARD}(s, \hat{\theta})$. There exists a constant $P > 0$ such that*

$$
|\eta_i - \mu_i - (\hat{\eta}_i - \hat{\mu}_i)| \leq P \sum_{i=1}^{n} \left| \theta_i - \hat{\theta}_i \right| |\eta_i - \mu_i|,
$$

$\forall s \in S, \forall \theta, \hat{\theta} \in [0, 1]^n$, $\forall i \in \{1, \dots, n\}$ *with* $\hat{\eta}_i := \eta_i(s, \hat{z}_{1:i-1})$ *and* $\hat{\mu}_i := \mu(s, \hat{z}_{1:i-1})$.

Lemma 10. *Suppose that Assumption 3 and 6 hold. Let $(s, \hat{\theta}, \textbf{status}) \leftarrow \text{FIX-DEGENERACY}(s, \theta, q)$, and $\hat{z} = \text{FORWARD}(s, \hat{\theta})$. If **status** $=$ SUCCESS then*

$$
\hat{\theta}_i \left(\frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right)^{+} + (1 - \hat{\theta}_i) \left(\frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right)^{-} \leq q((3 + P)\omega_0 + \hat{\eta}_i - \hat{\mu}_i) \tag{33}
$$

and $\omega_0 \leq \sqrt{2v/q}$.

Proof. Denote $\hat{\psi}_i := \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})$. From Line 23 of SAFE-PSI-MINIMIZATION

$$
\left| g_i - \frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right| \leq q\omega_0. \tag{34}
$$

Therefore,

$$
\hat{\theta}_i \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^{+} + (1 - \hat{\theta}_i) \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^{-} \leq g_i^{+} \hat{\theta}_i + g_i^{-}(1 - \hat{\theta}_i) + q\omega_0 \tag{35}
$$

where the inequality uses (34) and $\hat{\theta}_i \in [0,1]$. From Line 12-15 if $p_i > 2q(\omega_i + \eta_i - \mu_i)$ then

$$g_i^+ \hat{\theta}_i + g_i^-(1-\hat{\theta}_i) = 0 \Rightarrow \hat{\theta}_i \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^+ + (1-\hat{\theta}_i) \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^- \leq q\omega_0$$

where the implication uses (35). On the other hand, if

$$g_i^+ \theta_i + g_i^-(1-\theta_i) = p_i \leq 2q(\omega_i + \eta_i - \mu_i)$$

then by $\hat{\theta}_i = \theta_i$ and (35),

$$\hat{\theta}_i \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^+ + (1-\hat{\theta}_i) \left(\frac{\partial \hat{\psi}_i}{\partial \hat{z}_i} \right)^- \leq p_i + q\omega_0 = q(3\omega_0 + \eta_i - \mu_i). \tag{36}$$

Furthermore, by Assumption 6

$$|\eta_i - \mu_i - (\hat{\eta}_i - \hat{\mu}_i)| \leq P\omega_0$$

which combined with (36) yields (33).

It remains to show that $\omega_0 \leq \sqrt{2v/q}$. Note that

$$\eta_i - \mu_i \geq \left| \theta_i - \hat{\theta}_i \right| (\eta_i - \mu_i) = \omega_{i-1} - \omega_i \tag{37}$$

where the inequality uses $\theta_i, \hat{\theta}_i \in [0,1]$. Define $\mathcal{I} := \{ i \in \{1, \dots, n\} : \theta_i \neq \hat{\theta}_i \}$. Finally,

$$\begin{aligned}
v &= \frac{1}{2} \sum_{i \in \mathcal{I}} (\eta_i - \mu_i) p_i && \text{by definition of } v \\
&\geq \frac{1}{2} \sum_{i \in \mathcal{I}} (\omega_{i-1} - \omega_i) p_i && \text{by (37)} \\
&> q \sum_{i \in \mathcal{I}} (\omega_{i-1} - \omega_i) \omega_{i-1} && \text{by } p_i > 2q(\omega_i + \eta_i - \mu_i) > 2\omega_i \text{ for } i \in \mathcal{I} \\
&= q \sum_{i=1}^{n} (\omega_{i-1} - \omega_i) \omega_{i-1} && \text{by } \omega_{i-1} - \omega_i \text{ for } i \notin \mathcal{I} \\
&= q \sum_{i=1}^{n} \sum_{j=i}^{n} (\omega_{i-1} - \omega_i)(\omega_{j-1} - \omega_j) \\
&\geq \frac{q(\omega_0 - \omega_n)^2}{2} && \text{by Fact 3} \\
&= \frac{q\omega_0^2}{2} && \text{since } \omega_n = 0.
\end{aligned}$$

Rearranging this inequality gives the result. □

Fact 3. *Let* $\delta_1, \dots, \delta_n \in \mathbf{R}$ *then*

$$\left(\sum_{i=1}^{n} \delta_i \right)^2 \leq 2 \sum_{i=1}^{n} \sum_{j=i}^{n} \delta_i \delta_j.$$

Proof. It follows by

$$\begin{aligned}
\left(\sum_{i=1}^{n} \delta_i \right)^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \delta_j &= \sum_{i=1}^{n} \delta_i^2 + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \delta_i \delta_j + \sum_{i=1}^{n} \sum_{j=1}^{i-1} \delta_i \delta_j \\
&= \sum_{i=1}^{n} \delta_i^2 + 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} \delta_i \delta_j \\
&= -\sum_{i=1}^{n} \delta_i^2 + 2 \sum_{i=1}^{n} \sum_{j=i}^{n} \delta_i \delta_j.
\end{aligned}$$

□

Lemma 11. *Suppose that Assumption 3 and 6 hold. Let* $(s, \hat{\theta}, \textbf{\textit{status}}) \leftarrow \textsc{Fix-Degeneracy}(s, \theta, q)$*, and* $\hat{z} = \textsc{Forward}(s, \hat{\theta})$*. If* $\textbf{\textit{status}} = \text{SUCCESS}$ *then for all* $i \in \{1, \dots, n\}$*,*

$$\hat{r}_i \leq \sqrt{q} \left((3+P)\sqrt{v} + \max \left\{ \sqrt[4]{2L\hat{t}_{i,L}}, \sqrt{2\hat{t}_{i,L}} \right\} \right).$$

with $\hat{y}_i := \frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i}$*,* $\hat{r}_i := \hat{\theta}_i \hat{y}_i^+ + (1 - \hat{\theta}_i)\hat{y}_i^-$*, and* $\hat{t}_{i,L} := - \underset{\hat{\theta}_i + d_i \in [0,1]}{\textit{minimize}} \; \frac{\partial \psi_i(s, \hat{z}_{1:i}, \hat{\theta}_{i+1:n})}{\partial \hat{\theta}_i} d_i + \frac{L}{2} d_i^2$*.*

Proof. Define

$$\gamma_i := \begin{cases} \frac{1}{\hat{\eta}_i - \hat{\mu}_i} & \hat{\eta}_i - \hat{\mu}_i \neq 0 \\ \infty & \text{otherwise.} \end{cases}$$

$$a := \max \left\{ \sqrt{2L\hat{t}_{i,L}}, 2\hat{t}_{i,L} \right\}$$

$$b := q(3+P)\omega_0$$

$$c := q.$$

By Lemma 7, $\hat{r}_i \leq a\gamma_i$ and by Lemma 10, $\hat{r}_i \leq b + c/\gamma_i$. Therefore,

$$\hat{r}_i \leq \min\{a\gamma_i, b + c/\gamma_i\}.$$

Maximizing the upper bound with for $\gamma_i \in [0, \infty]$ gives

$$a\gamma_i = b + c/\gamma_i \Rightarrow a\gamma_i^2 - b\gamma_i - c = 0 \Rightarrow \gamma_i = \frac{b + \sqrt{b^2 + 4ac}}{2a}$$

where the second implication uses the quadratic formula and $\gamma_i \geq 0$. Plugging this back in gives

$$\hat{r}_i \leq \frac{b + \sqrt{b^2 + 4ac}}{2} \Rightarrow \hat{r}_i \leq b + \sqrt{ac}.$$

Therefore, using $\omega_0 \leq \sqrt{2v/q}$ from Lemma 10 and the definition of a, b, c we get

$$\hat{r}_i \leq (3+P)\sqrt{2vq} + \sqrt{q \max \left\{ \sqrt{2L\hat{t}_{i,L}}, 2\hat{t}_{i,L} \right\}}.$$

□

Lemma 12. *Suppose that Assumption 2, 3, 4, and 6 holds. Define* $\tau := \delta_L(s, \hat{\theta}) + v$*. If*

$$\tau \leq \min \left\{ 2L, \frac{2L}{n(3+P)^4}, \frac{4D_s^4 L^2}{c^4} \right\} \tag{38}$$

then

$$\Delta(s, \hat{\theta})^4 \leq 96c^4 q^2 L\tau.$$

Proof. First observe that

$$\begin{aligned} \|r\|_2^4 &\leq q^2 \left((3+P)^4 n v^2 + 4\delta_L(s, \hat{\theta})^2 + 2L\delta_L(s, \hat{\theta}) \right) \\ &\leq q^2 \left((3+P)^4 n \tau^2 + 4\tau^2 + 2L\tau \right) \\ &\leq 6q^2 L\tau \end{aligned} \tag{39}$$

where the first inequality uses Lemma 11, the second inequality uses the definition of τ, and the third inequality uses (38).

Next,

$$\begin{aligned} \Delta(s, \hat{\theta}) &\leq D_s \sqrt{2L\delta_L(s, \hat{\theta})} + c\|r\|_2 \\ &\leq D_s\sqrt{2L\tau} + c\sqrt[4]{6q^2 L\tau} \\ &\leq 2c\sqrt[4]{6q^2 L\tau} \end{aligned}$$

where the first inequality uses Lemma 6 and Assumption 4, the second inequality uses (39), and the third uses (38). □

Theorem 4. *Suppose that Assumption 2, 3, 4, 5, and 6 hold. Define*

$$\zeta_2 = 96c^4q^2L$$

$$\zeta_3 = \frac{\Delta(s^1,\theta^1)}{\min\left\{2L, \frac{2L}{n(3+B)^4}, \frac{4D_s^4L^2}{c^4}\right\}}$$

where L is the smoothness constant for ψ_0. Then for $k > \zeta_3$ SAFE-PSI-MINIMIZATION *satisfies*

$$\Delta(s^k,\theta^k) \leq \left(\frac{\zeta_2}{k-\zeta_3}\right)^{1/3}$$

Proof. Follows by Lemma 12 and Lemma 8 with $\zeta_1 = 3$. □

D.1 Proof of Theorem 3

Proof. Assumption 1 and 2 imply that f, η_i, and μ_i are β-smooth and B-Lipschitz for some constant $\beta, B > 0$. Since μ_i and η_i are Lipschitz Assumption 4 holds. Lemma 14 implies Assumption 5 holds. Corollary 1 implies Assumption 6 holds. Note that Corollary 1 and Lemma 14 appear in Section E

With Assumption 5, and 6 established, the result holds by Theorem 4. □

E Justifying assumptions

The purpose of this section is to show that if Assumptions 1 and 2 hold then Assumption 5 and 6 hold.

Definition 2. *A function $h : X \to \mathbf{R}$ is L-smooth with respect to $\|\cdot\|$ if $\|\nabla h(x) - \nabla h(x')\|_* \leq L\|x-x'\|$ for all $x, x' \in X$.*

Definition 3. *A function $h : X \to \mathbf{R}$ is B-Lipschitz with respect to $\|\cdot\|$ if $|h(x) - h(x')| \leq B\|x-x'\|$ for all $x, x' \in X$.*

Fact 4. *Suppose that $h : X \to \mathbf{R}$ differentiable and B-Lipschitz with respect to $\|\cdot\|$, then $\|\nabla h(x)\|_* \leq B$ for all $x \in X$.*

Fact 5. *Suppose that the function $h : X \to \mathbf{R}$ is smooth and X is bounded. Then (for any given norm) there exists constant B and β such that h is B-Lipschitz and β-smooth.*

E.1 Proof Assumption 6 holds

Lemma 13. *Suppose that $\eta_i - \mu_i$ is B-Lipschitz with respect to the ℓ_1-norm for $B > 0$. Let $z =$* FORWARD(s,θ) *and $\hat{z} =$* FORWARD$(\hat{s},\hat{\theta})$*. Then*

$$\|z-\hat{z}\|_1 \leq B^{-1}(1+B)^n \sum_{i=1}^{n} \left(\left|\theta_i - \hat{\theta}_i\right||\eta_i - \mu_i| + \|s-\hat{s}\|_1\right)$$

for all $\theta \in [0,1]^n$.

Proof. Denote $\eta_i(s, z_{1:i-1})$, $\mu_i(s, z_{1:i-1})$, $\eta_i(\hat{s}, \hat{z}_{1:i-1})$, $\mu_i(\hat{s}, \hat{z}_{1:i-1})$ by $\eta_i, \mu_i, \hat{\eta}_i$ and $\hat{\mu}_i$ respectively. Observe that

$$\begin{aligned}
|z_i - \hat{z}_i| &= \left|\theta_i\eta_i + (1-\theta_i)\mu_i - (\hat{\theta}_i\hat{\eta}_i + (1-\hat{\theta}_i)\hat{\mu}_i)\right| \\
&= \left|(\theta_i - \hat{\theta}_i)(\eta_i - \mu_i) + (1-\hat{\theta}_i)(\mu_i - \hat{\mu}_i) + \hat{\theta}_i(\eta_i - \hat{\eta}_i)\right| \\
&\leq \left|(\theta_i - \hat{\theta}_i)(\eta_i - \mu_i)\right| + |\mu_i - \hat{\mu}_i + \eta_i - \hat{\eta}_i| \\
&\leq \left|(\theta_i - \hat{\theta}_i)(\eta_i - \mu_i)\right| + B(\|s-\hat{s}\|_1 + \|z_{1:i-1} - \hat{z}_{1:i-1}\|_1). \qquad (40)
\end{aligned}$$

Applying (40) for $i = 1$ we have $|z_1 - \hat{z}_1| \leq \left|(\theta_i - \hat{\theta}_i)(\eta_i - \mu_i)\right| + B\|s-\hat{s}\|_1$. Suppose that,

$$\|z_{1:i} - \hat{z}_{1:i}\|_1 \leq \sum_{j=1}^{i}(B+1)^{i-j}\left(B\|s-\hat{s}\|_1 + \left|(\theta_j - \hat{\theta}_j)(\eta_j - \mu_j)\right|\right)$$

then by (40) we deduce

$$\|z_{1:i+1} - \hat{z}_{1:i+1}\|_1 \le \sum_{j=1}^{i+1} (B+1)^{1+i-j} \left(B\|s - \hat{s}\|_1 + \left|(\theta_j - \hat{\theta}_j)(\eta_j - \mu_j)\right| \right).$$

By induction and the fact $\sum_{j=1}^{n}(1+B)^{n-j} \le B(B+1)^n$, the result holds. □

Corollary 1. *Suppose that $\eta_i - \mu_i$ is B-Lipschitz with respect to the ℓ_1-norm for $B > 0$. Let $z = \text{FORWARD}(s, \theta)$ and $\hat{z} = \text{FORWARD}(\hat{s}, \hat{\theta})$. Then*

$$|\eta_i - \mu_i - (\eta_i(\hat{s}, \hat{z}_{1:i-1}) - \mu_i(\hat{s}, \hat{z}_{1:i-1}))| \le \beta\|s - \hat{s}\|_1 + \beta B^{-1}(1+B)^n \sum_{i=1}^{n} \left(\left|\theta_i - \hat{\theta}_i\right| |\eta_i - \mu_i| + \|s - \hat{s}\|_1 \right)$$

for all $\theta \in [0,1]^n$.

Proof. Follows from the assumption $\eta_i - \mu_i$ is β-smooth and Lemma 13. □

Therefore, if Assumptions 1 and 2 hold then, by Fact 5 and Corollary 1, we conclude Assumption 6 holds.

E.2 Proof Assumption 5 holds

Fact 6. *Let $a \in \mathbf{R}^n, b \in \mathbf{R}^{n+1}$. Suppose*

$$a_i := b_{n+1} + \sum_{j=i+1}^{n} a_j b_j,$$

then $\|a\|_1 \le (1 + \|b\|_\infty)^n$.

Proof. Note that

$$|a_i| \le \|b\|_\infty + \|a_{i+1:n}\|_1 \|b\|_\infty \Rightarrow \|a_{i:n}\|_1 \le \|b\|_\infty + \|a_{i+1:n}\|_1(1 + \|b\|_\infty).$$

Therefore, if

$$\|a_{i:n}\|_1 \le \sum_{j=1}^{n-i} \|b\|_\infty (1 + \|b\|_\infty)^j. \tag{41}$$

then

$$\|a_{i-1:n}\|_1 \le \|b\|_\infty + \sum_{j=1}^{n-i} \|b\|_\infty (1 + \|b\|_\infty)^{j+1} = \sum_{j=1}^{n-(i-1)} \|b\|_\infty (1 + \|b\|_\infty)^j.$$

Since (41) holds for $i = n$ by induction (41) holds for all i. By the bound on the sum of a geometric series the result holds. □

Fact 7. *Let $a, c \in \mathbf{R}^n, b \in \mathbf{R}^{n \times n}$. Suppose*

$$a_i := c_i + \sum_{j=i+1}^{n} a_j b_{i,j}, \quad \hat{a}_i := \hat{c}_i + \sum_{j=i+1}^{n} \hat{a}_j \hat{b}_{i,j},$$

then

$$\|a - \hat{a}\|_1 \le \left(\|c - \hat{c}\|_\infty + (1 + \|\hat{b}\|_\infty)^n \|\hat{b} - b\|_\infty \right) \frac{(1 + \|b\|_\infty)^n}{\|b\|_\infty}.$$

Proof. Note that

$$\begin{aligned} |a_i - \hat{a}_i| &= \left| c_i - \hat{c}_i + \sum_{j=i+1}^{n} (a_j b_{i,j} - \hat{a}_j \hat{b}_{i,j}) \right| \\ &= \left| c_i - \hat{c}_i + \sum_{j=i+1}^{n} ((a_j - \hat{a}_j) b_j - \hat{a}_j (\hat{b}_{i,j} - b_{i,j})) \right| \\ &\le \|a_{i+1:n} - \hat{a}_{i+1:n}\|_1 \|b\|_\infty + \|c - \hat{c}\|_\infty + \|\hat{a}\|_1 \|\hat{b} - b\|_\infty \\ &\le \|a_{i+1:n} - \hat{a}_{i+1:n}\|_1 \|b\|_\infty + \|c - \hat{c}\|_\infty + (1 + \|\hat{b}\|_\infty)^n \|\hat{b} - b\|_\infty \end{aligned}$$

where the last inequality uses Fact 6. By induction,

$$\|a_{i:n} - \hat{a}_{i:n}\|_1 \leq \left(\|c - \hat{c}\|_\infty + (1 + \|\hat{b}\|_\infty)^n \|\hat{b} - b\|_\infty\right) \sum_{j=1}^{n-i} (1 + \|b\|_\infty)^j.$$

By the bound on the sum of a geometric series the result holds. □

Lemma 14. *Suppose f, η_i, and μ_i are β-smooth functions with respect to the ℓ_1-norm. Also, suppose f, η_i and μ_i are B-Lipschitz with respect to the ℓ_1-norm and $\hat{\theta}, \theta \in [0,1]^n$. Let $z \leftarrow \textsc{Forward}(s, \theta)$,*

$$g_i := \frac{\partial f}{\partial z_i} + \sum_{j=i+1}^{n} g_j \left(\hat{\theta}_j \frac{\partial \eta_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \mu_j}{\partial z_i} \right),$$

and $\hat{z} \leftarrow \textsc{Forward}(s, \hat{\theta})$. Then there exists $Q \leq 2\beta B^{-2}(1+B)^{3n}$ s.t.

$$\sum_{i=1}^{n} \left| g_i - \frac{\partial \psi_i(s, \hat{z}_{1:i-1}, \hat{\theta}_{i+1:n})}{\partial \hat{z}_i} \right| \leq Q \sum_{i=1}^{n} \left| \theta_i - \hat{\theta}_i \right| |\eta_i - \mu_i|.$$

Proof. Denote $\eta_i(s, z_{1:i-1})$, $\mu_i(s, z_{1:i-1})$, $\eta_i(s, \hat{z}_{1:i-1})$, $\mu_i(s, \hat{z}_{1:i-1})$, $\psi_i(s, \hat{z}_{1:i-1}, \hat{\theta}_{i+1:n})$ by $\eta_i, \mu_i, \hat{\eta}_i$, $\hat{\mu}_i$, and $\hat{\psi}_i$ respectively. This proof is based on Fact 7. Define,

$$a_i = \frac{\partial \psi_i}{\partial z_i}, \quad b_{i,j} = \hat{\theta}_j \frac{\partial \eta_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \mu_j}{\partial z_i}, \quad c_i = \frac{\partial f}{\partial z_j}$$
$$\hat{a}_i = \frac{\partial \hat{\psi}_i}{\partial \hat{z}_i}, \quad \hat{b}_{i,j} = \hat{\theta}_j \frac{\partial \hat{\eta}_j}{\partial z_i} + (1 - \hat{\theta}_j) \frac{\partial \hat{\mu}_j}{\partial z_i}, \quad \hat{c}_i = \frac{\partial \hat{f}}{\partial z_j}.$$

Moreover, observe that by our assumptions

$$\left| b_{i,j} - \hat{b}_{i,j} \right| \leq \hat{\theta}_j \left| \frac{\partial \eta_j}{\partial z_i} - \frac{\partial \hat{\eta}_j}{\partial \hat{z}_i} \right| + (1 - \hat{\theta}_j) \left| \frac{\partial \mu_j}{\partial z_i} - \frac{\partial \hat{\mu}_j}{\partial \hat{z}_i} \right| \leq \beta \|z - \hat{z}\|_1 \tag{42a}$$

$$\|c - \hat{c}\|_\infty \leq \beta \|z - \hat{z}\|_1 \tag{42b}$$

$$\max\{\|b\|_\infty, \|\hat{b}\|_\infty\} \leq B. \tag{42c}$$

Therefore,

$$\begin{aligned} \|a - \hat{a}\|_1 &\leq (\beta \|z - \hat{z}\|_1 + (1+B)^n \beta \|z - \hat{z}\|_1) \frac{(1+B)^n}{B} \\ &\leq 2\beta B^{-1} (1+B)^{2n} \|z - \hat{z}\|_1 \\ &\leq 2\beta B^{-2} (1+B)^{3n} \sum_{i=1}^{n} \left| \theta_i - \hat{\theta}_i \right| |\eta_i - \mu_i| \end{aligned}$$

where the first inequality uses (42) and Fact 7, the second inequality uses $(1+B)^n \geq 1$, and the third uses Lemma 13. □

Therefore, if Assumptions 1 and 2 hold then, by Fact 5 and Lemma 14, we conclude Assumption 5 holds.

F Implementation details

The fastest known way to derive bounds on the values taken by neural network is Interval Bound Propagation (IBP) [25]. It does not attempt to solve the optimization problem of the type of (1) which we define in Appendix A.2 but simply performs interval analysis to derive loose bounds on the values taken by the neural network. As a result, running IBP is a very fast procedure, whose cost is analogous to performing a forward pass through the network, but will give much looser bounds than solving the optimization problem. We include it in Table 1 as a lower bound on the runtime that can be achieved by any of the method. We also use it to derive the intermediate bounds l_i and u_i required for the definition of the constraints (14a) when formulating the optimization problems for the methods we are benchmarking.

For DeepVerify [6], we use the code released by the authors. The Lagrangian dual is optimized using a sub-gradient method. We use the Adam optimizer [46], starting with an initial step size of 10^{-2} and decreasing it by a factor of 10 every 100 steps. We run the algorithm for a maximum of 500 steps, but allow early stopping if the dual objective value reaches the value returned by the dual objective of our NonConvex reformulation. In its original formulation, DeepVerify does not have access to a primal objective and can therefore not be provided with an appropriate stopping criterion. We provide it here with one in order to generate a more fair comparison,

Our NonConvex relaxation is solved following the principle of SIMPLE-PSI-MINIMIZATION. To optimize the objective function (line 10 of Algorithm 1), we use the FISTA with backtracking linesearch [19] algorithm. We initialize our step size to a high value of 100, and at each optimization step, perform a line-search starting from the previous step-size, progressively shrinking it by a backtracking coefficient of 0.8 until sufficient progress can be guaranteed. In order to prevent the use of too small learning rates, we increase the step-size by a factor of 1.5 when no backtracking was necessary, and clip the step size to a minimum of 10^{-5}. We run a maximum of 50 steps of this algorithm, unless we first reach a situation where the relative gap between the primal and the dual objective becomes smaller than 10^{-2}.

For the CVXPY [28] based solvers, we used the default settings.

G Benchmarking on Robustly trained Networks

In addition to the results reported in Section 3, we provide additional results for networks trained with Interval Bound Propagation (IBP) [25], to achieve robustness against $\mathcal{L}_\infty$-bounded perturbations of size $\epsilon = 8.0/255$. Because of the strong regularization effect of IBP, there is very little difference between the solution to the optimization problem described in Section A.2 and the bounds produced by IBP, so better optimization algorithms do not prove as useful.

For these experiments, we didn't employ the relative duality gap criterion and instead optimized until the duality gap reached a value smaller than 1e-4.

Even on those networks for which the solution should be easier to obtain, we observe similar results. Off-the-shelf solvers based on CVXPY do not scale beyond the Tiny network, and the runtime for DeepVerify is significantly larger than for our nonconvex solver, as shown in Table 3. The discrepancy in runtime comes from the fact that DeepVerify does not manage to converge to an accurate enough solution, as shown by the low percentage of early stopping in Table 4 and the observable plateaus in Figure 7a.

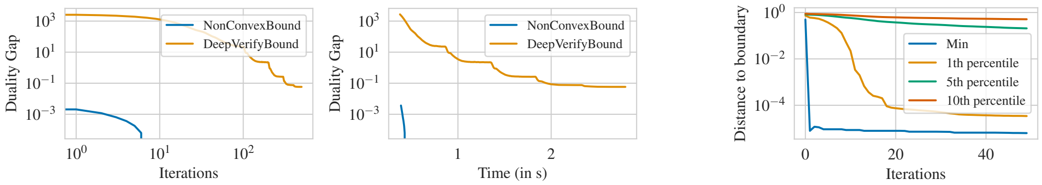
We note that as opposed to networks trained with adversarial training, Figure 7b reveals that when computing bounds on networks trained to achieve certifiable robustness, activations do get close to potentially degenerate points, which might be problematic for the optimization procedure and require us to employ Algorithm 3 to find optimal solutions.

Methods	ReLU activation			SoftPlus activation		
	Tiny	Small	Medium	Tiny	Small	Medium
IBP [25]	3.4	2.8	3.2	3.9	2.3	2.8
DeepVerify [6]	338	658	1171	403	842	1620
NonConvex (ours)	93	176	297	6.1	61	210
Solver (SCS) [26]	$78.4 \cdot 10^3$	-	-	$32.1 \cdot 10^4$	-	-
Solver (ECOS) [27]	$41.2 \cdot 10^3$	-	-	-	-	-

Table 3: Runtime of bound computation, in milliseconds. Results correspond to the computation of a lower bound on the gap between ground truth and all other classes for a network subject to an L_∞ adversarial attack with budget $\frac{8.0}{255}$, averaged over the CIFAR-10 test set, on network trained with IBP.

	Proportion of early stopping		Average number of iterations	
	DeepVerify	NonConvex	DeepVerify	NonConvex
Tiny ReLU	92%	99%	242	6.5
Small ReLU	25%	99%	479	5.5
Medium ReLU	3%	100%	491	4.92
Tiny SoftPlus	91 %	100%	238.33	7.4
Small SoftPlus	0%	99%	500	8.5
Medium SoftPlus	0%	98%	500	9.8

Table 4: Proportion of bound computations on CIFAR-10 where the algorithm converges within the iteration budget, and average number of iterations for each first order algorithm for an IBP trained network.



(a) Evolution of the duality gap as a function of time or number of iteration, for the NonConvex and DeepVerify Solver.

(b) Distribution of distance to potentially degenerate point.

Figure 7: Evaluation on the Medium-sized network with SoftPlus activation function trained with IBP.