Journal of Machine Learning Research ? (2026) 1-28 Submitted 3/26; Revised ?; Published ?

A Different Perspective on Out-of-Sample Generalization

Tom Winckelman WINCKELMAN@TAMU.EDU
Department of Mathematics
Mailstop 3368
Texas A&M University
College Station, TX 77843-3368, USA

Simon Foucart FOUCART@TAMU.EDU
Department of Mathematics
Mailstop 3368
Texas A&M University
College Station, TX 77843-3368, USA

Editor: ?

Abstract

We propose a theoretical framework for designing and analyzing neural networks in regression tasks. The crux is that, for certain families of functions, we can mathematically relate their error on arbitrary unseen data to their error on a training data set, via a natural error bound which is novel in the context of neural networks. We present a complete characterization of which families of functions satisfy the error bound in question, allowing us to study a relatively tractable equivalent condition. Thereby, we confirm that conventional ReLU networks do not satisfy this bound, which is consistent with the empirically observed phenomenon of overfitting.

In the univariate case, we show that the bound in question can be recovered by imposing simple inequality constraints on the parameters of a ReLU network. This results in neural networks whose error on unseen data is properly controlled by the training error. We also develop an algorithm that fits such a network to the data by solving a semidefinite program encoding the constraints. In particular, the algorithm does not have hyperparameters in the typical sense and produces a model that generalizes certifiably well. This study should be viewed as prerequisite for the more impactful multivariate setting.

Keywords: unisolvence, optimal recovery, generalization, quadratic programs, duality

1 Introduction

Why is it that, for some models like ordinary least squares, we can mathematically relate performance on new data to performance on a data set used to fit the model, meanwhile for other models like ReLU neural networks, we cannot? We explore one possible explanation.

With some freedom to choose $x_1, \dots, x_m \in \mathbb{R}^d$, we will observe $y_1, \dots, y_m \in \mathbb{R}$ satisfying $y_i \approx f(x_i)$, where f is an unknown continuous function. Because f is unknown, our job is to produce some relatively simple continuous function S as a stand-in for it. Suppose we want $S(x) \approx f(x)$ whenever x belongs to a set $\mathcal{X} \subset \mathbb{R}^d$ of "realistic" possible inputs (with $x_1, \dots, x_m \in \mathcal{X}$). For the error $|f(x) - S(x)|$ to be small at *any* $x \in \mathcal{X}$ (not necessarily

related to $x_1, \ldots, x_m$, beyond the fact that they are all in $\mathcal{X}$), we would require the term

$$\|f - S\|_\infty \;\coloneqq\; \sup_{x \in \mathcal{X}} |f(x) - S(x)|$$

to be small. We take $\|f - S\|_\infty$ as our definition of the error on unseen data, which we call the "generalization error," even though this terminology usually refers to a much weaker notion of error—an $L^2(\mathcal{X}, \mathbb{P})$ norm. Based on only $x_1, \ldots, x_m \in \mathcal{X}$ and $y_1, \ldots, y_m \in \mathbb{R}$, we aim to design an algorithm that selects S from some class of simple functions, certified by an assurance that $\|f - S\|_\infty$ will be small (see section 6 for our example of this).

Let Σ denote the set of functions that we would consider fitting to the data. Abstractly, this could be any subset of $C(\mathcal{X})$. However, concrete examples include the set of polynomials of degree at most 20, or the set of functions expressible as a shallow width 20 ReLU network. On one hand, $\|f - S\|_\infty \geq \inf_{Z \in \Sigma} \|f - Z\|_\infty$. This lower bound is the "approximation error." On the other hand, we introduce a relatively simple quantity $\eta \in [1, \infty]$ (see Lemma 1) depending on $\mathcal{X}$, $x_1, \ldots, x_m$, and Σ such that, for *all* $f \in C(\mathcal{X})$, $S \in \Sigma$, and $y_1, \ldots, y_m \in \mathbb{R}$,

$$\|f - S\|_\infty \;\leq\; (1 + \eta) \inf_{Z \in \Sigma} \|f - Z\|_\infty \;+\; \eta \left(\max_{i=1}^m \big|y_i - S(x_i)\big| \;+\; \max_{i=1}^m |y_i - f(x_i)| \right). \quad (1)$$

If developed carefully, this conveys a powerful assurance that $\|f - S\|_\infty$ is small. We say "carefully" because there are innumerable ways to cheat. For instance, (1) becomes easier to establish if the domain $\mathcal{X}$ is unrealistically small (such as a union of small balls containing $x_1, \ldots, x_m$). Most importantly, for (1) to be substantial, we require that the approximation error is small (which is necessary, anyway) and η is not too big (ideally, $\eta < 10$). The "measurement error" $y_i \approx f(x_i)$ should of course also be small. Plus, in order to be useful in practice, it is important to have an algorithm for producing an $S \in \Sigma$ with small "training error," i.e., $y_i \approx S(x_i)$, which is the only place that S enters the upper bound on $\|f - S\|_\infty$.

We wish to emphasize the design of Σ. One one hand, enlarging Σ reduces the approximation error, and Σ must be large enough that the approximation error is small for a good variety of functions f. On the other hand, we shall see that η is an increasing function of Σ (see Theorem 4), which means that (1) can easily cease to be meaningful if Σ is too large. The extreme cases are $\Sigma = \{0\}$ and $\Sigma = C(\mathcal{X})$, with η being infinite in the latter case. Kernel regression—with kernel $\mathrm{kern}(x, y)$—provides our first non-trivial example:

$$\Sigma_{\mathrm{kern}} \;\coloneqq\; \mathrm{Span}\{\mathrm{kern}(\,\cdot\,, x_1), \ldots, \mathrm{kern}(\,\cdot\,, x_m)\}.$$

For Σ_{kern}, the value of η appearing in (1) is finite. This can be verified using Theorem 4, presented below. Analogously—if not necessitated by the Mairhuber-Curtis theorem—we intend to choose Σ depending on $x_1, \ldots, x_m$.

Vector spaces of dimension at most m, including Σ_{kern}, are often considered too small to offer satisfactory approximation error. This is one of the main sentiments that has led to neural networks' surge in popularity over kernel-based methods. Yet, bounds such as (1) are mostly unknown outside of cases when Σ is a vector sub-space of $C(\mathcal{X})$. In this paper, we develop one in which Σ is built using a recognizable ReLU neural network architecture.

2 Relation to Other Work

In statistical learning theory, Σ is called the hypothesis class. Very roughly speaking, if a hypothesis class is "PAC-learnable," $x_1, \ldots, x_m$ is a random sample from a distribution $\mathbb{P}$,

and $y_i = f(x_i) + \varepsilon_i$ where $\varepsilon_1, \ldots, \varepsilon_m$ are likewise random, then a bound similar to (1) holds with probability $1 - \delta$. However, the constants are different, the ∞-norms are replaced by 2-norms, and an additional term, usually of the form $D\sqrt{\ln(1/\delta)/m}$, is added to the upper bound. The first key difference is that (1) is a purely deterministic statement. Second, whereas an $L^2(\mathcal{X}, \mathbb{P})$ norm measures error "within sample," the $C(\mathcal{X})$ norm measures error even at adversarially chosen inputs. Third, removing the additional term is crucial, as the constant D is often proportional to the VC dimension of Σ and, thus, too large [1] to be controlled by the term "$\sqrt{\cdots/m}$." In contrast, we prove that η can be independent of the number of degrees of freedom in a regression model.

There exists a plethora of methods intended to bolster the robustness of neural networks. However—to the authors' knowledge—none feature a theoretical assurance as strong as a bound on $\|f - S\|_\infty$, and all rely on empirical techniques such as hyperparameter tuning. Many of these are stunningly effective, empirically. The authors are not data scientists by discipline, but we are familiar with some of the most popular such methods, including ensemble approaches and Bayesian neural networks. In this paper, our goal is to minimize our reliance on empiricism and explore whether mathematical alternatives are possible. So, our methodology is not comparable to these approaches, although our goals are very aligned.

There is literature on "robustness verification" for neural networks, which tests whether or not a model's output is stable with respect to its input, such as Zhang et al. (2018); Wang et al. (2021). Given $x_o \in \mathbb{R}^d$ and $\varepsilon_o > 0$, this is typically done by producing upper and lower bounds on $S(x)$ that are valid when $\|x - x_o\| \leq \varepsilon_o$. Our goal is subtly different. Rather than scrutinizing the *volatility* of S, we instead wish to scrutinize the *error*. Of course, if S is has small error at, say, x_i—and is not too volatile with respect to small deviations in input—then indeed the error remains small for inputs close to x_i, as well. However, this only assures small error on $\mathcal{X} = \bigcup_j B(x_j, \varepsilon_j)$. In exchange for reduced model flexibility, we aim to provide a much more powerful assurance, such as controlling the error on a less porous set $\mathcal{X}$. Unlike the literature on model verification, this present work does *not* accept general user-supplied models (see section 4, in particular). Instead, we explore highly specific restrictions needed to assure a model's performance at unfamiliar inputs. Similarly, we would be perfectly willing to compromise on all typical benchmarks in order to achieve this goal, but the theory is not developed enough to have reached such a crossroads. [2]

This paper adheres to the mathematical field of optimal recovery—only we have minimized notation and terminology. For an introduction to the early perspective in this field, see Micchelli and Rivlin (1977). In this paper, we extend aspects of DeVore et al. (2019), which makes the assumption that Σ is a vector space. In our view, this is a crucial limitation of known results in optimal recovery. For inspiration on how to remove this assumption, we are interested in analogies to compressive sensing, as noted in DeVore et al. (2019), but here we instead explore the assumption $\mathcal{X} \subset \mathbb{R}$. This still leaves the most impactful cases open (viz. $\mathcal{X} \subset \mathbb{R}^d$), but there is at least novelty in conceding a different point.

We initially anticipated connections with the very interesting field of mathematical super-resolution which, for example, Poon et al. (2023) have applied to neural networks.

1. Intuitively, the VC dimension can be thought of as the number of degrees of freedom required to describe a function belonging to Σ. See Bartlett et al. (2019) for a discussion in the context of neural networks.
2. The "boutique" training methods, which we develop in section 6, seem to almost completely compensate for our prototype's reduced approximation power in comparison to its conventional counterparts.

However, as with "PAC-learnability," this paper's key concept turns out to be fundamentally different than the key concept of "separation" in super-resolution (see Proposition 9).

Finally, we rely on connections between ReLU networks and classical spline theory. For instance, we employ a special case of the celebrated Schoenberg-Whitney theorem to prove one of our main results. We point out that connections with spline theory have been explored by other authors, as well, such as Balestriero and Baraniuk (2018).

3 Unisolvence and Other Key Theoretical Concepts

We are given distinct points $x_1, \ldots, x_m$ belonging to a subset $\mathcal{X}$ of $\mathbb{R}^d$. Let us examine which subsets Σ of $C(\mathcal{X})$ satisfy the property that, for some $M \geq 1$, for *any* possible "target" values $y_1, \ldots, y_m \in \mathbb{R}$, *all* $S \in \Sigma$ and *all* $f \in C(\mathcal{X})$, we have

$$\underbrace{\|f - S\|_\infty}_{\substack{\text{generalization} \\ \text{error}}} \leq M \left(\underbrace{\inf_{Z \in \Sigma} \|f - Z\|_\infty}_{\substack{\text{approximation} \\ \text{error}}} + \underbrace{\max_{i=1}^m |y_i - S(x_i)|}_{\text{training error}} + \underbrace{\max_{i=1}^m |y_i - f(x_i)|}_{\substack{\text{measurement} \\ \text{error}}} \right). \tag{2}$$

In order to prove (2), of course, Σ must depend on $\mathcal{X}$, since it is a subset of $C(\mathcal{X})$. Moreover, we also allow Σ to potentially depend on $x_1, \ldots, x_m$. Most importantly, M is allowed to depend on Σ, $\mathcal{X}$, and $x_1, \ldots, x_m$. The number of function inputs (the "dimension" d) is present in $\mathcal{X}$, and so may enter into the constant M.

For the $C(\mathcal{X})$ norm, (2) is essentially the best type bound in general. Indeed, the inequality in (2) reverses in the ideal "noiseless" case that $y_i = f(x_i)$ for all i. In that case,

$$\begin{aligned} \|f - S\|_\infty &= \frac{1}{2}\|f - S\|_\infty + \frac{1}{2}\|f - S\|_\infty \\ &\geq \frac{1}{2} \inf_{Z \in \Sigma} \|f - Z\|_\infty + \frac{1}{2} \max_{i=1}^m |f(x_i) - S(x_i)| \\ &= \frac{1}{2} \left(\inf_{Z \in \Sigma} \|f - Z\|_\infty + \max_{i=1}^m |y_i - S(x_i)| + \max_{i=1}^m |y_i - f(x_i)| \right). \end{aligned}$$

In the introduction, we have given additional context for (1) that applies to (2), as well. Now, let us begin the analysis, starting with a relatively simple necessary condition that serves as a conceptual anchor.

Lemma 1 ***"Unisolvence" and $\eta < \infty$ are Necessary for*** (2)
If (2) *holds with $M < \infty$, then it must be the case that, for any $S, Z \in \Sigma$, $S(x_i) = Z(x_i)$ for all $i = 1, \ldots, m$ implies $S(x) = Z(x)$ for all $x \in \mathcal{X}$. In this case, M must also satisfy*

$$\eta \coloneqq \sup_{\substack{S, Z \in \Sigma \\ S \neq Z}} \frac{\|S - Z\|_\infty}{\max_i |S(x_i) - Z(x_i)|} \leq M.$$

Proof 1 We apply (2) by choosing $f = Z$ and the noiseless labels $y_i = Z(x_i)$. ■

Definition 2 ***Unisolvence with Respect to a Scattering of Points***
We call a set $\Sigma \subset C(\mathcal{X})$ ***unisolvent with respect to given points*** *$x_1, \ldots, x_m \in \mathcal{X}$ if the necessary condition from* Lemma 1 *holds, i.e., if no two* distinct *functions $S, Z \in \Sigma$ have $S(x_i) = Z(x_i)$ for all $i = 1, \ldots, m$. We may say "unisolvent in $C(\mathcal{X})$" for emphasis.*

Borrowed from the theory of the finite element method, the term "unisolvent" is just an "adjectivization" of the phrase "one solution," referring to the fact that the equation $y = \Lambda(S)$ has at most one solution $S \in \Sigma$. Unisolvence means that, based only on the observed values, we can fully determine the behavior of an element of Σ on the rest of $\mathcal{X}$.

Any singleton is trivially unisolvent. A doubleton $\{S, Z\}$ where $S \neq Z$ is unisolvent with respect to $x_1, \ldots, x_m$ *if and only if* $S(x_i) \neq Z(x_i)$ for some i. If $\mathcal{X} = \{x_1, \ldots, x_m\}$, then any set Σ is unisolvent. Here are some more substantial examples.

Proposition 3 ***Examples of Large, Non-Convex, Unisolvent Sets***
Let $a \leq x_1 < \cdots < x_m \leq b$, $\sigma > 0$, and $k \in \mathbb{N}$ by given. If $m \geq 2k$, then the sets

- $\Big\{p(x) \ : \ \text{polynomial, at most } k/2 \text{ non-zero coefficients}\Big\}$,
- $\Big\{S(x) \coloneqq \sum_{j=1}^{k} c_j e^{\tau_j x} \ : \ c_j, \tau_j \in \mathbb{R}\, \forall j\Big\}$,
- $\Big\{S(x) \coloneqq \sum_{j=1}^{k} c_j e^{-(x-\tau_j)^2/2\sigma^2} \ : \ c_j, \tau_j \in \mathbb{R}\, \forall j\Big\}$,

are all unisolvent in $C([a,b])$ with respect to $x_1, \ldots, x_m$. If $a > 0$, then so is

- $\Big\{p(x) \ : \ \text{polynomial, at most } k \text{ non-zero coefficients}\Big\}$.

To keep things moving, we defer the proof of Proposition 3 to Appendix A. Our next result shows that the necessary conditions from Lemma 1 are essentially sufficient, too. Outside of trivial cases, if we define η as in Lemma 1, then (1) holds. We prove this under extra generality, for the sake of future work and consistency with the literature on optimal recovery. However, in this paper, we only consider the case in which $\|\cdot\|_{\mathbb{R}^m}$ is the ℓ^∞ norm, $F = C(\mathcal{X})$, and $\Lambda(f) = (f(x_1), \ldots, f(x_m))$, in which case $L = 1$.

Theorem 4 ***Main Theoretical Result:*** $\eta \leq M \leq \max\{\eta, 1 + L\eta\}$
Let F be any normed vector space, Σ any subset of F that has cardinality greater than 1, $\|\cdot\|_{\mathbb{R}^m}$ any norm on $\mathbb{R}^m$, and $\Lambda : F \to \mathbb{R}^m$ any bounded linear map. Call its operator norm $L \coloneqq \sup\{\|\Lambda f\|_{\mathbb{R}^m} / \|f\|_F : f \neq 0\}$, and also define opposing quantity

$$\eta \ \coloneqq \ \begin{cases} \sup\limits_{\substack{S,Z \in \Sigma \\ S \neq Z}} \dfrac{\|S - Z\|_F}{\|\Lambda(S - Z)\|_{\mathbb{R}^m}} & \text{if } \Sigma \text{ is unisolvent, i.e., } \ker(\Lambda) \cap (\Sigma - \Sigma) = \{0\}, \\ \infty & \text{if not.} \end{cases}$$

For all $f \in F$, $S \in \Sigma$, and $y \in \mathbb{R}^m$, we have

$$\underbrace{\|f - S\|_F}_{\substack{\text{generalization} \\ \text{error}}} \quad \leq \quad (1 + L\eta) \underbrace{\inf_{Z \in \Sigma} \|f - Z\|_F}_{\substack{\text{approximation} \\ \text{error}}} \ + \ \eta \underbrace{\|y - \Lambda S\|_{\mathbb{R}^m}}_{\text{training error}} \ + \ \eta \underbrace{\|y - \Lambda f\|_{\mathbb{R}^m}}_{\substack{\text{measurement} \\ \text{error}}}.$$

Conversely, if $\|f - S\|_F \leq M\big(\inf_{Z \in \Sigma} \|f - Z\|_F + \|y - \Lambda S\|_{\mathbb{R}^m} + \|y - \Lambda f\|_{\mathbb{R}^m}\big)$ for all $f \in F$, $S \in \Sigma$, and $y \in \Lambda(\Sigma)$, it must be the case that $\ker(\Lambda) \cap (\Sigma - \Sigma) = \{0\}$ and $M \geq \eta$.

Proof 4 Right away, let us assume that Σ is "unisolvent" in the generalized sense that $\ker(\Lambda) \cap (\Sigma - \Sigma) = \{0\}$. Otherwise, each conclusion is a trivial edge case.

If $M < \infty$, adapting the proof of Lemma 1 shows that $M \geq \eta$. Otherwise, $M \geq \eta$ regardless. The affirmative proof is short, too. Fix f, y, and S. Then, introduce $Z \in \Sigma$ over which we later take the infimum. Triangle inequality gives $\|f - S\|_F \leq \|S - Z\|_F + \|f - Z\|_F$. Next, using η and L to go back and forth between the norms of F and $\mathbb{R}^m$, we can bound

$$\begin{aligned} \|S - Z\|_F &\leq \eta \|\Lambda(S - Z)\|_{\mathbb{R}^m} \\ &\leq \eta \|y - \Lambda S\|_{\mathbb{R}^m} + \eta \|y - \Lambda f\|_{\mathbb{R}^m} + \eta \|\Lambda(f - Z)\|_{\mathbb{R}^m} \\ &\leq \eta \|y - \Lambda S\|_{\mathbb{R}^m} + \eta \|y - \Lambda f\|_{\mathbb{R}^m} + L\eta \|f - Z\|_F. \end{aligned}$$

Finally, since $Z \in \Sigma$ was arbitrary, infimizing yields the presented bound on $\|f - S\|_F$. ■

Researchers from optimal recovery may recognize that $(1 + \eta)\varepsilon$ is the intrinsic error of the model class $\{f \in F : \mathrm{dist}_F(f, \Sigma) \leq \varepsilon\}$ assuming that, say, Σ and $\Sigma - \Sigma$ are closed in F.

For the purposes of this paper, Theorem 4 says that M in (2) must satisfy $\eta \leq M \leq 1 + \eta$ or, roughly, $\eta \approx M$, where η is defined as in Lemma 1. To wrap up this section, we illustrate the gulf between being unisolvent and having a tolerable value of η. In fact, we observe that it is even possible to have $\eta = \infty$ when Σ is unisolvent.

Proposition 5 ***The Gap Between Unisolvence and a Usable Value of*** η
With an even number m of equally spaced points $a \leq 1 = x_1 < \cdots < x_m = 3 \leq b$,

- $\Big\{ p(x) \ : \ \text{polynomial, at most } m/2 \text{ non-zero coefficients} \Big\}$ *has* $\eta \geq 2^{m-3}/(m-1)^2$.
- $\Big\{ S(x) \coloneqq ce^{-(x-\tau)^2/2\sigma^2} \ : \ c, \tau \in \mathbb{R} \Big\}$ *has* $\eta = \infty$ *in* $C([a,b])$, *provided* $[a,b] \neq [1,3]$.

Proof 5 In Appendix A, Proposition 24 handles the case for which $\eta = \infty$ under extra generality. For now, let $\Sigma \subset C([a,b])$ denote the set of polynomials with at most $m/2$ non-zero coefficients. We check $\eta \geq 2^{m-3}/(m-1)^2$. For comparison, let $\mathcal{P}_m$ denote the set of all polynomials with degree at most $m-1$. Since $\Sigma - \Sigma \supset \mathcal{P}_m$, and $[a,b] \supset [1,3]$, we have

$$\eta \ = \ \sup\left\{ \frac{\|w\|_{C([a,b])}}{\max_i |w(x_i)|} \ : \ w \in (\Sigma - \Sigma) \setminus \{0\} \right\} \ \geq \ \sup\left\{ \frac{\|p\|_{C([1,3])}}{\max_i |p(x_i)|} \ : \ p \in \mathcal{P}_m \setminus \{0\} \right\}.$$

The latter is, exactly, the well studied "Lebesgue constant" for polynomial interpolation, which Theorem 2 in Trefethen and Weideman (1991) states is at least $2^{m-3}/(m-1)^2$. ■

4 Conventional ReLU Networks

We are now equipped to prove that (2) does *not* hold for any $M < \infty$ (equivalently, that $\eta = \infty$) when Σ is the set of functions expressible using a conventional, unrestricted feedforward ReLU architecture. According to the preceding analysis, it suffices to show that such a set is not unisolvent. The strongest such result has a short, self-contained proof under a convenient assumption on $\mathcal{X}$ that we later remove.

Theorem 6 ***Standard ReLU Networks are Not Unisolvent***
Assume that $\mathcal{X} \subset \mathbb{R}^d$ contains a point not in the closed convex hull of $\{x_1, \ldots, x_m\}$. If a subset Σ of $C(\mathcal{X})$ includes all width 1 shallow ReLU networks, then it is not unisolvent.

Proof 6 Let $x_o \in \mathcal{X} \setminus \mathcal{K}$, where $\mathcal{K} \coloneqq \mathrm{cl}(\mathrm{conv}(\{x_1, \ldots, x_m\}))$. By the separating hyperplane Theorem, there are $\theta \in \mathbb{R}^d$ and $c \in \mathbb{R}$ such that $c \geq \langle \theta, x \rangle$ for all $x \in \mathcal{K}$ while $\langle \theta, x_o \rangle > c$. Then, $S(x) \coloneqq \mathrm{ReLU}(\langle \theta, x \rangle - c)$ is zero on $\mathcal{K}$, but $S(x_o) \neq 0$. Letting Z denote the zero function, we have $S(x_i) = Z(x_i)$ for all $i = 1, \ldots, m$ but S and Z are not equal. Therefore, any set including both S and Z is not unisolvent with respect to $x_1, \ldots, x_m$. ■

Note that definitions of "shallow ReLU network" may differ on whether or not they include a bias term in the outer layer. From the proof of Theorem 6, we can see that the statement of the theorem remains true regardless of which convention is adopted. Also clear from the proof is that, by "shallow," we mean there is exactly one ReLU layer.

To be clear, since deeper and wider ReLU networks are able to replicate any width 1 shallow ReLU network by setting most coefficients to zero, Theorem 6 implies that the sets of functions expressed using such architectures are not unisolvent, either—assuming of course that the domain $\mathcal{X}$ is not a subset of the closed convex hull of $\{x_1, \ldots, x_m\}$. However, that assumption is merely a technical convenience, which we can readily remove.

Theorem 7 ***Similar Results Apply for More General $\mathcal{X}$, Leaky ReLU, etc.***
Let $\mathcal{X} \subset \mathbb{R}^d$ be any *strict superset of $\{x_1, \ldots, x_m\}$. If a subset Σ of $C(\mathcal{X})$ includes* either *every width 2 shallow ReLU network,* or *every width 3 shallow leaky ReLU network, then it is not unisolvent.*

We leave the proof of Theorem 7 to Appendix A. The lesson learned from Theorems 6 and 7 is that unisolvence is not a simple question of the number of model parameters.

However, unisolvence is not a simple question of "regularization," either. When we think of model regularization, a norm constraint on the parameters comes to mind. Let us point out that this does not achieve unisolvence.

Proposition 8 ***"Fair" Norm Constraints are Unlikely to Achieve Unisolvence***
For any $x_1, \ldots, x_m \in \mathbb{S}^{d-1}$, if a set $\Sigma \subset C(\mathbb{S}^{d-1})$ includes all shallow width 1 ReLU networks whose parameters' squared ℓ^2 norms sum to a total of 2 or less, then it is not unisolvent.

Proof 8 The counterexample from Theorem 6 has only three non-zero parameters. We simply quantify their magnitudes. Almost any point $x_o \in \mathbb{S}^{d-1}$ is not in the closed convex hull $\mathcal{K}$ of $\{x_1, \ldots, x_m\}$. Fix one. Then, $c \coloneqq \max\{\langle x_o, x \rangle_{\mathbb{R}^d} : x \in \mathcal{K}\}$ must be strictly less than 1. Hence, $S(x) \coloneqq \sqrt{1 - c^2}\,\mathrm{ReLU}(\langle x_o, x \rangle_{\mathbb{R}^d} - c)$ vanishes on $\mathcal{K}$, but not at x_o. At the same time, these parameters' squared ℓ^2 norms sum to $1 - c^2 + \|x_o\|^2_{\mathbb{R}^d} + c^2 = 2$. Consequently, a set Σ is not unisolvent if it includes both S and the zero function. ■

In Proposition 8, we say "fair" because the networks whose parameters' squared ℓ^2 norms sum to a total of γ or less may be excessively austere when $\gamma < 2$. That said, since Proposition 8 does not rule out literally every norm constraint, the possibility of an

extremely carefully calibrated one is left open. Additionally, by choosing the "small" domain $\mathcal{X} = \mathbb{S}^{d-1}$ we have tried to make it as easy as possible for ReLU networks to be unisolvent, and still they are not.

Finally, the work on mathematical super-resolution—as well as the classic counterexample of a "spike function" (which is indeed employed in the proof of Theorem 7)—both suggest that the problematic cases may be ruled out by prohibiting the breakpoints of a ReLU network from bunching together. However, that does not achieve unisolvence, either.

Proposition 9 ***Separation of Breakpoints does not Achieve Unisolvence*** *Assume $\mathcal{X} \subset \mathbb{R}^d$ contains a point not in the closed convex hull of $\{x_1, \ldots, x_m\}$. The set*

$$\left\{ S(x) \coloneqq \sum_{j=1}^{w} c_j \mathrm{ReLU}(\langle \alpha_j, x \rangle + \tau_j) \, : \, c_j \in \mathbb{R}, (\alpha_j, \tau_j) \in \mathbb{S}^d, \min_{j \neq \ell} \rho\big((\alpha_j, \tau_j), (\alpha_\ell, \tau_\ell)\big) \geq \delta \right\}$$

is not unisolvent for any $w, \delta > 0$ and function $\rho : \mathbb{S}^d \times \mathbb{S}^d \to \mathbb{R}$ (nor if $\mathbb{R}^{d+1}$ replaces $\mathbb{S}^d$).

Proof 9 Theorem 6 still applies. Indeed, even constraining $(\alpha_j, \tau_j) \in \mathbb{S}^d$, this set includes all width 1 shallow ReLU networks because, given $S(x) \coloneqq c\mathrm{ReLU}(\langle a, x \rangle + b)$, where $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$, we can replicate $S(x) = c_1 \mathrm{ReLU}(\langle \alpha_1, x \rangle + \tau_1)$ for all $x \in \mathbb{R}^d$, by choosing $(\alpha_1, \tau_1) = (a, b)/\sqrt{\|a\|^2 + b^2} \in \mathbb{S}^d$ and $c_1 \coloneqq c\sqrt{\|a\|^2 + b^2}$. Thus, the presented set includes S. To see that these are *all* shallow width 1 ReLU networks, recall from the above discussion that, for the purposes of Theorem 6, we need not consider a bias in the outer layer. ■

Based on the previous four results, we claim that no typical family of ReLU networks is unisolvent (outside of degenerate cases like $\mathcal{X} = \{x_1, \ldots, x_m\}$). To overcome this, we must find some family of ReLU networks that is unisolvent, even if non-obvious.

5 A Non-Standard Univariate ReLU Network

To gain traction, in this section and the next, we attend to the simplest case $\mathcal{X} = [a, b]$. Since ReLU networks are simply continuous piecewise linear functions, we study which sets of piecewise linear functions are unisolvent in $C([a, b])$. However, we must put restrictions on which ones we consider; otherwise, unisolvence does not even imply any limit on the number of breakpoints, as shown below.

Proposition 10 ***Strictly Speaking, Unisolvence does not Restrict Breakpoints*** *Given $a \leq x_1 < \cdots < x_m \leq b$ and any positive integer K, there exists a sub-space Σ of $C([a, b])$ with $\dim(\Sigma) = m$ consisting of piecewise linear functions, which is unisolvent and, favorably, even has $\eta \leq 2$, yet every non-zero $S \in \Sigma$ has more than K breakpoints.*

Proof 10 Take piecewise linear functions $\varphi_1, \ldots, \varphi_m \in C([a, b])$ that are very jagged versions of the typical "nodal" basis of hat functions. Specifically, we require (i) $\varphi_i(x_j) = \delta_{i,j}$, (ii) φ_i is zero on $[a, b] \setminus [x_{i-1}, x_{i+1}]$ (where $x_0 \coloneqq a$ and $x_{m+1} \coloneqq b$), (iii) each $\varphi_i(x)$ has supremum norm of 1 (so that $\eta \leq 2$), (iv) each φ_i has more than K breakpoints, and (v)

$\varphi_{i+1}(x)$ shares no breakpoints with $\varphi_i(x)$ besides x_i and x_{i+1} (so that we do not lose all breakpoints when taking linear combinations). These are easy to draw, or to construct probabilistically. Then, $\Sigma \coloneqq \text{Span}\{\varphi_1, \ldots, \varphi_m\}$ has the required properties. ■

The example from the proof of Proposition 10 may seem unfair. In approximation theory, a set $\Sigma \subset C([a,b])$ of piecewise linear functions is conventionally expected to have the property that, given any $S \in \Sigma$, all piecewise linear functions with the same breakpoints as $S(x)$ are included in Σ, too. In other words, with t being a discrete subset of $\mathbb{R}$, the convention is to focus on sets of piecewise linear functions of the form

$$\Sigma_t \coloneqq \{\text{all continuous, piecewise linear functions with breakpoints allowed in } t\}$$

or (for breakpoints of variable location), with $\mathcal{T}$ being a collection of discrete subsets of $\mathbb{R}$,

$$\Sigma_{\mathcal{T}} \coloneqq \bigcup_{t \in \mathcal{T}} \Sigma_t. \tag{3}$$

For sets $\Sigma_{\mathcal{T}}$ with this particular structure, we *can* characterize unisolvence, although the characterization is rather technical and difficult to conceptualize. For the reader's convenience, a direct and intuitive proof of the important Corollary 12 is provided in Appendix A.

Theorem 11 ***Classical Characterization of Unisolvence, via Knot Sequences*** *Let $a \leq x_1 < \cdots < x_m \leq b$. Assume that, given any $t, \tilde{t} \in \mathcal{T}$, there exist* disjoint *$u, \tilde{u} \in \mathcal{T}$ such that $u \cup \tilde{u}$ has cardinality $m-2$ and contains $t \cup \tilde{t}$. Then, the following are equivalent.*

- *$\Sigma_{\mathcal{T}}$ is unisolvent in $C([a,b])$ with resepct to $x_1, \ldots, x_m$.*
- *For any* disjoint *$t, \tilde{t} \in \mathcal{T}$, such that $t \cup \tilde{t}$ has cardinality $m-2$, the sorted values $s_1 < \cdots < s_{m-2}$ of $t \cup \tilde{t}$ satisfy $x_i < s_i < x_{i+2}$ for all $i = 1, \ldots, m-2$.*

Proof 11 Since $\Sigma_t - \Sigma_{\tilde{t}} = \Sigma_{t \cup \tilde{t}}$ for any $t, \tilde{t} \subset \mathbb{R}$, it follows that $\Sigma_{\mathcal{T}} - \Sigma_{\mathcal{T}} = \Sigma_{\mathcal{S}}$ where $\mathcal{S} \coloneqq \{t \cup \tilde{t} : t, \tilde{t} \in \mathcal{T}\}$. Instead of $\mathcal{S}$, though, in order to simplify the proof of Corollary 12, we will prefer to work with $\mathcal{R} \coloneqq \{t \cup \tilde{t} : t, \tilde{t} \in \mathcal{T}, t \cap \tilde{t} = \varnothing, \text{card}(t \cup \tilde{t}) = m-2\}$. On one hand, the fact that $\mathcal{S} \supset \mathcal{R}$ implies immediately that $\Sigma_{\mathcal{S}} \supset \Sigma_{\mathcal{R}}$. On the other hand, our assumption on $\mathcal{T}$ means that any $s \in \mathcal{S}$ is the *subset* of some $r \in \mathcal{R}$, from which it follows that $\Sigma_{\mathcal{S}} \subset \Sigma_{\mathcal{R}}$. In particular, $\Sigma_{\mathcal{T}} - \Sigma_{\mathcal{T}} = \Sigma_{\mathcal{R}}$ under our assumption on $\mathcal{T}$.

Unisolvence of $\Sigma_{\mathcal{T}}$ means that $w \in \Sigma_{\mathcal{T}} - \Sigma_{\mathcal{T}}$ is zero whenever $w(x_1) = \cdots = w(x_m)$. In other words, unisolvence of $\Sigma_{\mathcal{T}}$ means that, for every $s \in \mathcal{R}$, the implication

$$w(x_1) = \cdots = w(x_m) = 0 \quad \Longrightarrow \quad w = 0 \tag{4}$$

is valid for all $w \in \Sigma_s$. Fix $s \in \mathcal{R}$. Denote its sorted values as $s_1 < \cdots < s_{m-2}$. We prove that (4) holds for all $w \in \Sigma_s$ *if and only if* $x_i < s_i < x_{i+2}$ for all $i = 1, \ldots, m-2$.

The Schoenberg-Whitney theorem—which can be found in Chapter XIII, page 171 of de Boor (2001)—exhibits a basis $B_1, \ldots, B_m$ for Σ_s such that the matrix $A \in \mathbb{R}^{m \times m}$ defined $A_{i,j} \coloneqq B_j(x_i)$ is invertibile *if and only if* $x_i < s_i < x_{i+2}$ for all $i = 1, \ldots, m-2$. By the rank-nullity theorem, invertibility of this matrix is equivalent to injectivity. Finally, injectivity of A is in turn equivalent to the validity of the implication (4) for all $w \in \Sigma_s$ as—by definition of a basis—we can write $(w(x_1), \ldots, w(x_m)) = Ac$ where $w(x) = \sum_{j=1}^m c_j B_j(x)$. ■

Corollary 12 ***Breakpoints are Separated by 2 Data Sites*** $\implies$ ***Unisolvence***
Given $a \le x_1 < \cdots < x_m \le b$, call $k \coloneqq \lfloor m/2 \rfloor$. A set $\Sigma \subset C([a,b])$ of piecewise linear functions is unisolvent in $C([a,b])$ with respect to $x_1, \ldots, x_m$ if *every $S \in \Sigma$ has at most one breakpoint in each interval $[x_{2j}, x_{2j+1}]$ for $j < k$ and has no breakpoints elsewhere.*

Proof 12 (as a corollary to Theorem 11) Calling $\mathcal{T} \coloneqq \{\{t_1, \ldots, t_{k-1}\} : t_j \in [x_{2j}, x_{2j+1}]\}$, we have $\Sigma \subset \Sigma_{\mathcal{T}}$. Thus, it suffices to establish unisolvence of $\Sigma_{\mathcal{T}}$. We claim that $\mathcal{T}$ satisfies the assumptions of Theorem 11. Indeed, given $t, \tilde{t} \in \mathcal{T}$ with sorted values $t_1 < \cdots < t_{k-1}$ and $\tilde{t}_1 < \cdots < \tilde{t}_{k-1}$, respectively, we define $u_j \coloneqq t_j$ if $t_j \neq \tilde{t}_j$, otherwise we u_j to be a distinct element of $[x_{2j}, x_{2j+1}]$. Then, by construction, $\{u_1, \ldots, u_{k-1}\}, \{\tilde{t}_1, \ldots, \tilde{t}_{k-1}\} \in \mathcal{T}$ are disjoint and their union contains $t \cup \tilde{t}$. Since unisolvence with respect to $x_1, \ldots, x_{2k}$ implies unisolvence with respect to $x_1, \ldots, x_m$, we may assume without loss of generality that m is even, so that the union any disjoint elements of $\mathcal{T}$ has cardinality $m-2$.

Now, fix *disjoint* $t, \tilde{t} \in \mathcal{T}$. Let $s_1 < \cdots < s_{m-2}$ denote the sorted values of $t \cup \tilde{t}$. By Theorem 11, it will suffice to check that $x_i < s_i < x_{i+2}$ for all $i = 1, \ldots, m-2$. Fix $j < k$. We will check this for $i = 2j-1$ and $i = 2j$. Since $t_j \neq \tilde{t}_j$ by disjointness, we can be sure that $s_{2j-1} = \min\{t_j, \tilde{t}_j\}$ and $s_{2j} = \max\{t_j, \tilde{t}_j\}$ (this would fail if $t_j = \tilde{t}_j$). In particular, $s_{2j-1}, s_{2j} \in [x_{2j}, x_{2j+1}]$, so that $x_{2j-1} < x_{2j} \le s_{2j-1} < s_{2j} \le x_{2j+1}$ and $x_{2j} \le s_{2j-1} < s_{2j} \le x_{2j+1} < x_{2j+2}$. By Theorem 11, we conclude that $\Sigma_{\mathcal{T}}$ is unisolvent. ■

Let $\Sigma^{(k)}$ be the largest possible unisolvent set in the setting of Corollary 12. That is,

$$\Sigma^{(k)} \;\coloneqq\; \left\{ S(x) \coloneqq sx + d + \sum_{\ell=1}^{k-1} c_\ell \,\mathrm{ReLU}(x - \tau_\ell) \;:\; \tau_j \in [x_{2j}, x_{2j+1}] \,\forall j < k \right\}. \tag{5}$$

It is common to see $\mathrm{ReLU}(a_\ell x + b_\ell)$ instead of $\mathrm{ReLU}(x - \tau_\ell)$. However, this is just a different choice of parameterization, and produces the same set of functions. Using the latter convention, $-b_\ell/a_\ell$ are the breakpoints, and $\Sigma^{(k)}$ can be identically expressed as

$$\left\{ S(x) \coloneqq a_0 x + b_0 + \sum_{\ell=1}^{k-1} c_\ell \mathrm{ReLU}(a_\ell x + b_\ell) \;:\; -b_j \in \begin{cases} [a_j x_{2j}, a_j x_{2j+1}] & \text{if } \; a_j > 0 \\ [a_j x_{2j+1}, a_j x_{2j}] & \text{if } \; a_j < 0 \\ \mathbb{R} & \text{if } \; a_j = 0 \end{cases} \forall j > 0 \right\}.$$

Although the set is identical in both cases, (5) will be more convenient for optimization because, in that parameterization, the constraints are convex (in fact, linear).

Since we loosened the equivalent condition of Theorem 11 in order to derive Corollary 12, one should wonder how much slack we have sacrificed in the process. Our next result says that we have essentially only conceded a mild topological assumption on $\mathcal{T}$. We defer its somewhat lengthy proof to Appendix A.

Theorem 13 ***Concerning the Sharpness of*** **Corollary 12**
Let $a \le x_1 < \cdots < x_m \le b$. Let $T_1, \ldots, T_n \subset [a,b]$ be intervals with $\inf(T_1) \le \cdots \le \inf(T_n)$. Call $k \coloneqq \lfloor m/2 \rfloor$. If $k \ge n$, then the set of piecewise linear functions

$$\Sigma(T_1, \ldots, T_n) \;\coloneqq\; \left\{ S(x) \coloneqq sx + d + \sum_{\ell=1}^{n} c_\ell \mathrm{ReLU}(x - \tau_\ell) \;:\; \tau_j \in T_j \,\forall j \right\}$$

cannot be unisolvent. If $n = k-1$ and m even, $\Sigma(T_1, \ldots, T_n)$ is unisolvent if and only if $T_j \subset [x_{2j}, x_{2j+1}]$ $\forall j$. *If $n = k-1$ and m odd, it is unisolvent* if and only if, *for some ℓ, $T_\ell \subset [x_{2\ell}, x_{2\ell+2}]$, while $T_j \subset [x_{2j+1}, x_{2j+2}]$ $\forall j > \ell$ and $T_j \subset [x_{2j}, x_{2j+1}]$ $\forall j < \ell$.*

If the connectedness assumption is removed from Theorem 13, then larger unisolvent sets of a similar form can be found. For instance, when $m = 4$, we have $\Sigma^{(2)} = \Sigma([x_2, x_3])$. However, if $s \in (x_1, x_2)$ and $t \in (x_3, x_4)$, then the larger set $\Sigma([x_2, x_3] \cup \{s, t\})$ is also unisolvent. Admittedly, the latter can be *meaningfully* larger than $\Sigma^{(2)}$, since it includes $\Sigma^{(2)}$ as a subset,, while also allowing breakpoints at substantially different locations. However, the constraint $\tau \in [x_2, x_3] \cup \{s, t\}$ is non-convex. With the excuse of numerical tractability, we stick with as many (and as large) as possible *connected* constraint sets $T_j = [x_{2j}, x_{2j+1}]$.

Now that we have an interesting unisolvent set, it is essential to check the value of η. Possibly discarding one data point, we do so for the relatively simple case when m is even.

Proposition 14 ***Computing η for $\Sigma^{(k)}$***
Given $a \le x_1 < \cdots < x_m \le b$, label $x_0 \coloneqq a$ and $x_{m+1} \coloneqq b$. If $m = 2k$, then $\Sigma^{(k)}$ satisfies

$$\begin{aligned} \eta &= 1 + 2 \max_{j=1}^{k} \frac{\Delta_j}{\delta_j} && \text{where} \quad \begin{cases} \Delta_j \coloneqq \max\{x_{2j+1} - x_{2j},\, x_{2j-1} - x_{2j-2}\}, \\ \delta_j \coloneqq x_{2j} - x_{2j-1}, \end{cases} \\ &\le 1 + 2\frac{h_{\max}}{h_{\min}} && \text{where} \quad \begin{cases} h_{\max} \coloneqq \max\{x_{i+1} - x_i : 0 \le i \le m\}, \\ h_{\min} \coloneqq \min\{x_{i+1} - x_i : 0 < i < m\}. \end{cases} \end{aligned}$$

Proof 14 Since $\Delta_j/\delta_j \le h_{\max}/h_{\min}$ for all j, the upper bound involving $h_{\max}/h_{\min}$ is an easy consequence. We now focus on verifying $\eta = 1 + 2\max_j(\Delta_j/\delta_j)$. Since η is defined via a supremum, it is natural to take a two-staged approach in which we separately prove both $\eta \le 1 + 2\max_j(\Delta_j/\delta_j)$ and $\eta \ge 1 + 2\max_j(\Delta_j/\delta_j)$, beginning with the former.

Let $\tau_1 \le \cdots \le \tau_p$ denote the breakpoints of some $w \in \Sigma^{(k)} - \Sigma^{(k)}$. Call $\tau_0 \coloneqq a$ and $\tau_{p+1} \coloneqq b$. For each $\ell = 0, \ldots, p+1$, there is $j = 1, \ldots, k$ such that τ_ℓ is an endpoint of an interval containing x_{2j-1} and x_{2j} on which $w(x)$ is linear. On that interval,

$$w(x) = w(x_{2j-1})\frac{x_{2j} - x}{x_{2j} - x_{2j-1}} + w(x_{2j})\frac{x - x_{2j-1}}{x_{2j} - x_{2j-1}} \tag{6}$$

(indeed, this formula for $w(x)$ is readily verified at both $x = x_{2j-1}$ and $x = x_{2j}$). Hence,

$$\begin{aligned} |w(\tau_\ell)| &\le |w(x_{2j-1})|\frac{|x_{2j} - \tau_\ell|}{x_{2j} - x_{2j-1}} + |w(x_{2j})|\frac{|\tau_\ell - x_{2j-1}|}{x_{2j} - x_{2j-1}} \\ &\le \frac{|x_{2j} - \tau_\ell| + |\tau_\ell - x_{2j-1}|}{\delta_j} \max_{i=1}^{m} |w(x_i)|. \end{aligned}$$

Moreover, the interval on which $w(x)$ is linear can be taken to be a subset of $[x_{2j-2}, x_{2j+1}]$. If $\tau_j \in [x_{2j}, x_{2j+1}]$, then $|x_{2j} - \tau_\ell| \le \Delta_j$ and $|\tau_\ell - x_{2j-1}| \le |x_{2j} - x_{2j-1}| + |x_{2j} - \tau_\ell| \le \delta_j + \Delta_j$. If, instead, $\tau_\ell \in [x_{2j-2}, x_{2j-1}]$ then, similarly, $|x_{2j-1} - \tau_\ell| \le \Delta_j$ and $|\tau_\ell - x_{2j}| \le \delta_j + \Delta_j$. So $|x_{2j} - \tau_\ell| + |\tau_\ell - x_{2j-1}| \le \delta_j + 2\Delta_j$ in either case, resulting in

$$|w(\tau_\ell)| \le \left(1 + 2\frac{\Delta_j}{\delta_j}\right) \max_{i=1}^{m} |w(x_i)|.$$

Using the property of continuous piecewise linear functions that $\|w\|_\infty = \max_\ell |w(\tau_\ell)|$,

$$\|w\|_\infty \;=\; \max_{\ell=0}^{p+1} |w(\tau_\ell)| \;\le\; \max_{j=1}^{k} \left(1 + 2\frac{\Delta_j}{\delta_j}\right) \max_{i=1}^{m} |w(x_i)| \;=\; \left(1 + 2\max_{j=1}^{k} \frac{\Delta_j}{\delta_j}\right) \max_{i=1}^{m} |w(x_i)|.$$

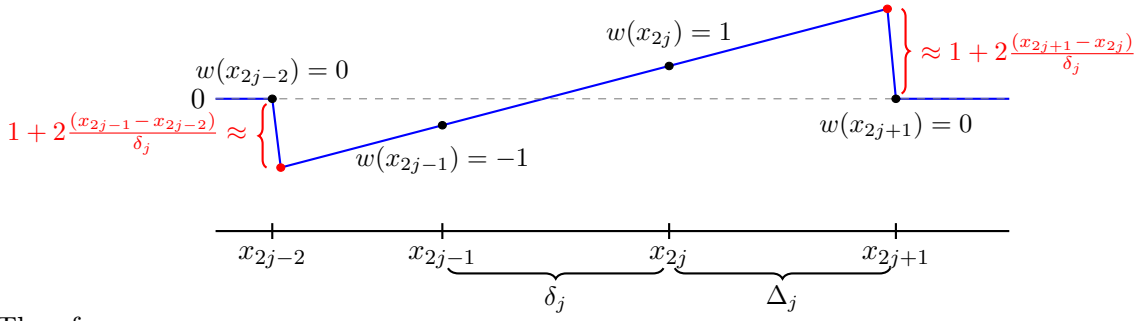
Since w was arbitrary, this means that $\eta \le 1 + 2\max_j(\Delta_j/\delta_j)$. Next, we address the reverse.

Fix $j = 1, \dots, k$. We begin to build w by connecting the dots with prescribed values $w(x_{2j-1}) = -1$ and $w(x_{2j}) = 1$. Next, we introduce two or four breakpoints to ensure that $w(x_i) = 0$ for $i \neq 2j, 2j-1$. Take a small $\varepsilon > 0$. If $j \ge 2$, then let us introduce breakpoints at x_{2j-2} and $x_{2j-2} + \varepsilon\delta_j$ and set $w(x) = 0$ for all $x \le x_{2j-2}$. Similarly, if $j \le k-1$, then we introduce breakpoints at $x_{2j+1} - \varepsilon\delta_j$ and x_{2j+1} and set $w(x) = 0$ for all $x \le x_{2j+1}$. By construction, $\max_i |w(x_i)| = 1$, so that $\eta \ge \|w\|_\infty$. It remains to lower bound $\|w\|_\infty$.

Whether or not a breakpoint is located at $x_{2j+1} - \varepsilon\delta_j$, we can see from (6) that

$$\begin{aligned} w(x_{2j+1} - \varepsilon\delta_j) \;&=\; \frac{x_{2j+1} - \varepsilon\delta_j - x_{2j}}{\delta_j} \;+\; \frac{x_{2j+1} - \varepsilon\delta_j - x_{2j-1}}{\delta_j} \\ &=\; \frac{x_{2j+1} - x_{2j}}{\delta_j} \;-\; \varepsilon \;+\; \frac{x_{2j+1} - x_{2j}}{\delta_j} \;+\; 1 \;-\; \varepsilon \;=\; 1 \;+\; 2\frac{x_{2j+1} - x_{2j}}{\delta_j} \;-\; 2\varepsilon \end{aligned}$$

and, similarly, $w(x_{2j-2} + \varepsilon\delta_j) = -\big(1 \,+\, 2(x_{2j-1} - x_{2j-2})/\delta_j \,-\, 2\varepsilon\big)$, as depicted below for the typical case $1 < j < k$.



Therefore,

$$\eta \;\ge\; \|w\|_\infty \;\ge\; \max\big\{w(x_{2j+1} - \varepsilon\delta_j),\, -w(x_{2j-2} + \varepsilon\delta_j)\big\} \;=\; 1 + 2\frac{\Delta_j}{\delta_j} - 2\varepsilon.$$

Since we can construct such a w for any $j = 1, \dots, k$ and any sufficiently small $\varepsilon > 0$, this means that $\eta \ge 1 + 2\max_j(\Delta_j/\delta_j)$. ∎

Evidently, when using $\Sigma^{(k)}$, the value of η depends on the arrangement of the data. Notably, if $x_1, \dots, x_m$ form an evenly spaced grid including the endpoints $x_1 = a$ and $x_m = b$, then $\eta = 3$. However, if some x_i and x_{i+1} are extremely close to one another, it can lead to a large value of η with this set $\Sigma^{(k)}$. For example, if $[a, b] = [0, 1]$ and $x_1, \dots, x_m$ are the sorted values of a uniform sample from $[0, 1]$, then it is probable that $\eta \gtrsim m$.

In real-world scenarios, if two data sites happen to be very close together, it may be necessary to opt for a unisolvent set *smaller* than $\Sigma^{(k)}$, trading approximation power for stability by reducing the value of η. If two data sites happen to be nearly right on top of each other, for example, this can be accomplished by simply dropping one of them from the data set. We do not explore this further, considering the limited impact of univariate regression. Rather, we regard $\Sigma^{(k)}$ in (5) as more of a proof of concept.

6 Designing a Model with Small Training Error

In this section, we assume that an even number of points $a \le x_1 < x_2 < \cdots < x_{2k} \le b$ have been given and we let $\Sigma^{(k)} \subset C([a,b])$ denote the largest unisolvent set in the setting of Corollary 12—see (5). We now consider how to approximately solve the training problem

$$\min_{S \in \Sigma^{(k)}} \max_{i=1}^{2k} |S(x_i) - y_i|. \tag{7}$$

This is a non-convex, constrained optimization program. To make it tractable, we have two main options: either to minimize over the *parameters*, such as

$$\min_{\substack{s,\, d,\, c_1,\ldots,c_{k-1} \in \mathbb{R} \\ \tau_j \in [x_{2j}, x_{2j+1}]\ \forall j}} \max_{i=1}^{2k} \left| s x_i + d + \sum_{\ell=1}^{k-1} c_\ell \,\mathrm{ReLU}(x_i - \tau_\ell) \;-\; y_i \right| \tag{8}$$

(recall, the constraints on τ_j make $\Sigma^{(k)}$ unisolvent), or to minimize over the *predictions*:

$$\min_{z \in \Lambda(\Sigma^{(k)})} \|z - y\|_{\ell^\infty(\mathbb{R}^{2k})} \qquad \text{where} \qquad \Lambda(f) \;:=\; (f(x_1), \ldots, f(x_m)). \tag{9}$$

The latter is equivalent thanks to unisolvence and is arguably more canonical because it does not depend on any particular choice of parameterization.

The loss landscape of (7), (8), and (9) is much less treacherous than what we are accustomed to in machine learning. Normally, the models with smallest training error must be avoided, since it is not assumed that $y_i = f(x_i)$. In this case, however, (1) says that the training error remains proportional to the error at unseen data. Therefore, we counter-intuitively have no use for validation data during training. As a data-free alternative, we develop a mathematical test, which serves a similar purpose (see Proposition 20). If excess data is available, then we may either reserve it for validating assumptions that have nothing to do with training—discussed in the following section—or simply include it as further training data. The former may reduce uncertainty stemming from modeling assumptions, while the latter may result in a more powerful fit. In particular, the former use of non-training data is still warranted in risk averse settings.

Let us suggest a couple of options for approximately solving (8) and (9). We discourage the popular techniques of implementing either an inexact projection [3] onto $\prod_{j<k}[x_{2j-1}, x_{2j}]$ or penalty functions, due to the availability of these more canonical and reliable methods:

1. Since the constraints are (8) is convex, we can adapt any gradient-based method (say, ADAM) by applying the orthogonal projection onto the constraint set after each step, by simply clamping each parameter τ_j to the range between x_{2j} and x_{2j+1}.

2. The constraints in (9) will turn out to be $\Lambda(\Sigma^{(k)}) = \bigcap_{\ell<k}\{z : q_\ell(z) \le 0\}$ where $q_1, \ldots, q_{k-1} : \mathbb{R}^m \to \mathbb{R}$ are non-convex quadratic functions (see Proposition 15). Therefore, (9) can be cast as a quadratic program (QP). We know how to solve the

3. Recall that the orthogonal projection of a point x_o onto a convex subset Q of a Hilbert space is defined to the unique element of Q closest to x_o in Hilbert norm. Any other point may be called an "inexact projection" of x_o onto Q. For example, if $f : \mathbb{R} \to \mathbb{R}$ is a sigmoid function rescaled to satisfy $f_j(t) \searrow x_{2j}$ as $t \searrow -\infty$ and $f_j(t) \nearrow x_{2j+1}$ as $t \nearrow \infty$, then $f_j(t)$ is an inexact projection of t onto $[x_{2j}, x_{2j+1}]$.

dual of *any* QP (see Lemma 18) and, in this case, how to convert the dual solution into an approximate minimizer of (7) (see Lemma 19 and the following discussion). However, this approach will be more effective when used to minimize $\frac{1}{m}\|z-y\|_{\ell^2}^2$ instead of $\|z-y\|_{\ell^\infty}$ (see Proposition 17 and the following discussion).

In the remainder of this paper, we focus on the second approach—in part because it requires more explanation—but moreover because the machinery of dual programming provides lower bounds on (7), which serve to test near-optimality (see Proposition 20 and the following discussion). We feel this is necessary for a theoretical analysis, as both algorithms are otherwise not rigorously justified. We wish to focus on theory instead of simulations, but we also make our code available at `https://github.com/ThomasLastName/near-optimal`.

First of all, let us understand the constraint set in (9). To avoid distraction, we simply state the conclusion and leave the derivation to Appendix B.

Proposition 15 ***Explicit Description of*** $\Lambda\big(\Sigma^{(k)}\big)$
There are $\alpha^{(1)},\dots,\alpha^{(k-1)},\beta^{(1)},\dots,\beta^{(k-1)}\in\mathbb{R}^{2k}$ *depending only on* $x_1,\dots,x_{2k}$ *such that*

$$\Lambda(\Sigma^{(k)}) \;=\; \bigcap_{\ell=1}^{k-1}\{z : q_\ell(z)\le 0\} \qquad \text{where} \qquad q_\ell(z) \;\coloneqq\; z^\intercal\big(\alpha^{(\ell)}\alpha^{(\ell)\intercal}-\beta^{(\ell)}\beta^{(\ell)\intercal}\big)z.$$

Specifically, for $\ell=1,\dots,k-1$, *both* $\alpha^{(\ell)}$ *and* $\beta^{(\ell)}$ *have four non-zero entries given by*

$$\alpha^{(\ell)}_{2\ell-1} = -\frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell-1}}, \quad \alpha^{(\ell)}_{2\ell} = \frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell-1}}+2, \quad \alpha^{(\ell)}_{2\ell+1} = -\frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell+1}}-2, \quad \alpha^{(\ell)}_{2\ell+2} = \frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell+1}},$$
$$\beta^{(\ell)}_{2\ell-1} = \frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell-1}}, \quad \beta^{(\ell)}_{2\ell} = -\frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell-1}}, \quad \beta^{(\ell)}_{2\ell+1} = -\frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell+1}}, \quad \beta^{(\ell)}_{2\ell+2} = \frac{\widehat{x}_{2\ell}}{\widehat{x}_{2\ell+1}},$$

where we define the first order differences $\widehat{x}_i\coloneqq x_{i+1}-x_i$ *for all* $i=1,\dots,2k-1$.

Now that we know the constraints in (9) are a system of quadratic inequality constraints, we can then naturally express the *epigraph form* of (9) as a quadratic program (QP):

$$\inf_{z\in\Lambda(\Sigma^{(k)})}\|y-z\|_{\ell^\infty} \;=\; \inf\big\{\, t \;:\; -t\le y_i-z_i\le t\,\forall i,\;\; q_\ell(z)\le 0\,\forall\ell\,\big\}. \tag{10}$$

Let us immediately point out the main difficulty here.

Proposition 16 ***For every*** $\ell=1,\dots,k-1$***, the Function*** q_ℓ ***is Non-Convex***

Proof 16 For purely quadratic functions, convexity is equivalent to non-negativity. Suppose $q_\ell(z)\ge 0$ for all $z\in\mathbb{R}^{2k}$. Then, whenever $z\perp\alpha^{(\ell)}$, we get $0\le q_\ell(z)=-\big|\big\langle z,\beta^{(\ell)}\big\rangle\big|^2$ and so $z\perp\beta^{(\ell)}$. This means $\mathrm{Span}\{\alpha^{(\ell)}\}^\perp\subset\mathrm{Span}\{\beta^{(\ell)}\}^\perp$ and so $\mathrm{Span}\{\beta^{(\ell)}\}\subset\mathrm{Span}\{\alpha^{(\ell)}\}$. That is impossible, since $\beta^{(\ell)}$ is not a multiple of $\alpha^{(\ell)}$. ■

For non-convex constraints, the first thing to try should usually be the Lagrangian dual program, [4] but the following exercise shows that (10) does not take well to this technique.

4. An alternative approach—specific to QPs—is to form a semidefinite relaxation. Our implementation of this is, also, included in the corresponding code `https://github.com/ThomasLastName/near-optimal`,

Proposition 17 ***The Dual Maximum of*** (10) ***is Zero***

Proof 17 First, we check that, for any $\lambda \geq 0$, we have

$$\inf_{z,t}\left(t \,+\, \sum_{\ell=1}^{k-1} \lambda_\ell q_\ell(z) \,+\, \sum_{i=1}^{m} \lambda_{i+k-1}(y_i - z_i - t) \,+\, \sum_{i=1}^{m} \lambda_{i+m+k-1}(z_i - y_i - t) \right) \;\leq\; 0. \quad (11)$$

If $\lambda_\ell > 0$ for some $\ell = 1, \ldots, k-1$, then the infimum is unbounded below, due to Proposition 16. However, if $\lambda_1 = \cdots = \lambda_{k-1} = 0$, then (11) again holds since a value of zero is obtained when $z_i = y_i$ and $t = 0$. To complete the proof, note that (11) holds with equality when $\lambda_{i+k-1} = \lambda_{i+m+k-1}$ for all $i = 1, \ldots, m$, $\sum_{i \geq k} \lambda_i = 1$, and $\lambda_1 = \cdots = \lambda_{k-1} = 0$. ■

There several ways to rewrite (10) as an equivalent QP, which all exhibit similar behavior. For instance, if we replace the constraint $-t \leq y_j - z_j \leq t$ with $(z_j - y_j)^2 \leq t^2$, then minimizing t is unbounded below. Adding the constraing $t \geq 0$, we empirically found a dual max of zero. Minimizing t^2 instead of t, we found a non-zero but quite small dual max.

Something else we can try—which is intuitive, form a machine learning perspective—is to replace the ℓ^∞ norm with the much more conventional mean-squared error, resulting in

$$\inf_{z \in \Lambda(\Sigma^{(k)})} \frac{1}{m}\|y - z\|_{\ell^2}^2 \;\;=\;\; \inf_{\substack{z \in \mathbb{R}^m \\ q_\ell(z) \leq 0 \,\forall \ell}} \frac{1}{m} z^\intercal I z \,+\, 2z^\intercal(-y) \,+\, \|y\|_{\ell^2}^2. \quad (12)$$

This is a relaxation of (the square of) (10). More generally, various relaxations follow from

$$\|y - z\|_{\ell^\infty}^2 \;\;\geq\;\; \sum_{i=1}^{m} w_i (y_i - z_i)^2, \quad (13)$$

so long as $w \geq 0$ and $\sum_i w_i = 1$. In (12), we just happen to choose $w_i = 1/m$. We experimented with the choice of weights, but did not find much room for improvement.

Unlike (10), the dual problem of (12) is non-trivial. Here is how we solve it.

Lemma 18 ***The Lagrangian Dual Program of any QP***
The dual max d^* of $p_* := \min\{z^\intercal Q z + 2z^\intercal r + s \,:\, z^\intercal A^{(\ell)} z + 2z^\intercal b^{(\ell)} + c_\ell \leq 0 \;\forall \ell = 1, \ldots, L\}$ and optimal penalty parameters $\lambda_1, \ldots, \lambda_L$ can be computed by the semidefinite program

$$d^* \;\;=\;\; \sup\left\{ \, t \,:\, \begin{pmatrix} Q + \sum_\ell \lambda_\ell A^{(\ell)} & r + \sum_\ell \lambda_\ell b_\ell \\ \left(r + \sum_\ell \lambda_\ell b_\ell\right)^\intercal & s + \sum_\ell \lambda_\ell c_\ell - t \end{pmatrix} \succeq 0 \;,\; \lambda_1, \ldots, \lambda_L \geq 0 \, \right\}.$$

If $t^*, \lambda_1^*, \ldots, \lambda_L^*$ solve the latter, then we also have

$$d^* \;\;=\;\; \inf_{z \in \mathbb{R}^m} \left(z^\intercal Q z \,+\, 2z^\intercal r \,+\, s \,+\, \sum_{\ell=1}^{L} \lambda_\ell^* \big(z^\intercal A^{(\ell)} z + 2z^\intercal b^{(\ell)} + c_\ell \big) \right) \;\leq\; p_*. \quad (14)$$

Proof 18 The semidefinite constraint is, in fact, the epigraph constraint

$$t \;\leq\; \inf_z \left(z^\intercal Q z + 2z^\intercal r + s \,+\, \sum_{\ell=1}^{L} \lambda_\ell \big(z^\intercal A^{(\ell)} z + 2z^\intercal b^{(\ell)} + c_\ell \big) \right),$$

so that $d^* = \sup\{t : t \leq \inf_z(\cdots),\, \lambda_1, \ldots, \lambda_L \geq 0\}$ can be written in the announced manner. Then, (14) consists of simply recalling the definition of the dual program. ■

The unconstrained infimum in z—appearing in (14)—is the sharpest relaxation that can be obtained using "penalty parameters" for the constraints. Its first order condition on z is

$$\left(Q + \sum_{\ell=1}^{L} \lambda_\ell^* A^{(\ell)}\right) z \;=\; -\left(r + \sum_{\ell=1}^{L} \lambda_\ell^* b_\ell\right). \tag{15}$$

An ideal outcome is if a solution z_* of (15) solves the primal QP (the one with infimum p_*). Otherwise, we rely on tricks to "adjust" z_* with the hopes of justifying them a posteriori (in our case, by Proposition 20). So far, these are general ideas applicable to *any* QP.

In our case, recall that z_* is supposed to represent $(S(x_1), \dots, S(x_{2k}))$. Even in the best case scenario, there still remains the task of converting $z_* \in \Lambda\big(\Sigma^{(k)}\big)$ into a neural network $S \in \Sigma^{(k)}$ that can be used for inference. Before discussing any "tricks," let us point out how to perform this final step under ideal circumstances.

Lemma 19 ***How to Reverse Engineer*** $S \in \Sigma^{(k)}$ ***from*** $z_* \in \Lambda(\Sigma^{(k)})$
If $z_* \in \Lambda(\Sigma^{(k)})$, then the unique $S \in \Sigma^{(k)}$ satisfying $z_* = \Lambda(S)$ is given by

$$S(x) \;\coloneqq\; s_1 x + a_1 - s_1 m_1 \;+ \sum_{\substack{j=1 \\ s_{j+1} \neq s_j}}^{k-1} (s_{j+1} - s_j)\mathrm{ReLU}(x - \tau_j) \tag{16}$$

where we define the midpoint $m_j \coloneqq (x_{2j-1} + x_{2j})/2$, the average $a_j \coloneqq (z^*_{2j-1} + z^*_{2j})/2$, and the slope $s_j \coloneqq \frac{z^*_{2j} - z^*_{2j-1}}{x_{2j} - x_{2j-1}}$ for $j = 1, \dots, k$, meanwhile, for $j < k$ with $s_{j+1} \neq s_j$, we define

$$\tau_j \;\coloneqq\; \frac{(s_{j+1} m_{j+1} - s_j m_j) - (a_{j+1} - a_j)}{s_{j+1} - s_j}. \tag{17}$$

To not derail the presentation, the proof of Lemma 19 is left to Appendix B. Next, we turn to the reality that a solution z_* of (15) does not exactly solve the intended QP.

Our simple heuristic for what to do in this scenario is to apply Lemma 19, regardless! Generally, solutions z_* of (15) need not be feasible, meaning the assumption $z_* \in \Lambda(\Sigma^{(k)})$ from Lemma 19 may be violated. In that case, we can still define S as in Lemma 19, but we end up with $S \notin \Sigma^{(k)}$. For the presented theory, it is non-negotiable that the constraints must be exactly satisfied. So, we still need to fix this.

Exploiting the fact that the constraints in (8) are convex, [5] *our extremely ad hoc remedy is to clamp each value of τ_j to the range between x_{2j} and x_{2j+1}.* Let $\overline{S}$ denote the function resulting from this recipe (to distinguish it from the function S in Lemma 19). That is,

$$\overline{S}(x) \;\coloneqq\; s_1 x + a_1 - s_1 m_1 \;+ \sum_{\substack{j=1 \\ s_{j+1} \neq s_j}}^{k-1} (s_{j+1} - s_j)\mathrm{ReLU}(x - \overline{\tau}_j)$$

where, for indices $j = 1, \dots, k-1$ with $s_{j+1} \neq s_j$, we define τ_j as in Lemma 19 but then define $\overline{\tau}_j$ to be the orthogonal projection of τ_j onto $[x_{2j}, x_{2j+1}]$.

In summary, as an alternative to ADAM with a projection step, we also propose the following recipe, which we feel is a relatively natural approach to (9).

5. If we could project z_* onto $\Lambda\big(\Sigma^{(k)}\big)$, that would be another option. However, this is generally infeasible.

```
Def fit( x_1<···<x_{2k}, y_1,...,y_{2k} ):

    # Solve the dual of (12) and convert to z_*
    Define α^(1),...,α^(k-1),β^(1),...,β^(k-1) as in Proposition 15
    Define A^(ℓ) ≔ α^(ℓ)α^(ℓ)ᵀ − β^(ℓ)β^(ℓ)ᵀ for all ℓ = 1,...,k−1
    Define Q ≔ I, r ≔ (−y_1,...,−y_{2k}), and s ≔ ‖r‖²_{ℓ²}
    Maximize for t*, λ*_1,...,λ*_L in Lemma 18 # use any semidefinite solver
    Solve (15) for z_*

    # The rest is basically just adjusting z_* to satisfy the constraints
    s, d, c_1,...,c_{k-1}, τ_1,...,τ_{k-1} = connect_the_dots(z_*)  # defined below
    Modify τ̄_j = clamp( τ_j lower=x_{2j}, upper=x_{2j+1} ) for j = 1,...,k−1
    Return the function S̄(x) ≔ sx + d + Σ_{j=1}^{k-1} c_j ReLU(x − τ̄_j)

Def connect_the_dots(z_1,...,z_{2k}): # given z_* define S as in Lemma 19
    For j = 1,...,k:
        Define m_j ≔ (x_{2j−1} + x_{2j})/2
        Define a_j ≔ (z_{2j−1} + z_{2j})/2
        Define s_j ≔ (z_{2j} − z_{2j−1})/(x_{2j} − x_{2j−1})
        If j > 1 and s_j ≠ s_{j−1}:   # can't define τ_j (no 'j+1' terms yet)
            Define τ_{j−1} via (17) # τ_{j−1} = ((s_j m_j − s_{j−1} m_{j−1}) − (a_j − a_{j−1}))/(s_j − s_{j−1})
        Else:
            Define τ_{j−1} ≔ x_{2(j−1)} # dummy value satisfying the constraints
    Return the list [ s_1, a_1 − s_1 m_1, s_2 − s_1, ..., s_k − s_{k−1}, τ_1, ..., τ_{k−1} ]
```

It remains to justify this recipe. Of course, if we find any $S \in \Sigma^{(k)}$ with a small training error, then (1) already justifies it. However, (1) does not tell us whether S is *optimal*—is there an $\tilde{S} \in \Sigma^{(k)}$ with meaningfully smaller training error that we just have not found yet? We could always try a handful of options $S_1, \ldots, S_N \in \Sigma^{(k)}$ and take best among them, as in hyperparameter tuning. However, we would prefer a more absolute criterion, in order to avoid unnecessary trial and error. One of the core ideas from dual programming is that we may run the following test, say, taking D to be the dual max of any relaxation of (10).

Proposition 20 ***A Posteriori Test of Near-Optimality***
Let $0 < D \le \inf_{Z \in \Sigma^{(k)}} \max_i |y_i - Z(x_i)|$. *For* any $\overline{S} \in \Sigma^{(k)}$, *no matter how it is obtained,*

$$\max_{i=1}^{2k} \left|y_i - \overline{S}(x_i)\right| \;\le\; \widehat{C} \inf_{Z \in \Sigma^{(k)}} \max_{i=1}^{2k} |y_i - Z(x_i)| \quad \text{where} \quad \widehat{C} \;\coloneqq\; \frac{1}{D} \max_{i=1}^{2k} \left|y_i - \overline{S}(x_i)\right|.$$

Further, for any $f \in C([a,b])$, *the generalization error at arbitrary inputs* $x \in [a,b]$ *obeys*

$$\left\|f - \overline{S}\right\|_\infty \;\le\; \Big(1 + \eta + \widehat{C}\eta\Big) \inf_{Z \in \Sigma^{(k)}} \|f - Z\|_\infty \;+\; \Big(\eta + \widehat{C}\eta\Big) \max_{i=1}^{2k} |y_i - f(x_i)|.$$

Proof 20 The first bound is automatic from the fact that $D \le \inf_Z \max_i |y_i - Z(x_i)|$. Then, we plug the first into (1) and use $\max_i |y_i - Z(x_i)| \le \max_i |y_i - f(x_i)| + \|f - Z\|_\infty$. ■

Notice that $\widehat{C}$ can never be smaller than 1, although smaller is better. Any value larger than 10 is dubious, whereas $\widehat{C} < 2$ is excellent. Proposition 20 proves that, if $\widehat{C} < 2$, then $\overline{S}$ approximately solves (7) with a relative error of less than 1, accrediting that $\overline{S}$ fits the data practically just as well as any solution of (7). Absent a fully rigorous algorithm, Proposition 20 may be used to justify—in principle—any "applied trick," so long as the end result satisfies the constraints *exactly* (not "almost") and the value of $\widehat{C}$ is convincing.

The catch is that the value of $\widehat{C}$ cannot be anticipated in advance. If our proposed model $\overline{S}$ leads to a poor value of $\widehat{C}$, then we must explore more options (our code provides several). The good news is that we are free to experiment as greedily as desired until finding some $S \in \Sigma^{(k)}$ with a good value of $\widehat{C}$. Moreover, in this experimentation, it is not necessary to compare competing models—*any* $S \in \Sigma^{(k)}$ with $\widehat{C} < 2$ is well optimized. We simply cannot guarantee, in a worst case, how much experimentation might be needed to find one.

We also reiterate the assumptions $S \in \Sigma^{(k)}$ and $D \le \inf_{Z \in \Sigma^{(k)}} \max_i |y_i - Z(x_i)|$. Whereas the former is relatively clear, the latter generally cannot be verified empirically. So, the hard part is usually proposing a value of D. Thankfully, one of the main features of dual programming is that it provides one for us: because (12) is less than $(\inf_{Z \in \Sigma^{(k)}} \|f - Z\|_\infty)^2$, we may take D to be the square root of the dual max of (12). Whereas we recommend (12) for obtaining $\overline{S}$, in order to obtain D we instead recommend optimizing the weights in (13), as follows (see Appendix B for the derivation).

Proposition 21 ***A Sharpened Lower Bound on*** (7)
Let $x_1 < \cdots < x_{2k}$ and $y_1, \ldots, y_{2k} \in \mathbb{R}$. With $\alpha^{(\ell)}, \beta^{(\ell)} \in \mathbb{R}^{2k}$ as in Proposition 15*, define $A^{(\ell)} := \alpha^{(\ell)}\alpha^{(\ell)\intercal} - \beta^{(\ell)}\beta^{(\ell)\intercal}$ for each ℓ. Letting $\odot$ denote entry-wise multiplication of vectors,*

$$b^* := \sup\left\{ t \,:\, \begin{pmatrix} \mathrm{diag}(w) + \sum_{j<k} \lambda_j A^{(j)} & -w \odot y \\ -(w \odot y)^\intercal & \sum_j w_j y_j^2 - t \end{pmatrix} \succeq 0 \,,\ \lambda_j \ge 0,\, w \ge 0,\, \sum_j w_j = 1 \right\}$$

satisfies $\sqrt{d^*} \le \sqrt{b^*} \le \inf_{Z \in \Sigma^{(k)}} \max_i |y_i - Z(x_i)|$ where d^* is the dual max of (12)*.*

In our code, we test other lower bounds on (7), as well, but comparing them here is not pertinent, as one good one is enough. Pseudocode for computing $\widehat{C}$ using $D = \sqrt{b^*}$ follows.

```
Def near_optimality_ratio( x_1 < ··· < x_2k, y_1,...,y_2k, S̄, b*=optional ):
    Confirm that S̄ ∈ Σ^(k)          # otherwise, Proposition 20 is invalidated
    Define error = max( |S̄(z_i) − y_i| for i = 1,...,2k )
    If error==0: Return 1           # in this edge case, D > 0 is impossible
    Compute b* if not provided      # see Proposition 21; use any solver
    If b*==0: stop('D = 0! Try and see if Proposition 19 yields S ∈ Σ^(k)')
    else: Return error/sqrt(b*)  # always ≥ 1, but the smaller the better
```

This is a powerful test that can be used to justify the optimization error of *any* algorithm that produces an element of $\Sigma^{(k)}$. For instance, we can also recommend ADAM with a projection step. It is ad hoc, too, but since it enforces the constraints exactly, we can still turn to Proposition 20 in hopes of justifying it—say, using $\widehat{C} < 2$ as a stopping condition.

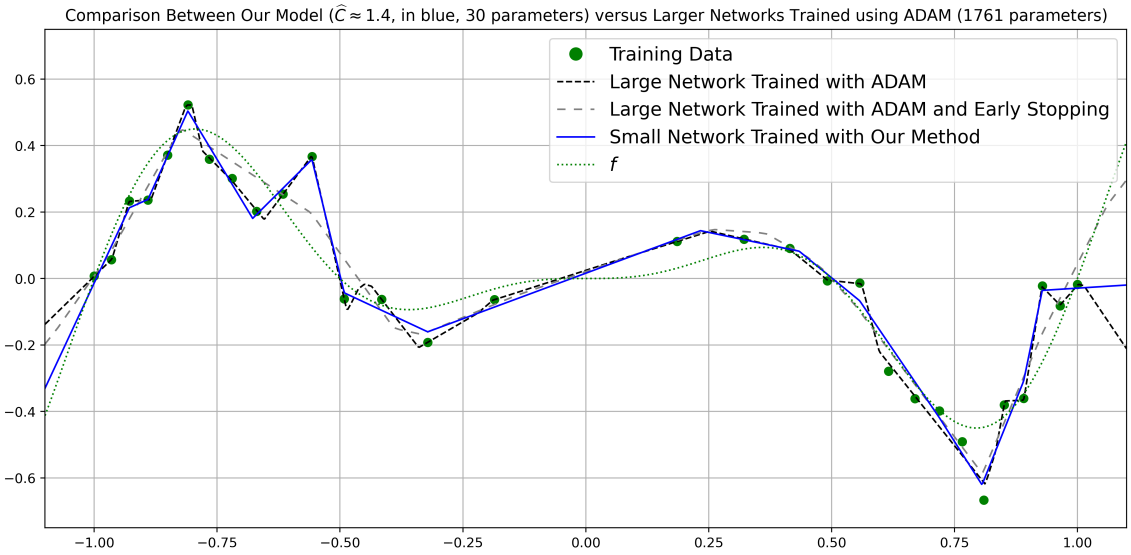


Figure 1: Image returned by `quadratic_univar.py` from the codebase for this paper, which can be found at `https://github.com/ThomasLastName/near-optimal`.

In cursory tests, our proposed algorithm (pseudocode above) returned a value of $\widehat{C} < 2$ and a qualitatively satisfactory fit, as depicted in Figure 1.

Observe that we cannot expect the generalization error (how far the solid curve deviates from f) to be any better than the measurement error (how far the scatterplot deviates from f), as is evident in (1). Also, notice that a contemporarily popular method marginally outperforms ours in a few narrow (unpredictable) regions. We reiterate that we do *not* aim to produce cutting edge *empirical* results. Rather, we aspire to develop theoretically rigorous methods that perform empirically even half as well as our contemporaries.

There are many nuances we have not addressed—most notably the instability of some semidefinite solvers (one can always fall back to ADAM with a projection step) and the time and memory complexity of these algorithms (clearly, ADAM with a projection step has the same asymptotic complexity as without). A refined implementation is not the intent of this theoretical paper, nor do we expect univariate regression to be impactful enough to warrant one. The important principles to be gleamed from this case study are the following.

1. We must consider how to numerically enforce any constraints achieving unisolvence.

2. By leveraging the structure of unisolvence, it may be possible to design highly precise "boutique" training methods. In the case studied, we have presented a rigorous certificate of optimality for an algorithm which, favorably, does not have hyperparameters.

3. As discussed above, in this framework there are no unwanted minimizers of (7). This alleviates the need for model comparison, as it suffices to find even a single model with small training error. Strictly speaking, test and validation data are not necessary for training, though they may be used to validate other modeling assumptions (see below).

7 Approximation and Measurement Errors

Consider the difference between $\widehat{C}$ and the more familiar concept of validation error. Whereas the latter is used to gauge a model's ability to generalize to unseen data, we use $\widehat{C}$ only to assess the quality of an optimization algorithm. Questions of generalization are instead handled by the rest of the theory that we have presented. In particular, even in the ideal case $\widehat{C} = 1$, that by itself does not imply a model generalizes well. Proposition 20 exposes all the factors that go into generalization: η, $\widehat{C}$, the *approximation error* $\inf_{Z \in \Sigma^{(k)}} \|f - Z\|_\infty$, and the *measurement error* $\max_i |y_i - f(x_i)|$ must all be adequate in order to control the generalization error $\|f - \overline{S}\|_\infty$ of a proposed model $\overline{S} \in \Sigma^{(k)}$. We have yet to address the latter two of these four factors. In fact, we will *assume* that the approximation and measurement errors are small. Let us discuss why.

In our view, a data-driven approach is doomed if the data is wrong, and so we regard the assumption that $\max_i |y_i - f(x_i)|$ is small as being *necessary*. The nice thing is that our theory does not suddenly fail catastrophically if the measurement error exceeds some threshold. However accurate the data is or is not, we do as well as we can—in the sense that the upper bound in (1) adapts to the severity of the measurement error $\max_i |y_i - f(x_i)|$.

There is a mathematical tradition of studying approximation error via the smoothness of the function f to be approximated. In that vein, one can estimate $\inf_{Z \in \Sigma^{(k)}} \|f - Z\|_\infty$ in terms of the spacing of $x_1, \ldots, x_m$ and the modulus of continuity of f. However, we believe that such techniques are unlikely to be fruitful in the multivariate setting. For instance, with $\mathcal{X} = [0,1]^d$ it is known that, for *any* $\Sigma \subset C(\mathcal{X})$, there is a 1-Lipschitz continuous function f satisfying $\inf_{S \in \Sigma} \|f - S\|_\infty \gtrsim p^{-1/d} \approx 1$, where $p = p(\Sigma)$ can be thought of as the number of parameters needed to describe Σ—for a more precise statement see, e.g., Proposition 25.4 in Foucart (2022). In theory, it helps if f has unit Sobolev norm with high smoothness, but the assumption of unit norm is often considered to be increasingly unrealistic as d increases. So, while (1) itself may be true for all $f \in C(\mathcal{X})$, we emphatically *cannot* assume that $\inf_{Z \in \Sigma} \|f - Z\|_\infty \le \varepsilon$ for all f.

Yet, as noted in the introduction, $\inf_{Z \in \Sigma} \|f - Z\|_\infty \le \varepsilon$ is *necessary* in order to obtain a generalization error smaller than ε. In data-driven settings, we do not know anything about f, and so likely we have no means of proving that f happens to be one of the lucky ones for which $\inf_{Z \in \Sigma} \|f - Z\|_\infty$ is small. We see no option other than to include this as an assumption. Many accepted methods do so, as well. For instance, one of the assumptions of ordinary least squares (sometimes called the "correct specification" assumption) is that the data-generating process is linear. That is, precisely, the assumption $\inf_{Z \in \Sigma} \|f - Z\|_\infty = 0$, where Σ is the set of affine functions on $\mathcal{X}$.

In the multivariate setting, we expect the assumption $\inf_{Z \in \Sigma} \|f - Z\|_\infty \le \varepsilon$ to be unrealistic whenever Σ is a vector space with dimension at most m. However, we are not necessarily satisfied otherwise. If we allow $\tau_1, \ldots, \tau_k$ to vary, then the set

$$\left\{ S(x) \coloneqq \sum_{j=1}^{k} c_j e^{-10000\|x - \tau_j\|^2} \ : \ c_1, \ldots, c_k \in \mathbb{R},\, \tau_1, \ldots, \tau_k \in \mathbb{R}^d \right\}$$

is not a finite dimensional vector space, but it still has unsatisfactory approximation power because it only consists of functions with k steep peaks, meaning that no element of this set can approximate a "regular" function well for bounded k (say, $k < m$).

In summary, our axioms are that the approximation and measurement errors are small. When η and $\widehat{C}$ are favorable, Proposition 20 shows that these two assumptions are *sufficient* to guarantee a small generalization error. By the above discussion, we also perceive them as *necessary* in a data-driven framework, where special information about f is not available.

If anything goes wrong *while η and $\widehat{C}$ are favorable*, we can be certain of the cause. By Proposition 20, the only possibility is that, either, the measurement or approximation error must not have been small. An advantage of this theory is that we can pinpoint the possible reasons for failure and not have to fear that any quirk of neural networks might betray us.

Ultimately, any regression method lives or dies based on its empirical performance. The remaining (important) role of empiricism in this work is to corroborate our assumptions with a reality check. In synthetic experiments where f is known, the effect of the approximation error can be isolated. In practical applications, a fitted model's error on a holdout data set would help to gauge the plausibility of our two axiomatic assumptions. Specifically, the root mean squared error is a lower bound on $\|f - \overline{S}\|_\infty$. So, when η and $\widehat{C}$ are under control, a large holdout error would constitute proof that one of our two assumptions are violated.

8 Future Work

The burning question is how to extend these techniques to the multivariate setting. The theory is certainly valid in complete generality. However, at the time of writing, we lack an example of, say, $\mathcal{X} \subset \mathbb{R}^2$, $x_1, \ldots, x_m \in \mathcal{X}$, and $\Sigma \subset C(\mathcal{X})$ meeting the following 5 criteria:

1. Σ is unisolvent with a moderate value of η (ideally, $\eta < 10$),
2. Σ is large enough to justify the assumption of small approximation error (see above),
3. Given $y \in \mathbb{R}^m$, there is a tractable algorithm for producing $S \in \Sigma$ with $y_i \approx S(x_i)$,
4. $\mathcal{X}$ is not unreasonably small ($\mathcal{X} = [0,1] \times [0,1]$ would be good), and
5. The locations of $x_1, \ldots, x_m$ are not too restricted (ideally, not required to be a grid).

Let us note several considerations when looking for such an example.

First, we dispel any doubt that unisolvence might be a purely univariate phenomenon. There certainly exist unisolvent sets of multivariate functions, such as the earlier example

$$\Sigma_{\text{kern}} \;\coloneqq\; \text{Span}\big\{\text{kern}(\,\cdot\,, x_1), \ldots, \text{kern}(\,\cdot\,, x_m)\big\}$$

from kernel-based methods, which is unisolvent whenever the kernel matrix is invertible. Furthermore, a moderate value of η is possible. For instance, given $x_1, \ldots, x_m \in \mathbb{R}^d$, one may construct $\varphi_1, \ldots, \varphi_m \in C(\mathbb{R}^d)$ satisfying $\varphi_j(x_i) = \delta_{i,j}$ (called a "nodal basis"). Then,

$$\Sigma_{\text{nodal}} \;\coloneqq\; \text{Span}\{\varphi_1, \ldots, \varphi_m\}$$

is unisolvent, and can be made to have a moderate value of η by prescribing further properties to $\varphi_1, \ldots, \varphi_m$, as in the proof of Proposition 10. However, these examples are both vector spaces, which we strongly discourage for fear that their approximation power is not scalable to the multivariate setting.

For more specific guidance on future work, we pose the following question, which asks whether or not Corollary 12 is true in two variables.

Question: *Let $x_1, \dots, x_{3k} \in \mathbb{R}^2$. Suppose that $\{x_1, \dots, x_{3k}\} = T_1 \cup \cdots \cup T_k$ where each set T_j contains 3 points forming a non-degenerate triangle (i.e., the points are not co-linear) and $\mathrm{cl}(\mathrm{conv}(T_i)) \cap \mathrm{cl}(\mathrm{conv}(T_j)) = \varnothing$ for all $i \neq j$.*

Let $\Sigma_2^{(k)} \subset C(\mathbb{R}^2)$ be a set of piecewise linear functions such that each $S \in \Sigma_2^{(k)}$ has the following property: any maximal region on which S is linear contains one of the triangles $\mathrm{cl}(\mathrm{conv}(T_j))$. Is $\Sigma_2^{(k)}$ unisolvent in $C(\mathrm{conv}(\{x_1, \dots, x_m\}))$?

To be clear, in this question, the triangles are *not* a triangulation of $x_1, \dots, x_m$. Rather, they play the role of the intervals $[x_1, x_2], \dots, [x_{2k-1}, x_{2k}]$ from the univariate case. To clarify further, observe that, for $k = 2$ and any admissible clusters T_1 and T_2, we have

$$\Sigma_2^{(2)} \subset \left\{ S(x,y) \coloneqq ax + by + c + \gamma \mathrm{ReLU}(\alpha x + \beta y + \tau) \ : \ \begin{smallmatrix} T_1 \text{ and } T_2 \text{ are separated by the} \\ \text{line } \{(x,y) : \alpha x + \beta y + \tau = 0\} \end{smallmatrix} \right\}$$

and an affirmative answer.

Proposition 22 ***$\Sigma_2^{(2)}$ is Unisolvent in $C(\mathbb{R}^2)$ with Respect to $x_1, x_2, x_3, x_4, x_5, x_6$***

Proof 22 Take $S, Z \in \Sigma_2^{(2)}$. Then, $S - Z$ is continuous and piecewise linear with at most 4 regions of linearity, forming 4 convex cones originating from the same point. Of those regions, 2 non-adjacent ones each contain 3 points at which $S - Z$ is zero. Hence, $S - Z$ is zero throughout those regions, by linearity. Next, $S - Z$ must be zero on the boundary of the remaining 2, and so is zero throughout those by linearity, as well. ■

For dimensions $d > 2$, the corresponding set $\Sigma_d^{(k)}$ would require $m = (d+1)k$ points clustered into sets $T_1, \dots, T_k$ of $d+1$ points each, whose closed convex hulls form disjoint, non-degenerate simplexes. Again, notice that this is consistent with Corollary 12. Also, notice that the described set $\Sigma_d^{(k)}$ is automatically unisolvent in $C\big(\bigcup_j \mathrm{conv}(T_j)\big)$. If the answer to the above question is affirmative, then some next steps are to consider how to enforce such a constraint practically, and to estimate the value of η analytically. If the answer is negative, then the next steps would depend on why, exactly, unisolvence fails.

9 Conclusion

We have presented a framework for how to mathematically guarantee favorable generalization in a regression setting. One of the key concepts is a certain a "complexity statistic" η. See Theorem 4 for the general definition, Lemma 1 for a simplified formula when Σ is unisolvent, and (1) for a simplified result. When analyzed carefully—most notably, checking that η is not too large—our theory delivers an exceedingly strong, deterministic assurance of a model's ability to generalize to out-of-sample and even adversarially selected inputs.

We thus have demonstrated that it may be possible to derive clear constraints which mathematically prevent overfitting. In exchange for the difficulty of deriving and enforcing them, we must reiterate the reward. Since theory-building is a one-time investment, it can be more efficient in the long run compared to the recurring expense of empirically validating each new instance of a ReLU network. In addition to explainability and robustness, we envision faster, cheaper, and less laborious model development by leveraging mathematical

assurances to reduce the need for hyperparameter tuning. However, this vision is contingent upon the design of multivariate models satisfying the aforementioned requirements. Proposing compliant multivariate models remains a compelling open challenge. Its resolution would offer a foundational alternative to empirical techniques in regression tasks. This paper provides an example and building blocks for such an approach.

Appendix A. Supplementary Proofs on Unisolvent Sets

The following lemma seems to have novelty value, although we doubt it is original.

Lemma 23 ***Non-Linear Fundamental Theorems of Algebra***
Consider $f(x) \coloneqq c_1 x^{\theta_1} + \cdots + c_n x^{\theta_n}$. If the domain is $(0,\infty)$, then we allow $\theta_j \in \mathbb{R}$, and f is identically zero once it vanishes at n distinct points. If the domain is $\mathbb{R}$, then we instead allow only $\theta_j \in \mathbb{N}$, and f is identically zero once it vanishes at $2n$ distinct points. In the latter case, it is indeed generally necessary to check $2n$ points.

Proof 23 Factor $f(x) = x^{\theta_n}(c_1 x^{\tau_1} + \cdots + c_{n-1}x^{\tau_{n-1}} + c_n) =: x^{\theta_n} g(x)$ where $\tau_j \coloneqq \theta_{n-1} - \theta_n$. Say, f vanishes at N distinct *non-zero* points, although we have yet to quantify how many that is. Since $x \neq 0 \implies x^{\theta_n} \neq 0$ and the points are non-zero, g vanishes at them, too. Between any two of them, $g'(x) = c_1\tau_1 x^{\tau_1 - 1} + \cdots + c_{n-1}\tau_{n-1}x^{\tau_{n-1}-1}$ must have a zero, by Rolle's theorem. So, g' vanishes at $N-1$ distinct points, at least. Assuming $\theta_1 > \cdots > \theta_n$ without loss of generality, then g' is the same type of function as f, only with one fewer term (in particular, $\tau_1 - 1, \ldots, \tau_{n-1} - 1 \in \mathbb{N}$ if $\theta_1, \ldots, \theta_n \in \mathbb{N}$). If the domain is $(0,\infty)$, then the $N-1$ points where g' vanishes are all *non-zero*. Otherwise, at least $N-2$ of them are.

With $N = n$ and $N = 2n-1$, we get induction loops for the two cases. (in the latter case, f vanishing at $2n$ points assures it vanishes at $2n-1$ *non-zero* points). Once down to only a single term cx^θ, it must be identically zero since it vanishes at a non-zero point.

Finally, let us confirm that it can be necessary to check $2n$ points by constructing a non-zero polynomial with n terms and $2n-1$ real roots. Take any $\xi_{n-1} > \cdots > \xi_1 > 0$. Then, $p(x) \coloneqq (x-\xi_1)\cdots(x-\xi_{n-1})$ has at most n terms and is non-zero at some $x_o > 0$. The same is true of $q(x) \coloneqq xp(x^2)$, which vanishes at $-\sqrt{\xi_{n-1}}, \ldots, -\sqrt{\xi_1}, 0, \sqrt{\xi_1}, \ldots, \sqrt{\xi_{n-1}}$. ■

Proof 3 We verify these functions are zero for all $x \in [a,b]$ if they vanish at $x_1, \ldots, x_m$:

1. $\sum_{j=1}^{2k} c_j e^{\tau_j x}$ with $\tau_j \in \mathbb{R}$,
2. $\sum_{j=1}^{2k} c_j e^{-(x-\tau_j)^2/2\sigma^2}$ with $\tau_j \in \mathbb{R}$, assuming $\sigma > 0$,
3. $\sum_{j=1}^{k} c_j x^{n_j}$ with $n_j \in \mathbb{N}$,
4. $\sum_{j=1}^{2k} c_j x^{n_j}$ with $n_j \in \mathbb{N}$, assuming $a > 0$.

For the first item, $p(y) = \sum_{j=1}^{2k} c_j y^{\tau_j}$ vanishes at $e^{x_1}, \ldots, e^{x_m}$, implying $c_1 = \cdots = c_{2k} = 0$ by Lemma 23. For the second item, $\sum_{j=1}^{2k} c_j e^{-(x-\tau_j)^2/2\sigma^2}$ has the same zeros as the function

$$e^{x^2/2\sigma^2} \sum_{j=1}^{2k} c_j e^{-(x-\tau_j)^2/2\sigma^2} \;=\; \sum_{j=1}^{2k} c_j e^{-\tau_j^2/2\sigma^2} e^{x\tau_j/\sigma^2}.$$

So, $c_1 e^{-\tau_1^2/2\sigma^2} = \cdots = c_m e^{-\tau_m^2/2\sigma^2} = 0$ by the preceding item and, thus, $c_1 = \cdots = c_m = 0$. The third and fourth items are even more direct consequences of Lemma 23. ■

Proof 7 We show that, if an activation function σ can be used to construct a non-zero, compactly supported *univariate* function, then unisolvence fails in the multivariate case.

Let $x_o \in \mathcal{X} \setminus \{x_1, \ldots, x_m\}$. For each $i = 1, \ldots, m$, because $x_i - x_o \neq 0$, the set H_i of vectors orthogonal to $x_i - x_o$ is a $(d-1)$-dimensional sub-space of $\mathbb{R}^d$, hence has zero Lebesgue measure. Thus, based on measure, $\mathbb{R}^d \setminus (H_1 \cup \cdots \cup H_m)$ must be non-empty. So, we can find a vector $\theta \in \mathbb{R}^d$ with $(x_i - x_o)^\intercal \theta \neq 0$ for all $i = 1, \ldots, m$. Call $\delta \coloneqq \min_i |(x_i - x_o)^\intercal \theta|$.

Say, a function $\varphi : \mathbb{R} \to \mathbb{R}$ is non-zero, compactly supported, and given by the formula

$$\varphi(t) \;=\; \sum_{j=1}^{w} c_j \sigma(a_j t + b_j). \tag{18}$$

Replacing $\varphi(t)$ by $\varphi((t - t_o)/K)$ where t_o is the center of φ's support, we may assume without loss of generality that $\varphi(0) \neq 0$ while $\varphi(t) = 0$ whenever $|t| \geq \delta$. Then, $S(x) \coloneqq \varphi((x - x_o)^\intercal \theta)$ is a width w shallow network with activation function σ satisfying $S(x_i) = 0$ for all i and, yet, $S(x_o) = \varphi(0) \neq 0$. Splitting the summation and negating half the coefficients, we see that S is the difference of two neural networks of width at most $\lceil w/2 \rceil$, whose values differ at x_o and match on $x_1, \ldots, x_m$. In particular, any set including all width $\lceil w/2 \rceil$ shallow networks with activation σ is not unisolvent in $C(\mathcal{X})$ with respect to $x_1, \ldots, x_m$.

As for constructing such a function φ, recall that the standard "hat function" φ can be written as $\varphi(t) = \mathrm{ReLU}(t+1) - 2\mathrm{ReLU}(t) + \mathrm{ReLU}(t-1)$ and $\varphi(t) = (|t+1| + |t-1|)/2 - |t|$. If σ is the leaky ReLU function, it suffices to note that $|t| = c \cdot (\sigma(t) + \sigma(-t))$ with $c \approx 1$. ■

Proof 12 (direct) By assumption, any $w \in \Sigma - \Sigma$ has at most *two* breakpoints in each interval $[x_2, x_3]$, $[x_4, x_5]$, ... and no breakpoints elsewhere. Suppose $w(x_i) = 0$ for all i. Immediately, w is zero anywhere *outside of* $\bigcup_{j<k} (x_{2j}, x_{2j+1}) = (x_2, x_3) \cup (x_4, x_5) \cup \cdots$ since, on each excluded region, w is linear while vanishing at two points.

Fix $j < k$. We will check that w is zero on $[x_{2j}, x_{2j+1}]$. At worst, w has two breakpoints τ_1 and τ_2 in this interval. Say, $x_{2j} \leq \tau_1 < \tau_2 \leq x_{2j+1}$. Since $w(x)$ is linear on $[x_{2j-1}, \tau_1]$ and zero on $[x_{2j-1}, x_{2j}]$, it must in fact be zero on $[x_{2j-1}, \tau_1]$. Similarly, w must be zero on $[\tau_2, x_{2j+2}]$. Hence, $w(\tau_1) = 0 = w(\tau_2)$ so that w is finally zero on $[\tau_1, \tau_2]$, as well. ■

Proof 13 Let us gradually discover conditions on $T_1, \ldots, T_n$ necessary for unisolvence.

First, we rule out the possibility that any T_j intersects $[a, x_1)$ or $(x_m, b]$. If any breakpoint is allowed outside of $[x_1, x_m]$, then we can easily construct $w \in \Sigma - \Sigma$ violating unisolvence, e.g., $w(x) = \mathrm{ReLU}(x - \tau)$ if $\tau > x_m$. So, $T_j \subset [x_1, x_m]$ is necessary.

Second, we rule out the possibility that any T_j intersects $[x_1, x_2)$ or $(x_{m-1}, x_m]$. Because T_j is an interval and (WLOG) $T_j \cap [a, x_1) = \varnothing = T_j \cap (x_m, b]$, there would be *two* points at which T_j intersects either (x_1, x_2) or (x_{m-1}, x_m). Say, $x_1 < q < r < x_2$ where $q, r \in T_j$. Then, $w(x)$ with two total breakpoints at q and r, defined by connecting the dots $w(x_1) = 0$, $w(q) = -1$, $w(r) = 0$, and $w(x_2) = 0$, is in $\Sigma - \Sigma$, violating unisolvence (draw it!). The case $T_j \cap (x_{m-1}, x_m) \neq \varnothing$ is similar. So, $T_j \subset [x_2, x_{m-1}]$ is necessary for unisolvence.

Third, we rule out the possibility that any two different sets T_j and T_ℓ respectively intersect (x_i, x_{i+1}) and $[x_i, x_{i+1}]$ for some $i = 2, \ldots, m-2$. Since T_j and T_ℓ are intervals, there would be two points where T_j intersects (x_i, x_{i+1}) and a third point (apart from those two) where T_ℓ intersects $[x_i, x_{i+1}]$. These three breakpoints allows us to construct a "hat" function $w \in \Sigma - \Sigma$ with narrow support that hides in between x_i and x_{i+1}, violating unisolvence. Therefore, for every $i = 2, \ldots, m-2$, it is necessary for unisolence that, if $T_j \cap (x_i, x_{i+1}) \neq \varnothing$, then we have $T_\ell \cap [x_i, x_{i+1}] = \varnothing$ for all $\ell \neq j$.

Fourth, we rule out the possibility that adjacent open intervals (x_i, x_{i+1}) and (x_{i+1}, x_{i+2}) $(i = 2, \ldots, m-3)$ are respectively intersected by distinct T_j and T_ℓ. Suppose $x_i < q < r < x_{i+1} < s < t < x_{i+2}$ where $q, r \in T_j$ and $s, t \in T_\ell$. Then, w with four total breakpoints, defined by connecting the dots $w(x_i) = 0$, $w(q) = 0$, $w(r) = -1$, $w(s) = (s - x_{i+1})/(x_{i+1} - r)$ (which simply is chosen so that $w(x_{i+1}) = 0$), $w(t) = 0$, and $w(x_2) = 0$, is in $\Sigma - \Sigma$, violating unisolvence (draw it!). Therefore, it is necessary for unisolvence that $T_1, \ldots, T_n$ never intersect neighboring open intervals (x_i, x_{i+1}) and (x_{i+1}, x_{i+2}).

All of this is to say, the intervals $T_1, \ldots, T_n$ must be somewhat separated, although they are not allowed to leave $[x_2, x_{m-1}]$. The conclusions of this theorem thus follow by simple counting arguments. For the sake of exposition, let us verify an unnecessarily high base case of $m = 6$ (so $k = 3$). Let us try to pack $T_1 = [a_1, b_1]$ and $T_2 = [a_2, b_2]$ with $a_1 \leq a_2$ into $[x_2, x_3]$, $[x_3, x_4]$, and $[x_4, x_5]$. If $b_1 > x_3$, it means $T_1 \cap (x_3, x_5) \neq \varnothing$, in which case our packing constraints are sure to be violated. Alternatively, $b_1 \leq x_3 \iff T_1 \subset [x_2, x_3]$, and in that case $T_2 \subset [x_4, x_5]$ is necessary. Clearly, for $n \geq k = 3$, there is no viable packing.

Roughly speaking, for $m = 2k$, assume that we cannot pack $T_1, \ldots, T_k$ or more intervals into $[x_2, x_3]$, $\ldots$, $[x_{m-2}, x_{m-1}]$ subject to our separation constraints on $T_1, \ldots, T_k$, and that we can pack $T_1, \ldots, T_{k-1}$ *only* if $T_j \subset [x_{2j}, x_{2j+1}]$ $\forall j < k$. For $m = 2(k+1)$, we want to show that $T_1, \ldots, T_{k+1}$ or more cannot be packed into two more intervals than we had before, and that $T_1, \ldots, T_k$ can be packed only if $T_j \subset [x_{2j}, x_{2j+1}]$ $\forall j < k+1$. Indeed, subject to our packing constraints, $[x_{m-3}, x_{m-1}]$ has room for one of the sets T_ℓ, and by the induction hypothesis the rest cannot fit into the remaining slots. Similarly, if we only need to pack $T_1, \ldots, T_k$, then the induction hypothesis says that only $T_j \subset [x_{2j}, x_{2j+1}]$ for all $j < k$ is acceptable, leaving room for T_k, but only if $T_k \subset [x_{2k}, x_{2k+1}]$.

Finally, let us only briefly sketch the situation in which m is odd. A base case for induction on the odd integers could be $m = 7$. Imagine trying to pack $T_1 = [a_1, b_1]$ and $T_2 = [a_2, b_2]$ with $a_1 \leq a_2$ into $[x_2, x_3]$, $[x_3, x_4]$, $[x_4, x_5]$, and $[x_5, x_6]$. There are several possible packings, all of a similar type. Unisolvence holds so long as, *either*, $T_1 \subset [x_2, x_4]$ and $T_2 \subset [x_5, x_6]$, *or* $T_1 \subset [x_2, x_3]$ and $T_2 \subset [x_4, x_6]$. In general, there are $k-1$ cases, as only one of the sets $T_1, \ldots, T_{k-1}$ may use this extra slack, as stated in this theorem. ■

Proposition 24 ***At best, Unconstrained RBF Networks have*** $\eta = \infty$
Assume $\mathcal{X} \subset \mathbb{R}^d$ contains a point not in the closed convex hull of $\{x_1, \ldots, x_m\}$ and $\sigma > 0$. If $\{e^{-\|x-\tau\|^2/2\sigma^2} : \tau \in \mathbb{R}^d\}$ is unisolvent in $C(\mathcal{X})$ with respect to $x_1, \ldots, x_m$, then $\eta = \infty$.

Proof 24 Suppose there was $\eta < \infty$ such that, for all $\tau, \tilde{\tau} \in \mathbb{R}^d$ and $x \in \mathcal{X}$, we have

$$\left| e^{-\|x-\tilde{\tau}\|^2/2\sigma^2} - e^{-\|x-\tau\|^2/2\sigma^2} \right| \leq \eta \max_{i=1}^{m} \left| e^{-\|x_i-\tilde{\tau}\|^2/2\sigma^2} - e^{-\|x_i-\tau\|^2/2\sigma^2} \right|.$$

Choosing $\tilde{\tau} = \tau + \varepsilon u$ for any $u \in \mathbb{R}^d$, and taking $\varepsilon \searrow 0$, we obtain directional derivatives

$$e^{-\|x-\tau\|^2/2\sigma^2} \left| \frac{\langle x - \tau, u \rangle}{\sigma^2} \right| \quad \leq \quad \eta \max_{i=1}^{m} e^{-\|x_i-\tau\|^2/2\sigma^2} \left| \frac{\langle x_i - \tau, u \rangle}{\sigma^2} \right|.$$

Multiplying by $e^{\|\tau\|^2/2\sigma^2}/\sigma^2$ on both sides to complete the square, then rearranging, we have

$$\frac{1}{\eta} \big| \langle x - \tau, u \rangle \big| \quad \leq \quad \max_{i=1}^{m} \frac{e^{\|x\|^2/2\sigma^2}}{e^{\|x_i\|^2/2\sigma^2}} \, e^{\langle x_i - x, \tau \rangle/\sigma^2} \, \big| \langle x_i - \tau, u \rangle \big|.$$

for all $\tau \in \mathbb{R}^d$, $x \in \mathcal{X}$, and $u \in \mathbb{R}^d$. Next, we choose these three to suit our needs.

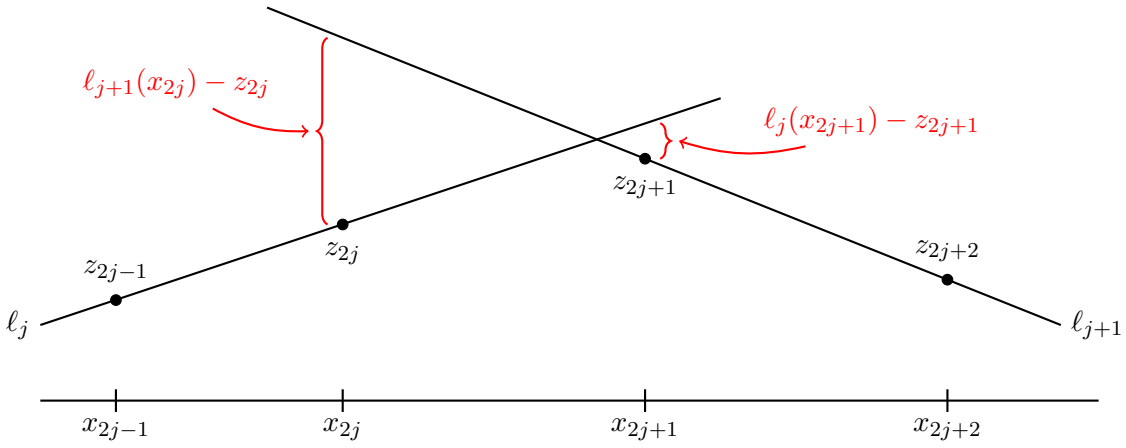
By assumption, we may choose $x_o \in \mathcal{X} \setminus \mathrm{cl}(\mathrm{conv}(\{x_1, \ldots, x_m\}))$. Then, 0 is not in the closed convex hull of $\{x_1 - x_o, \ldots, x_m - x_o\}$. So, the separating hyperplane theorem implies that there exists $\theta \in \mathbb{R}^d$ satisfying $\langle x_i - x_o, \theta \rangle > 0$ for all i. We will try $\tau = -N\theta$ for a very large value of $N > 0$. It remains to determine u, which we must choose so that $\langle x_i - \tau, u \rangle \neq 0$ for all $i = o, 1, \ldots, m$. Since $\|\tau\| = N\|\theta\|$ can be as large as desired, $u = \theta$ gets the job done for all sufficiently large N. Implementing these simplifications, we obtain

$$\frac{1}{\eta} \quad \leq \quad \left(\max_{i=1}^{m} \frac{e^{\|x_o\|^2/2\sigma^2}}{e^{\|x_i\|^2/2\sigma^2}} \right) \left(\max_{j=1}^{m} e^{-N\langle x_j - x_o, \theta \rangle/\sigma^2} \right) \left(\max_{\ell=1}^{m} \left| \frac{\langle x_\ell, \theta \rangle + N\|\theta\|^2}{\langle x_o, \theta \rangle + N\|\theta\|^2} \right| \right)$$

for all sufficiently large $N > 0$. Since $\langle x_j - x_o, \theta \rangle > 0$ for all j, we find $\eta = \infty$ as $N \to \infty$. ■

Appendix B. Supplementary Proofs on Optimization

Proof 15 Consider the lines $\ell_1, \ldots, \ell_k : \mathbb{R} \to \mathbb{R}$ with $\ell_j(x_{2j-1}) = z_{2j-1}$ and $\ell_j(x_{2j}) = z_{2j}$:



For ℓ_j and ℓ_{j+1} to intersect on $[x_{2j}, x_{2j+1}]$, the requirement is visibly that $\ell_{j+1}(x_{2j}) - z_{2j}$ and $\ell_j(x_{2j+1}) - z_{2j+1}$ have the same sign. In other words, $(\ell_{j+1}(x_{2j}) - z_{2j})(\ell_j(x_{2j+1}) - z_{2j+1}) \geq 0$. For brevity, call the first order differences $\widehat{x}_i \coloneqq x_{i+1} - x_i$ and $\widehat{z}_i \coloneqq z_{i+1} - z_i$. Then, (6) says

$$\ell_j(x) \quad = \quad z_{2j-1} \frac{x_{2j} - x}{\widehat{x}_{2j-1}} \; + \; z_{2j} \frac{x - x_{2j-1}}{\widehat{x}_{2j-1}}.$$

Plugging in $x = x_{2j+1}$, and noting for instance that $x_{2j+1} - x_{2j-1} = \widehat{x}_{2j} + \widehat{x}_{2j-1}$, we obtain

$$\ell_j(x_{2j+1}) \;-\; z_{2j+1} \;\;=\;\; z_{2j-1}\frac{-\widehat{x}_{2j}}{\widehat{x}_{2j-1}} \;+\; z_{2j}\frac{\widehat{x}_{2j}+\widehat{x}_{2j-1}}{\widehat{x}_{2j-1}} \;-\; z_{2j+1} \;\;=\;\; \widehat{z}_{2j-1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j-1}} \;-\; \widehat{z}_{2j}.$$

In a similar fashion, we also obtain

$$\ell_{j+1}(x_{2j}) \;-\; z_{2j} \;\;=\;\; z_{2j+1}\frac{x_{2j+2}-x_{2j}}{\widehat{x}_{2j+1}} \;+\; z_{2j+2}\frac{x_{2j}-x_{2j+1}}{\widehat{x}_{2j+1}} \;-\; z_{2j} \;\;=\;\; \widehat{z}_{2j} \;-\; \widehat{z}_{2j+1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j+1}}.$$

Hence,

$$\big(\ell_{j+1}(x_{2j}) - z_{2j}\big)\big(\ell_j(x_{2j+1}) - z_{2j+1}\big) \;\;=\;\; \left(\widehat{z}_{2j} \;-\; \widehat{z}_{2j+1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j+1}}\right)\left(\widehat{z}_{2j-1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j-1}} \;-\; \widehat{z}_{2j}\right)$$

By polarization $4ab = (-a-b)^2 - (a-b)^2$, non-negativity of the above is equivalent to

$$\left(\widehat{z}_{2j+1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j+1}} \;-\; \widehat{z}_{2j-1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j-1}}\right)^2 - \left(\widehat{z}_{2j+1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j+1}} \;+\; \widehat{z}_{2j-1}\frac{\widehat{x}_{2j}}{\widehat{x}_{2j-1}} \;-\; 2\widehat{z}_{2j}\right)^2 \;\;\geq\;\; 0.$$

Noticeably, the terms enclosed by either set of parentheses are linear in z. Thus, the constraint takes the form $(z^\intercal \beta^{(j)})^2 \geq (z^\intercal \alpha^{(j)})^2$ and, by re-expanding $\widehat{z}_i = z_{i+1} - z_i$, we can then read off the advertised formulas for $\alpha^{(j)}$ and $\beta^{(j)}$. ■

Proof 19 The lines $\ell_1, \ldots, \ell_k : \mathbb{R} \to \mathbb{R}$ with $\ell_j(x_{2j-1}) = z_{2j-1}$ and $\ell_j(x_{2j}) = z_{2j}$ are given by $\ell_j(x) \coloneqq a_j + s_j(x - m_j)$. If $s_{j+1} \neq s_j$, then we can solve to find that (17) is the unique point where $\ell_{j+1}(x) = \ell_j(x)$. In that case, since $z \in \Lambda(\Sigma^{(k)})$, this formula must give a value of τ_j satisfying $x_{2j-1} \leq \tau_j \leq x_{2j}$. If instead $s_{j+1} = s_j$, then there is simply not a breakpoint in $[x_{2j-1}, x_{2j}]$. The formula for $S(x)$ starts with the formula for $\ell_1(x)$, but switches the slope from s_j to s_{j+1} at τ_j, as in the proof of Theorem 24.1 in Foucart (2022). ■

Proof 21 Using that $(\max_i |v_i|)^2 = \sup_{w \in \Pi_m} \sum_i w_i v_i^2$ for any $v \in \mathbb{R}^m$, we obtain

$$\begin{aligned}\left(\inf_{S \in \Sigma^{(k)}} \max_{i=1}^{m} |y_i - S(x_i)|\right)^2 \;&=\; \inf_{z \,\in\, \Lambda(\Sigma^{(k)})} \left(\sup_{w \in \Pi_m} \sum_{i=1}^m w_i (y_i - z_i)^2\right) \\ &\geq\; \sup_{w \in \Pi_m} \left(\inf_{z \,\in\, \Lambda(\Sigma^{(k)})} \sum_{i=1}^m w_i (y_i - z_i)^2\right) \;\;\geq\; \sup_{w \in \Pi_m} d^*(w)\end{aligned}$$

where we define $d^*(w)$ to be the dual max of $\inf_{z \in \Lambda(\Sigma^{(k)})} \sum_j w_j (z_j - y_j)^2$. Lemma 18 states

$$d^*(w) \;\;=\;\; \sup\left\{\; t \;:\; \begin{pmatrix} \mathrm{diag}(w) + \sum_{j<k} \lambda_j A^{(j)} & -(w \odot y)^\intercal \\ -w \odot y & \sum_j w_j y_j^2 - t \end{pmatrix} \succeq 0 \;,\; \lambda_j \geq 0 \;\right\}.$$

To get the best dual bound, we can optimize in w. Conveniently, the constraint in our formula for $d^*(w)$ happens to be jointly linear in w, λ, and t! So, in the sprit of Lagrangian dual programming, we can can optimize over the penalty parameters $w_1, \ldots, w_m$ at the same time as we optimize for $t, \lambda_1, \ldots, \lambda_{k-1}$. Doing so, we find $\sup_{w \in \Pi_m} d^*(w) = b^*$. ■

References

Randall Balestriero and Richard G. Baraniuk. A spline theory of deep learning. volume 80 of *Proceedings of the 35th International Conference on Machine Learning*, pages 374–383. PMLR, 2018. URL https://proceedings.mlr.press/v80/balestriero18b.html.

Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019. URL https://jmlr.org/papers/v20/17-612.html.

Carl de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer New York, New York, NY, USA, 2001. URL https://link.springer.com/book/9780387953663.

Ronald DeVore, Simon Foucart, Guergana Petrova, and Przemysław Wojtaszczyk. Computing a quantity of interest from observational data. *Constructive Approximation*, 49: 461–508, 2019. URL https://doi.org/10.1007/s00365-018-9433-7.

Simon Foucart. *Mathematical Pictures at a Data Science Exhibition*. Cambridge University Press, Cambridge, UK, 2022. URL https://doi.org/10.1017/9781009003933.

Charles A. Micchelli and Theodore J. Rivlin. A survey of optimal recovery. In Charles A. Micchelli and Theodore J. Rivlin, editors, *Optimal Estimation in Approximation Theory*, The IBM Research Symposia Series, pages 1–54. Springer, Boston, MA, 1977. URL https://doi.org/10.1007/978-1-4684-2388-4_1.

Clarice Poon, Nicolas Keriven, and Gabriel Peyré. The geometry of off-the-grid compressed sensing. *Foundations of Computational Mathematics*, 23:241–327, 2023. URL https://doi.org/10.1007/s10208-021-09545-5.

Lloyd. N. Trefethen and J.A.C Weideman. Two results on polynomial interpolation in equally spaced points. *Journal of Approximation Theory*, 65(3):247–260, 1991. URL https://doi.org/10.1016/0021-9045(91)90090-W.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. volume 34 of *Advances in Neural Information Processing Systems 34*, pages 29909–29921. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fac7fead96dafceaf80c1daffeae82a4-Abstract.html.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. volume 31 of *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d04863f100d59b3eb688a11f95b0ae60-Abstract.html.