MARGIN, NOT DIMENSION, GOVERNS THE COMPLEXITY OF TRAINING A RELU

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a single rectified linear unit on n labeled points is NP-hard and W[1]-hard in the input dimension d. Under the Exponential Time Hypothesis (ETH), no algorithm runs in time $\varphi(d)n^{o(d)}$ for any computable φ. We study the geometric activation margin γ of an optimal neuron. If some optimum separates the sample with margin at least γ, squared-loss training on rational data has a global minimizer computable in time $n^{O(1/\gamma^2)}\mathrm{poly}(L)$, where L is the input bit length. The bound extends to convex coercive losses with polynomial-time exact solvers for fixed activation patterns. Its exponent is independent of d and optimal up to a constant factor: margin-promised training is W[1]-hard in $1/\gamma^2$ and, under ETH, admits no $\varphi(1/\gamma)n^{o(1/\gamma^2)}$ algorithm. For squared loss, no computable margin rate asymptotically above $1/\sqrt{d}$ holds uniformly across a family witnessing dimension hardness. Conversely, a computable uniform rate above $1/\sqrt{d}$ gives a $\varphi(d)n^{o(d)}\mathrm{poly}(L)$ algorithm.

1 INTRODUCTION

The rectified linear unit $\sigma(t) = \max\{0, t\}$ is widely used in modern machine learning, yet even its simplest training problem is difficult. Take a labeled sample of n points (x_i, y_i). Each consists of an input $x_i \in \mathbb{R}^d$ in dimension d and a label $y_i \in \mathbb{R}$, and the indices i range over $[n] = \{1, \ldots, n\}$. Consider fitting to it a single neuron $x \mapsto \sigma(w^\top x + b)$, with a weight vector $w \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$, so as to minimize the empirical error. Blum & Rivest (1992) initiated the complexity-theoretic study of neural network training. Even this single-neuron problem is NP-hard, and NP-hard to approximate within an almost polynomial factor (Manurangsi & Reichman, 2018; Goel et al., 2021). Which property of an instance makes training hard?

Input dimension is a natural suspect. The classical algorithm of Arora et al. (2018) solves the problem in time $n^{O(d)}$ by enumerating the ways a hyperplane can split the sample. Froese et al. (2022) showed that this dependence is essentially unavoidable: the problem is W[1]-hard with respect to d, and under the Exponential Time Hypothesis (ETH) no algorithm runs in time $n^{o(d)}$. In the language of parameterized complexity, recalled in Appendix A.1, W[1]-hardness means that unless the classes FPT and W[1] coincide, no algorithm runs in time $f(d)\,\mathrm{poly}(n)$ for any function f of the parameter alone. Read at face value, these results say that fitting a rectified linear unit is intractable precisely because the data lives in many dimensions, and later work has reinforced that impression (Froese & Hertrich, 2023; Bertschinger et al., 2023; Froese et al., 2026).

This reading sits uneasily beside practice, where high-dimensional models are trained routinely. The tension suggests that the instances behind the dimension lower bound are geometrically special, and it invites a sharper question, whether some structural quantity finer than the raw dimension controls the complexity. The kink of a rectified linear unit, the hyperplane $\{x : w^\top x + b = 0\}$, partitions the inputs into those on which the neuron fires and those it silences. We call the geometric separation across this hyperplane the *activation margin*, and this paper shows that the margin, and not the dimension, is the parameter that governs the complexity of training. The picture we obtain is complete in both directions: a margin promise removes the dimension from the exponent entirely, and the resulting running time cannot be improved to fixed-parameter time unless $\mathsf{FPT} = \mathsf{W}[1]$.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1.1 Our Results

Throughout, the sample is normalized so that every input lies in the unit ball, a rescaling that leaves the optimization unchanged and fixes the scale of the margin (Section 2.1), and $\gamma \in (0, 1]$ denotes a margin. A *loss* is a nonnegative function of a prediction and a label that is convex and coercive in the prediction, as Theorem 2.1 makes precise; every ℓ^p loss with $p \geq 1$ is one. To measure a margin, lift each input x_i to the point $(x_i, 1)$ of $\mathbb{R}^{d+1}$, and read the neuron (w, b) as a vector of the same space. The neuron separates the sample with margin γ when every lifted input lies at Euclidean distance at least γ from the hyperplane through the origin whose normal vector is (w, b); Theorem 2.5 makes this precise. An instance is γ-separated when some optimal neuron separates the sample in this sense, and $\gamma^\star(I)$ denotes the largest margin available at an optimum of the instance I. The first result removes the dimension from the exponent.

Theorem 1.1 (restated as Theorem 3.5)**.** *Let $\gamma \in (0, 1]$ and let L be the bit length of the rational input. On γ-separated instances under squared loss, single-neuron empirical risk minimization is solvable to global optimality in time $n^{O(1/\gamma^2)}\,\mathrm{poly}(L)$, an exponent that does not involve the dimension d, so for a constant margin the problem is solvable in polynomial time. The same conclusion holds for a convex coercive loss when each fixed-pattern convex program admits a polynomial-time exact solver.*

The engine is a dimension-free bound on the number of ways a large-margin neuron can split the sample. That bound is classical, and we recall it as Theorem A.6 rather than reprove it. The perceptron rule names every labeling of the sample realizable with margin γ by a signed sequence of at most $2/\gamma^2$ sample points, so there are at most $n^{O(1/\gamma^2)}$ such labelings (Novikoff, 1963; Ben-David et al., 2002; Moran et al., 2020). The proof of that count already supplies signed descriptions of the labelings; Theorem 3.1 explicitly enumerates them for use in training.

Under the Exponential Time Hypothesis, no algorithm improves the running time of Theorem 1.1 beyond the constant in the exponent.

Theorem 1.2 (restated as Theorem 5.1)**.** *Training a single rectified linear unit under an ℓ^p loss with $p \geq 1$, given the promise that an optimal neuron separates the sample with margin at least γ, is $\mathsf{W}[1]$-hard with respect to $1/\gamma^2$. Under the Exponential Time Hypothesis it admits no algorithm running in time $\varphi(1/\gamma) \cdot n^{o(1/\gamma^2)}$ for any computable φ.*

Because a large margin makes the problem easy, the instances that witness the dimension lower bound must have a small one, and the trade-off is quantitative.

Theorem 1.3 (restated as Theorem 4.1)**.** *Assume the Exponential Time Hypothesis. Consider a family of single-unit instances under squared loss that witnesses the $n^{o(d)}$ lower bound of Froese, Hertrich, and Niedermeier. No computable rate above $1/\sqrt{d}$ can bound the optimal-neuron margin from below throughout such a family. Conversely, the instances whose optimal margin is bounded below by a computable rate $\omega(1/\sqrt{d})$ are solvable in time $\varphi(d)n^{o(d)}\,\mathrm{poly}(L)$ for a computable φ, a running time that the worst case does not admit.*

The thread running through these results is that the algorithm's exponent is controlled by $1/\gamma^2$, and a uniform computable margin bound asymptotically above $1/\sqrt{d}$ would contradict dimension hardness.

1.2 Our Techniques

Once the activation pattern of the neuron is fixed, the training problem becomes convex, an observation due to Arora et al. (2018) and revisited through a convex lens by Pilanci & Ergen (2020). On the inputs the neuron fires on, its output is the affine function $w^\top x + b$; on the rest it is the constant zero. The empirical loss restricted to one pattern is therefore a convex function minimized over a polyhedron. The remaining difficulty is combinatorial: without a margin promise, the number of candidate patterns can grow as $n^{\Theta(d)}$ when d is fixed and n grows (Cover, 1965), placing the dimension in the exponent.

A margin breaks the dependence, because a separator of margin γ is robust and robustness has small description length. Run the perceptron rule on the sample, labeled by the pattern in question, with

the separating neuron as a witness. By the mistake bound of Novikoff (1963) the run halts after at most $2/\gamma^2$ updates, and it outputs a signed sum of the lifted inputs on which it updated. Every margin-γ pattern is therefore *named* by a short signed sequence of sample points. This compression phenomenon was turned into a count by Moran et al. (2020), who credit it to Ben-David et al. (2002). Enumerating the signed descriptions implicit in the counting proof supplies the list; one convex program per candidate then proves Theorem 3.5. The enumeration never inspects the labels, and it never inspects the margin except through the length bound.

A parameterized reduction to the promise problem must output, along with the instance, a margin promise bounded below by a function of the source parameter alone, so the hard instances it produces must be well separated relative to that parameter. The source problem is multicolored clique: given a graph whose vertices carry k colors, decide whether it has a clique with one vertex of each color. A clique is *colorful* when its vertices carry pairwise distinct colors, so the question asks for a colorful clique of size k. The reduction enforces adjacency and distinct colors with the midpoint gadget of Froese et al. (2022). The gadget forces a neuron that activates two vertices of the graph to activate their midpoint as well. It thereby penalizes a neuron that selects a pair of vertices that no colorful clique contains.

We embed the gadget at orthonormal points, one coordinate per vertex. The neuron that selects a colorful clique of size k then has a lifted weight vector of squared norm $16k + 9$, and it gives every sample point a preactivation of absolute value at least one. Its margin is therefore $\Theta(1/\sqrt{k})$, a function of the parameter alone, and the gadget's argument about adjacency and colors is untouched by the change of embedding. Theorem 4.1 then needs no new construction. For squared loss, if the instances witnessing the dimension lower bound had margin bounded below by a computable rate above $1/\sqrt{d}$, then Theorem 3.5 would solve them in time $\varphi(d)n^{o(d)}\,\mathrm{poly}(L)$, which ETH forbids.

1.3 Other Related Work

Beyond the single-neuron results cited above, two-layer training is NP-hard already in dimension two when the number of units grows, and W[1]-hard in the dimension for four units (Froese & Hertrich, 2023). Exact training of a two-layer network with two output neurons lies outside NP under standard assumptions (Bertschinger et al., 2023). On the positive side, Brand et al. (2023) identified architectures for which training is polynomial, and Khalife et al. (2024) gave algorithms for threshold networks that are polynomial at a fixed dimension and architecture. In the bit model of finite-precision arithmetic, training with piecewise-linear activations such as the rectifier lies in NP (Doron-Arad & Mossel, 2026). For two-layer networks with weight decay, a convex relaxation approximates the global optimum within a factor $O(\sqrt{\log n})$ on random data (Kim & Pilanci, 2024). The $n^{O(d)}$ enumeration of Arora et al. (2018) was extended to concave losses by Froese et al. (2022) and given a polyhedral reading by Matena & Raffel (2022). For weight-decayed two-layer networks, Pilanci & Ergen (2020) gave exact convex formulations, whose bounded-rank tractability was developed further by Ergen & Pilanci (2025). All of these parameterize by the ambient geometry or the architecture, and none parameterizes by a margin.

The quantity $1/\gamma^2$ is the central parameter of large-margin learning. It bounds the mistake number of the perceptron (Novikoff, 1963) and the Vapnik–Chervonenkis dimension of the partial concept class of margin-γ halfspaces in a unit ball, which Alon et al. (2021) determine to be $\Theta(1/\gamma^2)$ when the ambient dimension is at least $1/\gamma^2$. In online learning of networks with sign activations, a margin at every neuron of the first hidden layer does not by itself remove the dimension from the mistake bound, while a margin at every neuron of every layer does (Daniely et al., 2025). Our contribution is not to the count, which is that of Moran et al. (2020), but to its computational reading. Three precedents deserve emphasis. Arriaga & Vempala (2006) showed that a robust concept survives a random projection to $O(\log n/\gamma^2)$ dimensions, so that the ambient dimension can be traded for the margin in a learning algorithm.

Closer to our setting, Ben-David & Simon (2000) relaxed agnostic halfspace learning by discounting the points inside a μ-margin, and gave, for each fixed μ, a polynomial-time algorithm ranging over all subsamples of size at most $1 + \lceil 4/\mu^4 \rceil$. They complemented it with the statement that no algorithm polynomial in both the sample size and $1/\mu$ can succeed for all μ unless P = NP. Their algorithm and ours share the shape of an exhaustive search over small subsamples whose exponent

depends on the margin alone. For intersections of halfspaces, Gottlieb et al. (2022) instead use random projection and a search over candidate halfspaces to obtain margin-dependent running time.

Nearest of all is the work of Diakonikolas et al. (2019), who settle the complexity of properly learning γ-margin halfspaces agnostically as a function of $1/\gamma$. They give a 1.01-approximate learner in time $2^{\tilde{O}(1/\gamma^2)}\,\mathrm{poly}(d, 1/\varepsilon)$, and show that no algorithm solves the exact version in time $f(1/\gamma)\,\mathrm{poly}(d, 1/\varepsilon)$ unless $\mathsf{W}[1]$ is contained in randomized FPT. Parameterized hardness in a margin is therefore not new; what is new is the problem and the shape of the answer. Their bounds are fixed-parameter in $1/\gamma$ for classification under the zero-one loss, where the sample may be subsampled, whereas Theorems 3.5 and 5.1 pin exact rectified-linear-unit fitting on an adversarial sample at $n^{\Theta(1/\gamma^2)}$ from both sides. We are not aware of any exact algorithm for rectified-linear-unit training whose running time is parameterized by a margin, prior to this work.

Under Gaussian inputs, Chen et al. (2021) learn an unknown deep network in time polynomial in the dimension and exponential in network parameters. For bounded-weight depth-two networks, Goel et al. (2021) give proper learning algorithms with running times exponential in the number of units and inverse accuracy. These are learning results, whereas our guarantee concerns exact optimization on a fixed sample. Under Gaussian marginals, a single rectified linear unit with an arbitrary bias is learned in polynomial time to loss $O(\mathrm{OPT}) + \varepsilon$, where OPT is the least loss of any neuron (Guo & Vijayaraghavan, 2025). Under the same marginals, agnostic ReLU regression obeys near-optimal cryptographic lower bounds (Diakonikolas et al., 2023). Inputs on a manifold of bounded curvature keep learning hard unless the manifold's volume is controlled as well (Kiani et al., 2024).

The parameterized framework we use is standard (Downey & Fellows, 2013; Cygan et al., 2015; Impagliazzo & Paturi, 2001), and Ganian (2026) surveys its use in machine learning. The closest neighbors of our lower bound in that literature are the $\mathsf{W}[1]$-hardness results for zonotope containment and network verification (Froese et al., 2026). Close to our lower bound as well are the parameterized complexity of training quantized networks (Ganian et al., 2026) and that of testing stationarity of piecewise-affine losses (Ye, 2026). All of them take the input dimension as a parameter, alone or together with other parameters. The binary case of quantized training is $\mathsf{W}[1]$-hard even in the input and output dimensions together (Jiang et al., 2026).

1.4 Discussion

Theorems 3.5 and 4.1 together reinterpret the dimension lower bound. The $\mathsf{W}[1]$-hardness of Froese et al. (2022) is genuine, but it is not a statement about high dimensions as such; it is a statement about instances that are badly separated relative to their dimension. For squared loss, Theorem 4.1 says that no hard family has a uniform computable margin lower bound asymptotically above $1/\sqrt{d}$; on instances with such a bound, the dimension leaves the exponent. The margin is therefore the relevant complexity parameter for these bounds, and it explains both halves of the picture. The worst case is hard because margins can be driven to zero, and structured instances are easy because their margins are bounded away from zero. The guarantee is also exact and deterministic, so it speaks to optimization rather than to generalization. The parameter $1/\gamma^2$ is the same quantity that governs the statistical behavior of large-margin classifiers, which suggests that separation yields tractability on the computational and the statistical side at once. We do not claim that this tractability explains the empirical success of gradient methods. Theorem 3.5 is an existence result for an enumeration algorithm, and the relationship between the margin and the trajectories of local search is left untouched.

The lower bound qualifies the algorithmic result. A margin promise controls the exponent through $1/\gamma^2$, and Theorem 5.1 shows that the price $n^{\Theta(1/\gamma^2)}$ is the right one, up to the constant in the exponent.

The two hardness results, ours and that of Froese et al. (2022), inhabit orthogonal regimes. Their reduction places its instances in dimension $2k$ for a source instance with k colors, ours at margin $\Theta(1/\sqrt{k})$ and in dimension equal to the number of vertices, so neither subsumes the other. This paper confines itself to the three results above. The full version carries five further results on the same enumeration. Two of them extend the upper bound: one to the case where the optimum leaves a bounded number of inputs inside the margin, and one to an algorithm given no margin bound at all. A third extends both bounds to a sum of several units under a joint margin. The other two are

a fixed-parameter bound in the intrinsic dimension of the data and an unconditional lower bound against algorithms that enumerate activation patterns from the inputs alone.

1.5 Open Problems

The most obvious question, whether the running time of Theorem 3.5 can be improved to fixed-parameter time, is already answered. Theorem 3.5 places the problem in XP, the class of problems solvable in time $n^{f(\lambda)}$ for a parameter λ, and Theorem 5.1 rules out a fixed-parameter algorithm unless $\mathsf{FPT} = \mathsf{W}[1]$.

One question remains, and it meets a recognizable wall. Multiplicative approximation at a bounded margin is open, and the two natural routes to a hardness proof fail for the same reason. The almost polynomial inapproximability of Goel et al. (2021) is proved with no norm bound and plants a polynomially small target, so the margin of its witness vanishes. Their densest-subgraph gadget, which does carry a margin, separates its two cases only additively. Our own instances, built from a graph on N vertices with a largest colorful clique of size $k^\star$, have optimum $N - k^\star$ by Theorem C.2. A yes-instance, where $k^\star$ equals the number k of colors, therefore has optimum $N - k$, and a no-instance has optimum at least $N-k+1$. The two optima can differ by a factor of only $1+1/(N-k)$, and the identically zero neuron, whose loss is N, already approximates both within $1 + k/(N - k)$. The reduction of Wang & Pilanci (2023) from maximum cut does separate its cases multiplicatively, but for two-layer training with weight decay and no margin.

On the algorithmic side, a positive result would follow from a coreset preserving the loss of every margin-γ neuron, and we do not know whether one exists. For worst-case data, Dey et al. (2020) reach a multiplicative factor n/r in time $O\left(n^r\right)$ for a fixed integer r, without using a margin promise.

1.6 Paper Organization

Section 2 fixes notation and formalizes the training problem and the margin. Section 3 turns the counting bound into an enumeration and states Theorem 3.5, Section 4 states Theorem 4.1, and Section 5 states the $\mathsf{W}[1]$-hardness in the margin parameter. The appendices follow the order in which the main text first refers to them. Appendix A states the results imported from the literature, Appendix B holds two remarks placing our construction and our threshold beside their nearest neighbors, and Appendix C gives the proofs of every statement in the main text.

2 Preliminaries

This section states the training problem and the margin promise as formal objects, and introduces each symbol where it is first needed. The parameterized-complexity vocabulary and the facts about large-margin separation that the proofs invoke are collected in Appendix A, each stated as a numbered assertion and cited to its source.

2.1 Training a Rectified Linear Unit

A *sample* is a finite sequence $S = (x_i, y_i)_{i\in[n]}$ with inputs $x_i \in \mathbb{R}^d$ and labels $y_i \in \mathbb{R}$. Throughout, the letter n denotes the number of sample points, $[n] = \{1, \dots, n\}$ is the set of their indices, and the letter d denotes the dimension of the inputs. We assume $n \geq 2$, since a sample with one input is fitted by minimizing the convex function $\ell(\cdot, y_1)$ over $[0, \infty)$, with ℓ the loss of Theorem 2.1 below. Scaling every input by a common factor $c > 0$ and the weight vector by $1/c$ leaves each value $w^\top x_i + b$, and hence the output of the neuron and the loss, unchanged. We may therefore assume without loss of generality that every input has Euclidean norm $\|x_i\| \leq 1$, and we do so throughout. The normalization leaves the optimization problem unchanged, but it fixes the scale at which the margin of Theorem 2.5 is measured, since the lift of Section 2.2 appends the constant 1 to every input. Computational running-time bounds use rationally encoded inputs and labels; L denotes the total bit length of an algorithm's input, including a description of the loss and an integer enumeration bound when supplied.

Definition 2.1. A *single rectified linear unit* with weight $w \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$ is the function $h_{w,b} : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ given by $h_{w,b}(x) = \sigma\left(w^\top x + b\right)$. A *loss* is a function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ such

that $\ell(\cdot, y)$ is convex and coercive for every y, the latter meaning that $\ell(\hat{y}, y) \to \infty$ as $|\hat{y}| \to \infty$. The *empirical loss* of the neuron on the sample S is

$$\widehat{L}(w, b) = \sum_{i \in [n]} \ell\left(h_{w,b}(x_i), y_i\right).$$

Coercivity is a convenience rather than a restriction, used only to guarantee that the minima appearing below are attained rather than approached. Every ℓ^p loss $\ell(\hat{y}, y) = |\hat{y} - y|^p$ with $p \geq 1$, and in particular the squared loss, is convex and coercive.

Problem 2.2. In *single-unit empirical risk minimization* the input is a sample S together with a loss ℓ, and the goal is to compute a pair $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ minimizing $\widehat{L}(w, b)$.

Two remarks on the statement are in order. First, we state the problem as an optimization problem. Its decision version, in which a threshold τ is given and one asks whether $\widehat{L}(w, b) \leq \tau$ is achievable, is the version used in the hardness statements, and an algorithm for the optimization version decides it. Second, the minimum of $\widehat{L}$ is attained, although $\widehat{L}$ is not convex. The parameter space splits into the finitely many regions on which the activation pattern of the neuron, defined in Theorem 2.4 below, is constant. On the region of pattern A the value of $\widehat{L}$ is bounded below by the minimum of the convex program Φ_A of Theorem 3.3 below, and that minimum is attained and is itself the value of $\widehat{L}$ at a neuron. The minimum of $\widehat{L}$ is therefore the least of these finitely many attained values.

Remark 2.3. The combinatorial difficulty studied here arises from a loss that decreases on the rectifier's range. A single rectified linear unit $\sigma\left(w^\top x + b\right)$, and more generally a sum $\sum_j \sigma\left(w_j^\top x + b_j\right)$, is a convex function of the weights and the biases, being a sum of maxima of affine functions.

Note that a loss cannot be nondecreasing on all of $\mathbb{R}$. A nondecreasing function is bounded on $(-\infty, 0]$ by its value at 0, whereas coercivity forces the loss to grow without bound as its first argument tends to $-\infty$. But only the behavior of the loss on $[0, \infty)$, the range of the rectifier, ever enters $\widehat{L}$.

Suppose now that $\ell(\cdot, y)$ is nondecreasing on $[0, \infty)$. Replace it by the function that agrees with it on $[0, \infty)$ and is constant on $(-\infty, 0]$. That function is convex and nondecreasing, and the replacement leaves $\widehat{L}$ unchanged, since only values on $[0, \infty)$ enter. It exhibits $\widehat{L}$ as a sum of convex nondecreasing functions, each composed with the convex function $(w, b) \mapsto \sigma\left(w^\top x_i + b\right)$ noted above. Thus $\widehat{L}$ is convex and can be minimized with no margin assumption whenever this convex program has a polynomial-time solver. The same holds for any number of units with nonnegative output weights, each absorbed into its unit by $a\sigma(t) = \sigma(at)$ for $a \geq 0$. The enumeration, the margin parameter, and the hardness results of this paper therefore all concern losses that decrease somewhere on $[0, \infty)$, which is the case of interest. The squared loss and every other ℓ^p loss with $p \geq 1$ decrease on $[0, y)$ whenever the label y is positive.

2.2 The Activation Margin

The combinatorial content of Theorem 2.2 is which inputs the neuron fires on.

Definition 2.4. The *activation pattern* of the neuron (w, b) on the sample S is the set

$$A(w, b) = \left\{i \in [n] \mid w^\top x_i + b > 0\right\}$$

of indices on which the neuron fires. Equivalently, it is the *labeling* $\left(\operatorname{sign}\left(w^\top x_i + b\right)\right)_{i \in [n]}$, a vector in $\{+1, -1\}^n$ with one sign for each sample point. Here $\operatorname{sign}(t) \in \{+1, -1\}$ is the sign of a real number t, with the convention that $\operatorname{sign}(t) = +1$ exactly when $t > 0$, so that $\operatorname{sign}(0) = -1$.

It is convenient to make the bias into a coordinate. For an input x_i we write $\tilde{x}_i = (x_i, 1) \in \mathbb{R}^{d+1}$ for its *lift*, and for a neuron (w, b) we write $\tilde{w} = (w, b) \in \mathbb{R}^{d+1}$, so that

$$w^\top x_i + b = \tilde{w}^\top \tilde{x}_i, \qquad \|\tilde{x}_i\| = \sqrt{\|x_i\|^2 + 1} \leq \sqrt{2}.$$

The activation pattern is thus the sign pattern of a homogeneous halfspace applied to the lifted inputs, and the margin is the geometric separation across the corresponding hyperplane through the origin of $\mathbb{R}^{d+1}$.

Definition 2.5. Let $\gamma > 0$. A neuron (w, b) with $\tilde{w} \neq 0$ *separates the sample S with margin γ* if

$$\left|\tilde{w}^\top \tilde{x}_i\right| \geq \gamma \left\|\tilde{w}\right\| \qquad \text{for every } i \in [n],$$

equivalently, if every lifted input lies at Euclidean distance at least γ from the hyperplane through the origin with normal vector $\tilde{w}$. The neuron $(w, b) = (0, 0)$ is declared not to separate the sample with any margin. An instance of Theorem 2.2 is γ-*separated* if it admits an optimal solution that separates the sample with margin at least γ, and we write

$$\gamma^\star(I) = \sup\{\gamma > 0 \mid \text{some optimal solution of the instance } I \text{ separates its sample with margin } \gamma\}$$

for the largest margin available at an optimum, the supremum of the empty set being 0. Let $\chi \in \{+1, -1\}^n$ be a labeling. It is *realizable with margin γ* on S if some nonzero vector $\tilde{w}$ puts every lifted input on the side named by that input's sign in χ, at distance at least γ. In symbols, $\chi_i \left(\tilde{w}^\top \tilde{x}_i\right) \geq \gamma \left\|\tilde{w}\right\|$ for every $i \in [n]$.

Three comments fix the meaning of the promise that an instance is γ-separated. Since $\|\tilde{x}_i\| \leq \sqrt{2}$ and $\left|\tilde{w}^\top \tilde{x}_i\right| \leq \|\tilde{w}\| \, \|\tilde{x}_i\|$, no sample is separated with margin larger than $\sqrt{2}$, and restricting attention to $\gamma \in (0, 1]$ costs nothing. An input with $w^\top x_i + b = 0$ receives the output $\sigma(0) = 0$ whichever side it is assigned to, so a γ-separated instance is one whose optimum classifies every input strictly, away from the kink. Finally, the promise constrains the optimum and not the sample: it is a promise about the solution, which is what makes it a parameter of the optimization problem rather than of the data alone.

3 Training at a Large Margin

This section develops Theorem 3.5, whose proof, with those of its lemmas, is in Appendix C. The plan is the one sketched in Section 1.2: turn the counting bound of Theorem A.6 into an explicit list of candidate activation patterns, and solve one convex program for each candidate.

3.1 Enumerating the Large-Margin Activation Patterns

The proof of Theorem A.6 names every large-margin labeling by a short signed sequence of sample points. The following lemma spells out the enumeration implicit in the counting proof, which the training algorithm uses.

Lemma 3.1 (proved in Appendix C)**.** *Let $\gamma \in (0, 1]$ and let M be an integer with $M \geq 2/\gamma^2$. Let L be the total bit length of a rationally encoded sample S and the integer M. An algorithm outputs a list $\mathcal{P} \subseteq \{+1, -1\}^n$ of at most $(2n)^{M+1}$ labelings. It runs in time $(2n)^{M+1} \operatorname{poly}(L)$. Every labeling realizable with margin γ on S belongs to this list.*

The length of the list matches the count that the lemma makes constructive, up to the constant in the exponent.

Remark 3.2. Taking cardinalities in Theorem 3.1 with $M = \lceil 2/\gamma^2 \rceil$ recovers the bound of Theorem A.6 up to the value of the constant in the exponent, since $\gamma \leq 1$ gives $M + 1 \leq 4/\gamma^2$ and hence $(2n)^{M+1} \leq n^{8/\gamma^2}$ for $n \geq 2$. The bound n^{8/γ^2} makes no reference to the dimension d. Both the count and its signed descriptions come from the perceptron argument; our use of the descriptions is to generate candidate training patterns. The list is computed from the inputs x_i and the margin alone, never consulting the labels y_i, and the margin enters only through the length bound M, which Theorem 4.1 uses.

3.2 Freezing a Pattern Makes the Problem Convex

For an active set $A \subseteq [n]$, the optimization problem Φ_A minimizes the objective

$$\sum_{i \in A} \ell\left(w^\top x_i + b, y_i\right) + \sum_{i \notin A} \ell(0, y_i) \tag{1}$$

over the pairs $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ that satisfy the constraints $w^\top x_i + b \geq 0$ for $i \in A$ and $w^\top x_i + b \leq 0$ for $i \notin A$.

Algorithm 1 Large-margin training of a single rectified linear unit.

Require: Rationally encoded sample $S = (x_i, y_i)_{i \in [n]}$ with $\|x_i\| \leq 1$; loss ℓ and an exact solver for Φ_A; integer $M \geq 1$.

Ensure: A neuron that, for every $\gamma \in (0, 1]$ with $\lceil 2/\gamma^2 \rceil \leq M$, has empirical loss at most that of every neuron whose activation pattern is realizable with margin γ.

```
best ← +∞; (w*, b*) ← (0, 0)
lift each input to x̃_i ← (x_i, 1) ∈ R^{d+1}
for each name ν = ((i_1, s_1), ..., (i_T, s_T)) with T ≤ M, i_t ∈ [n] and s_t ∈ {+1, −1} do
    u ← Σ_{t=1}^{T} s_t x̃_{i_t}
    A ← { i ∈ [n] | u^⊤ x̃_i > 0 }
    solve the convex program Φ_A of Eq. (1); let v be its value and (w, b) a minimizer
    if v < best then
        best ← v; (w*, b*) ← (w, b)
return (w*, b*)
```

Lemma 3.3 (proved in Appendix C)**.** *Let ℓ be a loss in the sense of Theorem 2.1 and let $A \subseteq [n]$. Then the following hold.*

(i) The problem Φ_A is a convex program: its objective is a convex function of (w, b) and its feasible region is a nonempty polyhedron, and its minimum is attained.

(ii) For every feasible (w, b) of Φ_A, the objective value of Φ_A at (w, b) equals the empirical loss $\widehat{L}(w, b)$.

(iii) If a neuron (w, b) has activation pattern $A(w, b) = A$, then (w, b) is feasible for Φ_A.

The algorithm of Section 3.3 solves one program Φ_A for each candidate pattern, so the cost of solving one program enters the running time of the algorithm as a factor.

Remark 3.4. The enumeration works for every loss of Theorem 2.1, but its running time includes the cost of minimizing Φ_A. For rational data and squared loss, Φ_A is a convex quadratic program over a polyhedron and is solvable exactly in time polynomial in the input bit length L (Kozlov et al., 1980). The same polynomial bound applies to the ℓ^1 loss, whose program can be written as a linear program. For a general convex coercive loss, an exact running-time bound requires an explicit representation and an exact solver for every Φ_A; convexity alone does not supply either. If a solver instead returns a feasible neuron within ε of the minimum of each Φ_A, the enumeration returns a neuron within ε of the global minimum on γ-separated instances, with the solver's accuracy-dependent cost.

3.3 The Algorithm and Its Analysis

Algorithm 1 assembles the two ingredients.

Theorem 3.5 (restatement of Theorem 1.1; proved in Appendix C)**.** *Let ℓ be a loss in the sense of Theorem 2.1 and let $\gamma \in (0, 1]$. Suppose each program Φ_A can be solved exactly in time $T_{\mathrm{cvx}}(L)$, uniformly over A, where L is the input encoding length. On γ-separated instances, Algorithm 1 run with an integer $M \geq 2/\gamma^2$ returns a global minimizer in time $(2n)^{M+1}\,(T_{\mathrm{cvx}}(L) + \mathrm{poly}(L))$. For rational data and the squared or ℓ^1 loss, $T_{\mathrm{cvx}}(L) = \mathrm{poly}(L)$, so with $M = \lceil 2/\gamma^2 \rceil$ the running time is $n^{O(1/\gamma^2)}\,\mathrm{poly}(L)$. The exponent is independent of d; for every constant γ these losses are solvable in polynomial time.*

4 Localizing the Dimension Hardness

The algorithm of Section 3 interacts with the dimension lower bound of Theorem A.3 to constrain the margins of hard families. The idea is a single contradiction: a uniform computable margin bound asymptotically above $1/\sqrt{d}$ would let the algorithm of Theorem 3.5 run faster than ETH permits on the hard family.

Recall from Theorem 2.5 that $\gamma^\star(I)$ denotes the largest margin available at an optimum of the instance I, defined there as a supremum.

Theorem 4.1 (restatement of Theorem 1.3; proved in Appendix C)**.** *Fix $p \in \{1, 2\}$ and assume ETH. Let $(\mathcal{I}_d)_{d\in\mathbb{N}}$ be a family of rationally encoded instances of Theorem 2.2 under the ℓ^p loss, the instances in $\mathcal{I}_d$ having dimension d, that witnesses the lower bound of Theorem A.3 even allowing polynomial factors in the input bit length L. That is, no algorithm solves every $I \in \bigcup_d \mathcal{I}_d$ in time $\varphi(d) \cdot n^{o(d)} \operatorname{poly}(L)$ for a computable φ. Then there is no computable nondecreasing function $\psi\colon \mathbb{N} \to \mathbb{Q}$ with $\psi(d) \to \infty$ such that*

$$\frac{1}{\gamma^\star(I)^2} \le \frac{d}{\psi(d)} \qquad \text{for every } I \in \mathcal{I}_d \text{ and every sufficiently large } d,$$

where $1/\gamma^\star(I)^2$ is read as ∞ when $\gamma^\star(I) = 0$. In this sense no computable function witnesses an optimal-neuron margin asymptotically above $1/\sqrt{d}$ throughout such a family. Conversely, let $\lambda\colon \mathbb{N} \to \mathbb{N}$ be a computable function with $\lambda(d) = o(d)$. Consider an instance of dimension d that is γ-separated for some $\gamma \in (0, 1]$ with $1/\gamma^2 \le \lambda(d)$. It is solvable in time $\varphi(d)n^{o(d)} \operatorname{poly}(L)$ for a computable φ, strictly below the worst case that Theorem A.3 mandates.

One hypothesis of Theorem 4.1 is easy to take for a formality, and it is not one.

Remark 4.2. The hypothesis that the rate ψ be computable and rational-valued is used once, and the argument does not survive without it. The proof runs Algorithm 1 in dimension d with the length bound $M(d) = \lceil 2d/\psi(d) \rceil + 1$. Computability makes $M(d)$ computable from d, and rational values let the ceiling be computed exactly. The proof therefore produces a single algorithm for every dimension. Without the hypothesis the same proof yields, for each dimension d separately, an algorithm running in time $\varphi(d)n^{g(d)} \operatorname{poly}(L)$ with $g(d) = o(d)$, but no single algorithm. A separate algorithm for each dimension contradicts nothing, since the lower bound of Theorem A.3, like every lower bound under ETH, speaks of one algorithm for all dimensions.

5 Lower Bounds in the Margin Parameter

For squared and absolute loss, Theorem 3.5 places the large-margin problem in XP with respect to $1/\gamma^2$, that is, it solves the problem in time $n^{f(1/\gamma^2)} \operatorname{poly}(L)$ for a function f of the parameter alone. This section shows W[1]-hardness in that parameter and an ETH lower bound matching the squared- and absolute-loss exponents of Theorem 3.5 up to constants. A sample on which many margin-γ patterns are realizable is not by itself a hard instance: what a hard instance needs is that the patterns be coupled to one another, and coupling is what the construction adds.

The Construction. The coupling is supplied by the midpoint gadget of Froese et al. (2022). For every pair of vertices that no colorful clique can contain, a point midway between the two vertex points is added with label zero. A neuron that activates both vertices then activates their midpoint as well, and so incurs a loss on a point labeled zero. We re-embed the gadget at orthonormal points, one coordinate per vertex, which is what gives the witness neuron a margin $\Theta(1/\sqrt{k})$ in the number k of colors and makes the margin, rather than the dimension, the parameter the reduction transfers. Appendix C gives the construction in full, beside the lemma that computes the optimum of its instances.

Theorem 5.1 (restatement of Theorem 1.2; proved in Appendix C)**.** *Let $p \ge 1$ be a constant, and consider training a single rectified linear unit under the ℓ^p loss, given the promise that some optimal neuron separates the sample with margin at least γ. This problem is W[1]-hard with respect to the parameter $1/\gamma^2$. Moreover, under ETH it admits no algorithm running in time $\varphi(1/\gamma) \cdot n^{o(1/\gamma^2)}$ for any computable function φ. For $p \in \{1, 2\}$, this matches the exponent of Theorem 3.5 up to a constant factor. For every fixed $p \ge 1$, the large-margin problem is not fixed-parameter tractable in $1/\gamma^2$ unless FPT = W[1].*

The dimension and margin reductions occupy different regimes; Theorem B.3 compares them with Theorem 4.1.

AI USE STATEMENT

Claude (Anthropic) and Codex (OpenAI) assisted with developing and checking proofs, searching and verifying relevant literature, and drafting and editing the manuscript. Codex also assisted with the submission build script. The authors framed the problem, directed the work, reviewed the AI-assisted material, and take responsibility for the paper and any errors.

Under the ICLR 2027 AI Policy for Authors, theoretical model development, claim formulation, proof contributions, and proof writing require disclosure; drafting, identifying prior work, formatting references, and editing the build script are recommended for disclosure. No translation was used, and tasks involving data, experiments, or implementing research methods do not apply.

REPRODUCIBILITY STATEMENT

All results are theoretical; no experiments, datasets, or research code are involved. The sample normalization, loss assumptions, and margin promise are stated in the main text. The algorithmic and hardness arguments are proved in the body and appendices, which also state the external results they use with citations.

REFERENCES

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 658–671. IEEE, 2021. doi: 10.1109/FOCS52979.2021.00070. Full version: arXiv:2107.08444.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018. arXiv:1611.01491.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006. doi: 10.1007/s10994-006-6265-7.

Shai Ben-David and Hans Ulrich Simon. Efficient learning of linear perceptrons. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pp. 189–195. MIT Press, 2000.

Shai Ben-David, Nadav Eiron, and Hans Ulrich Simon. Limitations of learning via embeddings in Euclidean half spaces. *Journal of Machine Learning Research*, 3:441–461, 2002.

Daniel Bertschinger, Christoph Hertrich, Paul Jungeblut, Tillmann Miltzow, and Simon Weber. Training fully connected neural networks is $\exists\mathbb{R}$-complete. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. arXiv:2204.01368.

Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–127, 1992. doi: 10.1016/S0893-6080(05)80010-3.

Cornelius Brand, Robert Ganian, and Mathis Rocton. New complexity-theoretic frontiers of tractability for neural network training. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. arXiv:2607.20811.

Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006. doi: 10.1016/j.jcss.2006.04.007.

Sitan Chen, Adam R. Klivans, and Raghu Meka. Learning deep ReLU networks is fixed-parameter tractable. In *62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 696–707. IEEE, 2021. doi: 10.1109/FOCS52979.2021.00073. Full version: arXiv:2009.13512.

Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965. doi: 10.1109/PGEC.1965.264137.

Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi: 10.1007/978-3-319-21275-3.

Amit Daniely, Idan Mehalel, and Elchanan Mossel. Online learning of neural networks. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025. arXiv:2505.09167.

Santanu S. Dey, Guanyi Wang, and Yao Xie. Approximation algorithms for training one-node ReLU neural networks. *IEEE Transactions on Signal Processing*, 68:6696–6706, 2020. doi: 10.1109/TSP.2020.3039360.

Ilias Diakonikolas, Daniel M. Kane, and Pasin Manurangsi. Nearly tight bounds for robust proper learning of halfspaces with a margin. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019. Full version: arXiv:1908.11335.

Ilias Diakonikolas, Daniel M. Kane, and Lisheng Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and ReLU regression under Gaussian marginals. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7922–7938, 2023.

Ilan Doron-Arad and Elchanan Mossel. Why ReLU? A bit-model dichotomy for deep network training, 2026.

Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi: 10.1007/978-1-4471-5559-1.

Tolga Ergen and Mert Pilanci. The convex landscape of neural networks: Characterizing global optima and stationary points via Lasso models. *IEEE Transactions on Information Theory*, 71(5): 3854–3870, 2025. doi: 10.1109/TIT.2025.3545564.

Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1): 53–61, 2009. doi: 10.1016/j.tcs.2008.09.065.

Vincent Froese and Christoph Hertrich. Training neural networks is NP-hard in fixed dimension. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. arXiv:2303.17045.

Vincent Froese, Christoph Hertrich, and Rolf Niedermeier. The computational complexity of ReLU network training parameterized by data dimensionality. *Journal of Artificial Intelligence Research*, 74:1775–1790, 2022. doi: 10.1613/jair.1.13547.

Vincent Froese, Moritz Grillo, Christoph Hertrich, and Moritz Stargalla. Parameterized hardness of zonotope containment and neural network verification. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.22849.

Robert Ganian. Parameterized complexity in machine learning. *Computer Science Review*, 59: 100836, 2026. doi: 10.1016/j.cosrev.2025.100836.

Robert Ganian, Frank Sommer, and Manuel Sorge. Tractability via low dimensionality: The parameterized complexity of training quantized neural networks. In *International Conference on Learning Representations (ICLR)*, 2026. OpenReview BAQNrsr987.

Surbhi Goel, Adam Klivans, Pasin Manurangsi, and Daniel Reichman. Tight hardness results for training depth-2 ReLU networks. In *12th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 185 of *LIPIcs*, pp. 22:1–22:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.ITCS.2021.22. Full version: arXiv:2011.13550.

Lee-Ad Gottlieb, Eran Kaufman, Aryeh Kontorovich, and Gabriel Nivasch. Learning convex polyhedra with margin. *IEEE Transactions on Information Theory*, 68(3):1976–1984, 2022. doi: 10.1109/TIT.2021.3134898. Preliminary version in Advances in Neural Information Processing Systems 31 (NeurIPS), 2018.

Anxin Guo and Aravindan Vijayaraghavan. Agnostic learning of arbitrary ReLU activation under Gaussian marginals. In *Proceedings of the 38th Conference on Learning Theory (COLT)*, volume 291 of *Proceedings of Machine Learning Research*, pp. 2592–2631, 2025.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi: 10.1006/jcss.2000.1727.

Tao Jiang, Minbo Gao, and Shaowei Cai. Binary quantized neural network training is W[1]-hard parameterized by input and output dimensions, 2026.

Sammy Khalife, Hongyu Cheng, and Amitabh Basu. Neural networks with linear threshold activations: Structure and algorithms. *Mathematical Programming*, 206(1–2):333–356, 2024. doi: 10.1007/s10107-023-02016-5.

Bobak T. Kiani, Jason Wang, and Melanie Weber. Hardness of learning neural networks under the manifold hypothesis. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. doi: 10.52202/079017-0184.

Sungyoon Kim and Mert Pilanci. Convex relaxations of ReLU neural networks approximate global optima in polynomial time. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24458–24485, 2024.

M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. The polynomial solvability of convex quadratic programming. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980. doi: 10.1016/0041-5553(80)90098-1. Translated from Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 20(5):1319–1323, 1980.

Pasin Manurangsi and Daniel Reichman. The computational complexity of training ReLU(s). *CoRR*, abs/1810.04207, 2018. Subsumed by the ITCS 2021 paper of Goel, Klivans, Manurangsi, and Reichman.

Michael S. Matena and Colin Raffel. A combinatorial perspective on the optimization of shallow ReLU networks. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022. arXiv:2210.00176.

Shay Moran, Ido Nachum, Itai Panasoff, and Amir Yehudayoff. On the perceptron's compression. In *Beyond the Horizon of Computability: 16th Conference on Computability in Europe (CiE)*, volume 12098 of *Lecture Notes in Computer Science*, pp. 310–325. Springer, 2020. doi: 10.1007/978-3-030-51466-2_29. Preprint: arXiv:1806.05403.

Albert B. J. Novikoff. On convergence proofs for perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume 12 of *Microwave Research Institute Symposia Series*, pp. 615–622. Polytechnic Press of the Polytechnic Institute of Brooklyn, 1963. Symposium held in April 1962.

Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003. doi: 10.1016/S0022-0000(03)00078-3.

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

Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, 1986.

Yifei Wang and Mert Pilanci. Polynomial-time solutions for ReLU network training: A complexity classification via max-cut and zonotopes, 2023.

Yuhan Ye. Parameterized complexity of stationarity testing for piecewise-affine functions and shallow CNN losses, 2026.

A Results Imported from the Literature

This appendix collects the statements that the proofs of the main text import from the literature. None of them is proved here, and each is cited to its source.

A.1 Parameterized Complexity

A *parameterized problem* associates with every instance a parameter $\lambda \in \mathbb{N}$. It is *fixed-parameter tractable*, and lies in the class FPT, when it is solvable in time $f(\lambda)\,\mathrm{poly}(n)$ for some computable f, and it lies in XP when it is solvable in time $n^{f(\lambda)}$. The class $\mathsf{W}[1]$ is the parameterized analogue of NP, and a problem that is $\mathsf{W}[1]$-hard is not in FPT unless $\mathsf{FPT} = \mathsf{W}[1]$. The Exponential Time Hypothesis of Impagliazzo & Paturi (2001), abbreviated ETH below, asserts that 3-SAT on n variables admits no algorithm running in time $2^{o(n)}$, and $\mathsf{FPT} = \mathsf{W}[1]$ would refute ETH. A problem is co-$\mathsf{W}[1]$-hard when its complement is $\mathsf{W}[1]$-hard. We refer to the monographs of Downey & Fellows (2013) and of Cygan et al. (2015) for the framework, and to the survey of Ganian (2026) for its use in machine learning.

A *parameterized reduction* from a problem with parameter λ to a problem with parameter λ' is a map computable in time $f(\lambda)\,\mathrm{poly}(n)$ that preserves the answer and satisfies $\lambda' \leq f(\lambda)$, for a computable f. Such a reduction transfers $\mathsf{W}[1]$-hardness, and when it is computable in polynomial time and satisfies $\lambda' = O(\lambda)$, it also transfers ETH-based lower bounds of the form $n^{o(\lambda)}$.

Parameters are integers, and in a running-time bound written $\varphi(1/\gamma) \cdot n^{o(1/\gamma^2)}$ the parameter is $1/\gamma^2$. The bound means $\varphi'(1/\gamma^2) \cdot n^{g(1/\gamma^2)}$ for a computable φ' on the integers and a function g with $g(\lambda) = o(\lambda)$, the first factor being written in $1/\gamma$ for readability alone.

Our source of hardness is the following problem, which Fellows et al. (2009) established as a convenient starting point for parameterized reductions, and which Pietrzak (2003) had studied earlier under the name partitioned clique. Call a clique of a vertex-colored graph *colorful* if its vertices carry pairwise distinct colors.

Problem A.1. In *multicolored clique* the input is an undirected graph $G = (V, E)$ whose vertices are colored with k colors, together with the integer k. The question is whether G contains a colorful clique of size k, that is, a clique with exactly one vertex of each color. The parameter is k.

The proof of Theorem 5.1 transfers the following hardness of multicolored clique.

Theorem A.2 (Cygan et al., 2015, Theorem 13.25 and Corollary 14.23, after Fellows et al., 2009; Chen et al., 2006)**.** *Multicolored clique is $\mathsf{W}[1]$-hard with respect to k, and for no computable function φ is it solvable in time $\varphi(k) \cdot |V|^{o(k)}$ unless ETH fails.*

The $\mathsf{W}[1]$-hardness is that of clique, transported along the parameterized reduction of Cygan et al. (2015, Theorem 13.7), and the lower bound under ETH is the bound of Chen et al. (2006) for clique, transported along the same reduction.

The lower bound we localize in Section 4 is the following.

Theorem A.3 (Froese et al., 2022, Theorem 1)**.** *For every $p \in [0, \infty)$, single-unit empirical risk minimization under the ℓ^p loss is NP-hard and $\mathsf{W}[1]$-hard with respect to the dimension d. Unless ETH fails, it is not solvable in time $\varphi(d) \cdot n^{o(d)}$ for any computable function φ.*

Two points of contact with Theorem 2.2 are worth recording, since Section 4 reads Theorem A.3 against it. The problem that Froese et al. (2022) state carries an output coefficient $a_1 \in \{-1, 1\}$ and minimizes over it as well, whereas Theorem 2.2 fixes the output to $\sigma\left(w^\top x + b\right)$. The hardness holds for Theorem 2.2 nonetheless, because both directions of their reduction are carried out for the neuron $x \mapsto \sigma\left(w^\top x + b\right)$ itself. Their inputs lie on the unit circle of a two-dimensional block, together with midpoints of pairs of such points, so every input has norm at most 1 and no rescaling is needed. We invoke Theorem A.3 for $p \in \{1, 2\}$ in the localization theorem. Its reduction uses rational inputs whose bit length is polynomial in the size of the source graph. The resulting family therefore retains its ETH lower bound when the running time includes a polynomial factor in the input bit length.

In a fixed dimension the proof of Theorem 4.1 falls back on the following algorithm.

Theorem A.4 (Arora et al., 2018, Theorem 4.1)**.** *Let the loss be convex, consider n rationally encoded points of $\mathbb{R}^d$ with $d \geq 2$, and suppose every fixed-pattern convex program is exactly solvable in time $\mathrm{poly}(L)$, where L is the input bit length. Empirical risk minimization over the networks $x \mapsto a\sigma\left(w^\top x + b\right)$, with an output weight $a \in \mathbb{R}$, a weight vector $w \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$, is then solvable to global optimality in time $O\left(n^d\right)\mathrm{poly}(L)$.*

Their statement is for networks of width κ, with running time $O\left(2^\kappa n^{d\kappa}\right)\mathrm{poly}(n, d, \kappa)$ in our notation, and we have set $\kappa = 1$. It allows an output weight a, whereas Theorem 2.2 fixes $a = 1$, so we use it through its proof, which for $\kappa = 1$ reads as follows.

Write $a\sigma(t) = s\sigma(|a|t)$, where $s \in \{+1, -1\}$ is the sign of a. The algorithm ranges over both values of s and over the at most n^d partitions of the sample that a hyperplane induces. Fix a sign s and a partition, and let A be the part on the positive side of the hyperplane. The algorithm then solves the convex program that minimizes $\sum_{i\in A} \ell\left(s\left(w^\top x_i + b\right), y_i\right) + \sum_{i\notin A} \ell(0, y_i)$. Its constraints are $w^\top x_i + b \geq 0$ for $i \in A$ and $w^\top x_i + b \leq 0$ for $i \notin A$. For $s = +1$ this program is Φ_A of Eq. (1), and the parts A the algorithm ranges over are exactly the activation patterns $A(w, b)$ of Theorem 2.4. Restricted to $s = +1$, the algorithm therefore returns, in time $O\left(n^d\right)\mathrm{poly}(L)$, the least of the minima of Φ_A over the activation patterns A of all neurons.

For a loss in the sense of Theorem 2.1 that least value is the minimum of $\widehat{L}$. Indeed, every neuron (w, b) is feasible for $\Phi_{A(w,b)}$, with objective value $\widehat{L}(w, b)$, by Theorem 3.3(ii) and (iii). Conversely, every minimum of Φ_A is attained at a neuron whose loss it equals, by Theorem 3.3(i) and (ii). The count of partitions needs $d \geq 2$. An instance in dimension 1 is placed in dimension 2 by appending a zero coordinate to every input. This changes neither the norms of the inputs nor the loss of any neuron, so the same algorithm solves such an instance in time $O\left(n^2\right)\mathrm{poly}(L)$.

A.2 Counting Large-Margin Dichotomies

Everything the algorithm of Section 3 needs about margins is contained in one classical fact and its counting consequence. The fact is the mistake bound of the perceptron rule, which converts a margin into a separator supported on few points.

Lemma A.5 (Novikoff, 1963; see also Ben-David et al., 2002, Theorem 15 for the mistake count and Moran et al., 2020, Section 2 for the procedure)**.** *Let $z_1, \ldots, z_n \in \mathbb{R}^D$ be points of norm $\|z_i\| \leq \rho$, and let $\chi \in \{+1, -1\}^n$ be a labeling of them. Suppose that some unit vector $u^\star$ satisfies $\chi_i\left(u^{\star\top} z_i\right) \geq \gamma > 0$ for every $i \in [n]$. Start from $u \leftarrow 0$ and repeat: if some index i has $\chi_i\left(u^\top z_i\right) \leq 0$, replace u by $u + \chi_i z_i$. Then the procedure performs at most $(\rho/\gamma)^2$ updates and halts with a vector u satisfying $\chi_i\left(u^\top z_i\right) > 0$ for every $i \in [n]$.*

Because the vector produced by Theorem A.5 is the sum of the signed points on which the rule updated, every labeling realizable with margin γ is named by a signed sequence of at most $(\rho/\gamma)^2$ of the points. Counting names bounds the number of labelings. The compression ingredient is the mistake bound above, which is Novikoff's, and the count in the following form is proved by Moran et al. (2020), who attribute it to Ben-David et al. (2002).

Theorem A.6 (Moran et al., 2020, after Ben-David et al., 2002)**.** *Let $z_1, \ldots, z_n \in \mathbb{R}^D$ satisfy $\|z_i\| \leq \rho$. The number of labelings $\chi \in \{+1, -1\}^n$ for which some $u \neq 0$ satisfies $\chi_i\left(u^\top z_i\right) \geq \gamma\|u\|$ for every $i \in [n]$ is at most $(2(n+1))^{\rho^2/\gamma^2}$.*

B Comparison with Neighboring Constructions

The remarks collected here compare the dimension and margin bounds and place them beside neighboring constructions.

Remark B.1. Theorem 4.1 gives a one-sided threshold for squared and absolute loss: a uniform computable margin lower bound asymptotically above $1/\sqrt{d}$ is incompatible with dimension hardness. It does not assert that the hard instances achieve margin exactly $\Theta(1/\sqrt{d})$. The statement is the analogue, for fitting a rectified linear unit, of a phenomenon observed by Ben-David & Simon (2000) for agnostic halfspace classification. There the best separating halfspace problem is already

hard to approximate on instances whose optimum attains margin $\Theta(1/\sqrt{d})$, namely $1/\sqrt{36d}$ in the affine case and $1/\sqrt{45(d+1)}$ in the homogeneous one, where d is the dimension (Ben-David & Simon, 2000, Theorem 4.4). Their margin is normalized by the norm of the input as well as by the norm of the weight vector, so the two thresholds are directly comparable on samples in the unit ball, which is the setting here. They prove the other direction as well, exhibiting for every constant μ a polynomial-time μ-successful algorithm (Ben-David & Simon, 2000, Theorem 3.9), so the shape of the dichotomy is theirs.

The comparison turns on the exponent of the margin in the algorithms. The algorithm of Ben-David and Simon ranges over the subsamples of size at most $1 + \lceil 4/\mu^4 \rceil$ (Ben-David & Simon, 2000, Section 3), so its exponent is a function of $1/\mu^4$ rather than of $1/\mu^2$. At a margin just above their hardness threshold it therefore costs $n^{O(d^2)}$, which does not improve on enumerating the $n^{O(d)}$ activation patterns. The exponent $O(1/\gamma^2)$ of Theorem 3.5 yields an $n^{o(d)}$ algorithm for every computable margin rate asymptotically above $1/\sqrt{d}$, which is the implication used by Theorem 4.1.

The second remark turns from the threshold of Theorem 4.1 to the construction behind Theorem 5.1.
Remark B.2. A known device places the inputs of a clique reduction at the standard basis vectors e_v, one per vertex, and at points combining e_v with $e_{v'}$. The bias is carried by a coordinate of its own, and the witness has margin $\Theta(1/\sqrt{k})$. Diakonikolas et al. (2019) use it to reduce k-clique to proper agnostic learning of γ-margin halfspaces, with a constant coordinate $*$ in place of our bias and their non-edge samples $\frac{1}{2}\left(1.1k^{-1/2}e_* - e_i - e_j\right)$ in place of our midpoints.

The gadget we re-embed is the bias-carrying counterpart of the densest-subgraph gadget of Goel et al. (2021). Their neurons have no bias, so they work in $\mathbb{R}^{N+1}$, with one coordinate per vertex and one extra coordinate that plays the role of our bias. Write $\delta' = 1/(2\sqrt{q})$ for a selected set of size q. Their vertex points are $\frac{1}{2}(e_v - \delta' e_*)$. Their pair points are $\frac{1}{2}(e_v + e_{v'} - 3.5\delta' e_*)$, and they are indexed by the *edges* of the graph. The coordinate $*$ is pinned by copies of $\left(e_*, 1/\sqrt{2}\right)$. As here, that gadget carries a margin $\Theta(1/\sqrt{q})$. The difference lies in the labels and hence in what the construction can prove. Their labels reward edges, so the completeness and soundness cases of their source problem separate only additively, by $\Omega\left(1/\sqrt{N}\right)$ in the average loss, which is what their running-time lower bound for additive approximation needs. Ours penalize bad pairs, so the two cases separate exactly, by one unit of loss, which is what turns a similar geometry into a W[1]-hardness proof.

The final comparison explains why the dimension and margin lower bounds coexist.
Remark B.3. Theorems A.3 and 5.1 inhabit different regimes, and neither subsumes the other. The reduction behind Theorem A.3 places its instances in dimension $d = 2k$, two coordinates per color, and is hard there; the reduction behind Theorem 5.1 is hard at margin $\Theta(1/\sqrt{k})$ and in dimension $d = N$, the number of vertices of the source graph. In the regime $k = o(N)$ and for squared or absolute loss, Theorem 3.5 solves the instances of our reduction in time $n^{O(k)}\,\mathrm{poly}(L) = n^{o(d)}\,\mathrm{poly}(L)$, so there is no conflict with Theorem 4.1. For these losses, no hard family in d has a uniform computable margin bound asymptotically above $1/\sqrt{d}$, while hardness in $1/\gamma^2$ persists at margins $\Theta(1/\sqrt{k})$ far above that order.

C Omitted Proofs

This appendix carries every proof that the main text defers, in the order the statements appear, and, just before the lemma that analyzes it, the construction behind Theorem 5.1. It is self-contained given Section 2, which defines the problem and the margin, and Appendix A, which states the perceptron mistake bound of Theorem A.5, the counting bound of Theorem A.6, and the two hardness statements Theorems A.2 and A.3.

Proof of Theorem 3.1. Call a finite sequence $\nu = ((i_1, s_1), \ldots, (i_T, s_T))$ with $0 \le T \le M$, $i_t \in [n]$ and $s_t \in \{+1, -1\}$ a *name*, and associate with it the vector

$$u_\nu = \sum_{t=1}^{T} s_t \tilde{x}_{i_t} \in \mathbb{R}^{d+1}.$$

The algorithm ranges over all names ν, computes the vector u_ν, and appends to $\mathcal{P}$ the labeling it induces, namely $\left(\operatorname{sign}\left(u_\nu^\top \tilde{x}_i\right)\right)_{i\in[n]}$.

Size and running time. For each T there are exactly $(2n)^T$ names of length T, so the number of names is $\sum_{T=0}^{M}(2n)^T \le (2n)^{M+1}$, the inequality holding because $2n \ge 2$. Enumerate the names by depth-first search of the tree of prefixes, in which a name of length T is its parent extended by one signed point. The vector u_ν is then obtained from the parent's vector in $O(d)$ additions and restored on backtracking. Each node costs $O(d)$ arithmetic operations to form u_ν and $O(nd)$ more to evaluate the inner products. On rational data, intermediate bit lengths grow by at most $O(\log(M+1))$ beyond the input bit length. Because the encoding of M is included in L, the total running time is $(2n)^{M+1}\operatorname{poly}(L)$ and the list has the stated size.

Completeness. Let $\chi \in \{+1,-1\}^n$ be realizable with margin γ, witnessed by $\tilde{w} \ne 0$ with $\chi_i\left(\tilde{w}^\top \tilde{x}_i\right) \ge \gamma\|\tilde{w}\|$ for every $i \in [n]$. The vector $u^\star = \tilde{w}/\|\tilde{w}\|$ is then a unit vector with $\chi_i\left(u^{\star\top}\tilde{x}_i\right) \ge \gamma$ for every i. Apply Theorem A.5 to the lifted inputs $\tilde{x}_1,\dots,\tilde{x}_n$, which satisfy $\|\tilde{x}_i\| \le \sqrt{2} =: \rho$, with the labeling χ and the witness $u^\star$. The rule performs $T \le (\rho/\gamma)^2 = 2/\gamma^2 \le M$ updates and halts with a vector u satisfying $\chi_i\left(u^\top\tilde{x}_i\right) > 0$ for every $i \in [n]$. Write $i_1,\dots,i_T$ for the indices on which the rule updated, in order, and put $s_t = \chi_{i_t}$. The update rule then gives exactly $u = \sum_{t=1}^{T} s_t\tilde{x}_{i_t} = u_\nu$ for the name $\nu = ((i_1,s_1),\dots,(i_T,s_T))$, and this name has length $T \le M$, so it is among the names the algorithm ranges over. For that name, the inequality $\chi_i\left(u^\top\tilde{x}_i\right) > 0$ forces $\operatorname{sign}\left(u^\top\tilde{x}_i\right) = \chi_i$ for every i, so the labeling appended to $\mathcal{P}$ is χ. □

Proof of Theorem 3.3. (i) Each summand with $i \in A$ is the convex function $\ell(\cdot, y_i)$ composed with the affine map $(w,b) \mapsto w^\top x_i + b$, hence convex; each summand with $i \notin A$ is the constant $\ell(0,y_i)$. A finite sum of convex functions is convex. The constraints are n affine inequalities in (w,b), one per index, so the feasible region is a polyhedron, and it is nonempty because $(w,b) = (0,0)$ satisfies every constraint with equality.

For attainment, the case $A = \emptyset$ is immediate, the objective being constant, so let $A \ne \emptyset$. The objective depends on (w,b) only through the vector of preactivations $t = \left(w^\top x_i + b\right)_{i\in[n]}$. The set of achievable such vectors is $\left\{\tilde{X}\tilde{w} : \tilde{w} \in \mathbb{R}^{d+1}\right\}$, where $\tilde{X}$ is the matrix with rows $\tilde{x}_i^\top$; it is a linear subspace of $\mathbb{R}^n$ and hence a polyhedron. Intersect it with the n halfspaces that the constraints of Φ_A describe. They are $t_i \ge 0$ for $i \in A$ and $t_i \le 0$ for $i \notin A$. The intersection is a polyhedron, which we call Π. A vector t lies in Π exactly when it is the preactivation vector of a feasible (w,b). Write π_A for the projection onto the coordinates indexed by A and $\Gamma(t_A) = \sum_{i\in A}\ell(t_i,y_i)$, so that the objective equals $\Gamma(\pi_A(t))$ plus the constant $\sum_{i\notin A}\ell(0,y_i)$. The infimum of the objective over Π is therefore that of Γ over $Q = \pi_A(\Pi)$. The set Q is again a polyhedron, by Fourier–Motzkin elimination (Schrijver, 1986, Section 12.2), and in particular closed, and it contains the origin because $(w,b) = (0,0)$ is feasible.

The function Γ is continuous, each $\ell(\cdot,y_i)$ being convex and finite on all of $\mathbb{R}$, and it is coercive, as follows. Fix $\beta > 0$. Each $\ell(\cdot,y_i)$ is coercive and A is finite. There is therefore one radius $r > 0$ such that $\ell(\theta,y_i) \ge \beta$ for every $i \in A$ and every real θ with $|\theta| \ge r$. Suppose $\|t_A\| \ge r\sqrt{|A|}$, where $|A|$ is the number of indices in A. Then some coordinate satisfies $|t_i| \ge r$, so $\Gamma(t_A) \ge \ell(t_i,y_i) \ge \beta$, the remaining summands being nonnegative by Theorem 2.1. Taking $\beta = \Gamma(0)+1$, every $q \in Q$ with $\Gamma(q) \le \Gamma(0)$ therefore has $\|q\| < r\sqrt{|A|}$. Hence the sublevel set $\{q \in Q \mid \Gamma(q) \le \Gamma(0)\}$ is closed, nonempty and bounded, therefore compact, and Γ attains on it a minimum, which is its minimum over Q. The minimizer lies in $Q = \pi_A(\Pi)$, so it equals $\pi_A(t)$ for some $t \in \Pi$. Since $t \in \Pi$, some (w,b) has preactivation vector t, and every such (w,b) minimizes the objective.

(ii) Let (w,b) be feasible. Take $i \in A$. Feasibility gives $w^\top x_i + b \ge 0$, so $h_{w,b}(x_i) = \sigma\left(w^\top x_i + b\right) = w^\top x_i + b$. The i-th term of $\widehat{L}(w,b)$ is therefore $\ell\left(w^\top x_i + b, y_i\right)$, which is the i-th term of the objective. Take $i \notin A$. Feasibility gives $w^\top x_i + b \le 0$, so $h_{w,b}(x_i) = 0$. The i-th term of $\widehat{L}(w,b)$ is therefore $\ell(0,y_i)$, again the i-th term of the objective. Summing over $i \in [n]$ gives the claim.

(iii) If $A(w,b) = A$ then $w^\top x_i + b > 0$ for $i \in A$ and $w^\top x_i + b \le 0$ for $i \notin A$, which implies both families of constraints. □

Proof of Theorem 3.5. Write $\mathcal{P}$ for the list of labelings that the loop of Algorithm 1 produces, which is exactly the list of Theorem 3.1 for the given M, and identify a labeling χ with the active set $A = \{i \mid \chi_i = +1\}$.

Every recorded value is attained by a genuine neuron. Fix an iteration with active set A, and let (w,b) be the minimizer of Φ_A that the iteration produces, which exists by Theorem 3.3(i). It is feasible for Φ_A, so by Theorem 3.3(ii) the recorded value v equals $\widehat{L}(w,b)$. Hence every value recorded during the run is the empirical loss of an explicit neuron, and therefore is at least the global minimum of $\widehat{L}$. The loop is nonempty, since the empty name has length $0 \le M$, and every value it records is finite, so the first iteration replaces the initial *best* $= +\infty$. Thereafter *best* and $(w^\star, b^\star)$ are updated together, so the returned pair satisfies $\widehat{L}(w^\star, b^\star) =$ *best*.

The optimal pattern is enumerated. Let (w^o, b^o) be an optimal solution that separates the sample with margin at least γ, which exists because the instance is γ-separated. Let $A^o = A(w^o, b^o)$ be its activation pattern, and let χ^o be the associated labeling, with $\chi^o_i = +1$ for $i \in A^o$ and $\chi^o_i = -1$ otherwise. By Theorem 2.5, $\tilde{w}^o \neq 0$ and $\left|\tilde{w}^{o\top} \tilde{x}_i\right| \ge \gamma \left\|\tilde{w}^o\right\|$ for every i. By Theorem 2.4, the sign of $\tilde{w}^{o\top} \tilde{x}_i$ is χ^o_i. Together these give $\chi^o_i \left(\tilde{w}^{o\top} \tilde{x}_i\right) \ge \gamma \left\|\tilde{w}^o\right\|$ for every i. That is, χ^o is realizable with margin γ. By Theorem 3.1 we get $\chi^o \in \mathcal{P}$, so some iteration of the loop has $A = A^o$.

Optimality. Consider that iteration. By Theorem 3.3(iii) the neuron (w^o, b^o) is feasible for Φ_{A^o}, and by Theorem 3.3(ii) the objective of Φ_{A^o} at (w^o, b^o) equals $\widehat{L}(w^o, b^o)$, the global minimum. Hence the value v of Φ_{A^o} is at most the global minimum, so after this iteration *best* is at most the global minimum, and it never increases afterwards. Combined with the first part, this shows that *best* equals the global minimum at termination and is attained at the returned neuron. The last two parts used only that the activation pattern of (w^o, b^o) is realizable with margin γ, never that (w^o, b^o) is optimal. Applied to an arbitrary neuron (w,b) whose activation pattern is realizable with margin γ, for every $\gamma \in (0,1]$ with $\lceil 2/\gamma^2 \rceil \le M$, they show that *best* $\le \widehat{L}(w,b)$ at termination. This is the guarantee stated in Algorithm 1.

Running time. By Theorem 3.1 the loop performs at most $(2n)^{M+1}$ iterations. The loop is run as the depth-first enumeration of names in the proof of Theorem 3.1. The vector u of a name is therefore obtained from that of its parent in $O(d)$ additions rather than recomputed from its $T \le M$ terms, and forming A costs $O(nd)$ further operations. The enumeration overhead is therefore $(2n)^{M+1} \operatorname{poly}(L)$, with a polynomial that does not depend on M. Each iteration solves one convex program, at cost $T_{\mathrm{cvx}}(L)$ under the hypothesis of Theorem 3.5; Theorem 3.3(i) guarantees that a minimizer exists. The total is $(2n)^{M+1}\left(T_{\mathrm{cvx}}(L) + \operatorname{poly}(L)\right)$. Take the smallest admissible choice $M = \lceil 2/\gamma^2 \rceil$. Since $\gamma \le 1$, we have $M + 1 \le 2/\gamma^2 + 2 \le 4/\gamma^2$. Since $2n \le n^2$ for $n \ge 2$, the running time is at most $n^{8/\gamma^2}\left(T_{\mathrm{cvx}}(L) + \operatorname{poly}(L)\right)$. For squared and ℓ^1 loss on rational data this is $n^{O(1/\gamma^2)} \operatorname{poly}(L)$, with an exponent independent of d. □

One special case of Theorem 2.2 is easy with no margin promise at all.

Remark C.1. On a sample that some neuron fits with loss zero, the margin promise is not needed. Take an ℓ^p loss with $p \ge 1$. A neuron (w,b) has loss 0 exactly when $\sigma\left(w^\top x_i + b\right) = y_i$ for every $i \in [n]$. That happens exactly when every label is nonnegative and the neuron meets two conditions. Its preactivation equals the label, $w^\top x_i + b = y_i$, at every input with $y_i > 0$, and it is nonpositive, $w^\top x_i + b \le 0$, at every input with $y_i = 0$. Such a neuron has activation pattern $A_+ = \{i \in [n] \mid y_i > 0\}$, so it is feasible for Φ_{A_+} with objective value 0 by Theorem 3.3(ii) and (iii), and the minimum of Φ_{A_+}, which is nonnegative, is 0. Conversely, the minimum of Φ_{A_+} is attained by Theorem 3.3(i), and a minimizer with value 0 is a neuron of loss 0 by Theorem 3.3(ii). For rational data, deciding whether the optimum is 0 and finding a witness requires only the displayed affine equalities and inequalities, a linear feasibility problem solvable in time $\operatorname{poly}(L)$, on every sample and with no margin. This is the case usually called realizable, on which the algorithm of Dey et al. (2020) is likewise exact, and the promise of Theorem 3.5 matters only when the optimum is positive.

Proof of Theorem 4.1. Suppose, for contradiction, that such a ψ exists; we construct a single algorithm that solves every instance of the family in time $\varphi(d) \cdot n^{o(d)}\,\mathrm{poly}(L)$ with φ computable, and the defining property of the family forbids any such algorithm. Let d_0 be such that the displayed inequality holds and $\psi(d) > 0$ for every $d \geq d_0$. Such a d_0 exists because the inequality holds for every sufficiently large d and ψ tends to infinity. Consider the following algorithm. On an instance of dimension $d < d_0$ it runs the algorithm of Theorem A.4, at cost $n^{O(d)}\,\mathrm{poly}(L) \leq n^{O(d_0)}\,\mathrm{poly}(L)$. Since d_0 is fixed, this cost is $\mathrm{poly}(L)$. On an instance of dimension $d \geq d_0$ it runs Algorithm 1 with the parameter

$$M(d) = \left\lceil \frac{2d}{\psi(d)} \right\rceil + 1,$$

which is computable from d because ψ is. Let $h_\psi(d)$ bound the time to compute this integer.

Correctness on the hard family. Fix $d \geq d_0$ and $I \in \mathcal{I}_d$. The inequality assumed for ψ gives $\gamma^\star(I)^2 \geq \psi(d)/d > 0$, so $\gamma^\star(I) > 0$. As γ increases to $\gamma^\star(I)$, the quantity $2/\gamma^2$ therefore tends to $2/\gamma^\star(I)^2$, which is at most $2d/\psi(d)$. Choose $\gamma' \in (0, \gamma^\star(I))$ close enough to $\gamma^\star(I)$ that $2/\gamma'^2 < 2d/\psi(d) + 1$. Such a γ' exists because $\gamma \mapsto 2/\gamma^2$ is continuous at $\gamma^\star(I) > 0$. Moreover $2d/\psi(d) + 1 \leq M(d)$, because $M(d) = \lceil 2d/\psi(d) \rceil + 1$, so $2/\gamma'^2 < M(d)$. Put $\gamma = \min\{1, \gamma'\} \in (0, 1]$. Then $2/\gamma^2 = \max\{2, 2/\gamma'^2\} \leq M(d)$, because $2/\gamma'^2 < M(d)$ and $M(d) \geq 2$. The bound $M(d) \geq 2$ holds because $d \geq 1$ and $\psi(d) > 0$ give $2d/\psi(d) > 0$, hence $\lceil 2d/\psi(d) \rceil \geq 1$. By the definition of $\gamma^\star(I)$ as a supremum, some optimal solution of I separates its sample with a margin $\gamma'' > \gamma'$. It therefore separates the sample with margin at least $\gamma' \geq \gamma$, so I is γ-separated. The loss ℓ^p with $p \in \{1, 2\}$ has an exact polynomial-time fixed-pattern solver by Theorem 3.4, the margin satisfies $\gamma \in (0, 1]$, and $M(d)$ is an integer with $M(d) \geq 2/\gamma^2$. Together with the γ-separation of I just established, these are the hypotheses of Theorem 3.5, so Algorithm 1 run with $M(d)$ returns a global minimizer of I. Note that the algorithm needs no knowledge of γ: only the bound $M(d)$, which depends on d alone, enters the computation.

Running time. For all sufficiently large d, the condition $\psi(d) \to \infty$ gives $M(d) = O(d)$. Encoding $M(d)$ thus adds only $O(\log d)$ bits to the instance, which is absorbed by L. Fix a polynomial $P(L)$ that bounds the common per-pattern cost for $p \in \{1, 2\}$, including the polynomial overhead of Theorem 3.5, and a constant c'' such that Theorem A.4 runs in time $n^{c''d}P(L)$. In dimension $d \geq d_0$ the cost is at most $(2n)^{M(d)+1}P(L)$. Here $M(d) + 1 \leq 2d/\psi(d) + 3$, and $2^{M(d)+1} \leq n^{M(d)+1}$ because $n \geq 2$, so the cost is at most $n^{4d/\psi(d)+6}P(L)$. In each of the finitely many dimensions $d < d_0$ the cost is at most $n^{c''d_0}P(L)$. Put $g(d) = 4d/\psi(d) + 6$ for $d \geq d_0$ and $g(d) = c''d_0$ for $d < d_0$. Because $M(d)$ is computable from d, this is a single algorithm for the family. It solves every $I \in \bigcup_d \mathcal{I}_d$ in time at most $\varphi(d)n^{g(d)}P(L)$ for a computable φ that absorbs $h_\psi(d)$. Moreover $g(d)/d = 4/\psi(d) + 6/d \to 0$ because $\psi(d) \to \infty$, so $g(d) = o(d)$. This contradicts the defining property of the family, that no algorithm solves every $I \in \bigcup_d \mathcal{I}_d$ in time $\varphi(d) \cdot n^{o(d)}\,\mathrm{poly}(L)$, which Theorem A.3 supplies under ETH for its rationally encoded reduction family.

For the converse statement, let $\lambda\colon \mathbb{N} \to \mathbb{N}$ be computable with $\lambda(d) = o(d)$. Let I be an instance of dimension d that is γ-separated for some $\gamma \in (0, 1]$ with $1/\gamma^2 \leq \lambda(d)$. Run Algorithm 1 with $M = 2\lambda(d)$, an integer computable from d with $M \geq 2/\gamma^2$; let $h_\lambda(d)$ bound its computation time. For sufficiently large d, the condition $\lambda(d) = o(d)$ gives $M = O(d)$, so its encoding also adds only $O(\log d)$ bits. Since ℓ^p with $p \in \{1, 2\}$ has an exact polynomial-time fixed-pattern solver and $\gamma \in (0, 1]$, Theorem 3.5 applies, and the run returns a global minimizer in time $(2n)^{M+1}\,\mathrm{poly}(L)$. Because $n \geq 2$, this time is at most $n^{2(M+1)}\,\mathrm{poly}(L)$, and $2(M+1) = 4\lambda(d)+2 = o(d)$. Absorbing $h_\lambda(d)$ into a computable factor $\varphi(d)$ gives the stated time $\varphi(d)n^{o(d)}\,\mathrm{poly}(L)$. □

The construction behind Theorem 5.1. Fix a constant $p \geq 1$ and the loss $\ell(\hat{y}, y) = |\hat{y} - y|^p$. Let (G, k) with $G = (V, E)$ be an instance of multicolored clique with $k \geq 1$ colors. Let $N = |V|$ be the number of vertices, and assume $N \geq 2k$. On smaller instances the reduction decides the question by brute force and outputs one of two fixed instances accordingly, exhibited in the proof of Theorem 5.1 below. The brute force tests each of the at most N^k candidate k-sets for adjacency and colorfulness, spending time $O(k^2)$ on each. Its cost is therefore $O\left(k^2 N^k\right)$, which is $O\left(k^2(2k)^k\right)$ since $N < 2k$, a function of k alone. Call a pair $\{v, v'\}$ of distinct vertices *bad* if v and v' are non-adjacent or have the same color, so that no colorful clique contains both. Write $k^\star$ for the largest size of a colorful clique of G, so that $k^\star \leq k$ always and $k^\star = k$ exactly on yes-instances of

Theorem A.1. Let $m = \lceil p \rceil$ be the least integer that is at least p, and set

$$\delta = \frac{1}{2p(N-k+1)}, \qquad R = 2^m (N-k) \left(2m(N-k+1)\right)^m + 1, \qquad \gamma = \frac{1}{\sqrt{16k+9}}.$$

The number δ enters only the analysis, while R is a positive integer that the reduction computes from N and k by integer arithmetic, whatever the value of p. The analysis uses R through one inequality,

$$R\delta^p > 2^p(N-k), \tag{2}$$

which holds as follows. Since $p \geq 1$ and $N-k+1 \geq k+1 \geq 2$, we have $2p(N-k+1) \geq 4$, so $\delta \leq 1/4 < 1$. Put $\delta_m = 1/\left(2m(N-k+1)\right)$, so that $\delta \geq \delta_m > 0$ because $m \geq p$. A number in $(0,1)$ raised to the power $m \geq p$ is at most its p-th power, so $\delta^p \geq \delta^m \geq \delta_m^m$. Hence $R\delta^p \geq R\delta_m^m > (R-1)\delta_m^m = 2^m(N-k) \geq 2^p(N-k)$, the equality because $\left(2m(N-k+1)\right)^m \delta_m^m = 1$ and the last inequality because $m \geq p$ and $N-k \geq 0$.

The reduction outputs the promise as the integer $1/\gamma^2 = 16k+9$, which is the parameter. The feature space is $\mathbb{R}^N$, with one coordinate per vertex, and e_v denotes the standard basis vector of the coordinate of the vertex v. The sample consists of

- the point e_v with label 1, for each vertex $v \in V$; and
- R copies of the midpoint $\frac{1}{2}\left(e_v + e_{v'}\right)$ with label 0, for each bad pair $\{v, v'\}$.

Every input has norm at most 1, since $\|e_v\| = 1$ and $\left\|\frac{1}{2}(e_v + e_{v'})\right\| = \frac{1}{2}\sqrt{2} < 1$. Because m is a constant, $R = O\left(N^{m+1}\right)$, and since there are fewer than N^2 bad pairs the sample has

$$n = N + R \cdot |\{\text{bad pairs}\}| = O\left(N^{m+3}\right)$$

points and is computable from (G,k) in time polynomial in N. For a neuron (w,b) we write $t_v = w^\top e_v + b = w_v + b$ for the preactivation at the vertex point e_v, so that the preactivation at the midpoint of $\{v,v'\}$ is

$$w^\top\left(\tfrac{1}{2}\left(e_v + e_{v'}\right)\right) + b = \tfrac{1}{2}\left(w_v + w_{v'}\right) + b = \tfrac{1}{2}\left(t_v + t_{v'}\right).$$

Lemma C.2. *On the instance above, every neuron has ℓ^p loss at least $N - k^\star$, and this value is attained by the neuron*

$$w_v = \begin{cases} 4, & v \in K^\star, \\ 0, & v \notin K^\star, \end{cases} \qquad b = -3,$$

where $K^\star$ is a colorful clique of G of maximum size $k^\star$. That neuron separates the sample with margin $1/\sqrt{16k^\star+9} \geq \gamma$, so the instance is γ-separated with optimum exactly $N - k^\star$.

Proof. Upper bound. For the stated neuron, $t_v = 4 - 3 = 1$ for $v \in K^\star$ and $t_v = -3$ for $v \notin K^\star$. The vertex point of each $v \in K^\star$ has output $\sigma(1) = 1$ and label 1, so it contributes 0. Each of the $N - k^\star$ remaining vertex points has output $\sigma(-3) = 0$ and label 1, so it contributes 1. A bad pair contains at most one vertex of $K^\star$, because the members of $K^\star$ are pairwise adjacent and pairwise differently colored and therefore form no bad pair. Hence the midpoint of a bad pair has preactivation $\frac{1}{2}(1-3) = -1$ or $\frac{1}{2}(-3-3) = -3$. Its output is 0 and its label is 0, so each of its R copies contributes 0. The total loss is $N - k^\star$.

Margin of that neuron. The lifted weight vector of that neuron has $\|\tilde{w}\|^2 = 16k^\star + 9 > 0$, and by the computation above every sample point has preactivation in $\{1, -1, -3\}$. The value 1 is attained, since $k^\star \geq 1$: a single vertex is a colorful clique, and $V \neq \emptyset$ because $N \geq 2k \geq 2$. Hence $\min_i \left|\tilde{w}^\top \tilde{x}_i\right| = 1$ and the margin is $1/\sqrt{16k^\star+9}$. Since $k^\star \leq k$ this is at least $1/\sqrt{16k+9} = \gamma$.

Lower bound. Let (w,b) be any neuron, let t_v be as above, and set $U = \{v \in V \mid t_v > \delta\}$.

Suppose first that U contains a bad pair $\{v,v'\}$. Then the corresponding midpoint has preactivation $\frac{1}{2}\left(t_v + t_{v'}\right) > \delta$ and hence output greater than δ, while its label is 0, so each of its R copies contributes more than δ^p. The total loss exceeds $R\delta^p > 2^p(N-k) \geq 2(N-k) \geq N \geq N - k^\star$, where the first inequality is Eq. (2), the second uses $p \geq 1$ and the third uses $N \geq 2k$.

Suppose now that U contains no bad pair. Then the vertices of U are pairwise adjacent and pairwise differently colored, so U is a colorful clique and $|U| \leq k^\star$. Partition $V \setminus U$ into the set B of vertices

that form a bad pair with at least one member of U, and the set $C = V \setminus (U \cup B)$ of vertices that form a bad pair with no member of U.

For every $v \in B$ fix a vertex $\hat{v} \in U$ such that $\{v, \hat{v}\}$ is a bad pair. We claim that the sample points associated with v, namely the vertex point e_v together with the R copies of the midpoint of $\{v, \hat{v}\}$, contribute at least 1 to the loss. Indeed $v \notin U$ gives $t_v \le \delta$. If $t_v \le 0$, then e_v has output 0 and label 1 and contributes exactly 1. If $0 < t_v \le \delta$, then the midpoint of $\{v, \hat{v}\}$ has preactivation $\frac{1}{2}(t_v + t_{\hat{v}}) > \frac{1}{2}(0 + \delta) = \delta/2$, since $\hat{v} \in U$. Each of its R copies then contributes more than $(\delta/2)^p$, and the copies together contribute more than $R\delta^p/2^p > N - k \ge k \ge 1$, the first inequality by Eq. (2) and the last two using $N \ge 2k$ and $k \ge 1$. The points invoked for distinct $v \in B$ are distinct, because the bad pair $\{v, \hat{v}\}$ determines v as its unique member outside U, and none of them is a vertex point of a member of C.

For every $v \in C$ we have $t_v \le \delta$. The vertex point e_v then has output at most δ and label 1, so it contributes at least $(1 - \delta)^p$. Here $\delta \le 1/4$, as shown with Eq. (2). The output $\sigma(t_v) \in [0, \delta]$ therefore lies below the label 1, and the contribution is $(1 - \sigma(t_v))^p \ge (1 - \delta)^p$. Adding the two families of contributions, which involve pairwise distinct sample points,

$$\widehat{L}(w, b) \ge |B| + |C|(1 - \delta)^p = (N - |U|) - |C|\,(1 - (1 - \delta)^p)\,,$$

using $|B| + |C| = N - |U|$. By Bernoulli's inequality, $(1 - \delta)^p \ge 1 - p\delta$ for $p \ge 1$ and $\delta \in [0, 1]$, both of which hold here, so $1 - (1 - \delta)^p \le p\delta = 1/(2(N - k + 1))$. With $|C| \le N$ the subtracted term is therefore at most $N/(2(N - k + 1))$, which is smaller than 1 because $N \ge 2k$ gives $2(N - k + 1) > N$.

If $C \ne \emptyset$, pick a vertex $v \in C$. Then $U \cup \{v\}$ contains no bad pair and is therefore a colorful clique, so $|U| + 1 \le k^\star$ and $N - |U| \ge N - k^\star + 1$. Hence $\widehat{L}(w, b) > (N - k^\star + 1) - 1 = N - k^\star$. If $C = \emptyset$, then $\widehat{L}(w, b) \ge |B| = N - |U| \ge N - k^\star$. In both cases the loss is at least $N - k^\star$, which completes the proof. □

Proof of Theorem 5.1. The map that sends a multicolored clique instance (G, k) with $N \ge 2k$ to the sample constructed above, together with the margin $\gamma = 1/\sqrt{16k + 9}$ and the threshold $\tau = N - k$, is computable in time polynomial in N, as noted in the construction. By Theorem C.2 the produced instance is γ-separated and its optimum equals $N - k^\star$. The optimum is at most $\tau = N - k$ if and only if $k^\star \ge k$, that is, since $k^\star \le k$ always, if and only if $k^\star = k$, which holds if and only if G has a colorful clique with one vertex of each color. The map therefore preserves the answer. It sends the source parameter k to the target parameter $1/\gamma^2 = 16k + 9$, a function of k alone, so it is a parameterized reduction. Multicolored clique is $\mathsf{W}[1]$-hard by Theorem A.2, and hence the target problem is $\mathsf{W}[1]$-hard with respect to $1/\gamma^2$.

For the running-time bound, suppose an algorithm solved the target problem in time $\varphi(1/\gamma) \cdot n^{o(1/\gamma^2)}$ for a computable φ. Composing it with the reduction decides multicolored clique in time

$$\varphi\left(\sqrt{16k + 9}\right) \cdot \left(O\left(N^{m+3}\right)\right)^{o(16k+9)} + \mathrm{poly}(N) = \varphi'(k) \cdot N^{o(k)},$$

where φ' is computable and where the exponent $m + 3$ is a constant absorbed into the $o(k)$ because p, and with it m, is fixed. The steps behind that display are worth spelling out. Write the assumed running time as $\varphi(1/\gamma) n^{g(1/\gamma^2)}$ with $g(\lambda) = o(\lambda)$, and write $n \le cN^{m+3}$ with $c = c(p)$. The function g may be taken nondecreasing, by replacing $g(\lambda)$ with $\max_{\lambda' \le \lambda} g(\lambda')$, which is still $o(\lambda)$ and only enlarges the bound. Consider first $N \ge c$. There the factor $c^{g(16k+9)}$ is at most $N^{g(16k+9)}$ and joins the exponent. The exponent becomes $(m+4)g(16k+9)$, which is $o(k)$ because $16k+9 = O(k)$ and $m + 4$ is a constant. The finitely many instances with $N < c$, which have $k \le N/2 < c/2$, cost at most a constant, which joins φ'. The additive $\mathrm{poly}(N) = O(N^{c_0})$ is absorbed by replacing g with $\lambda \mapsto \max\{c_0, (m + 4)g(16\lambda + 9)\}$, which is still nondecreasing and $o(\lambda)$. The factor $\varphi\left(\sqrt{16k + 9}\right)$ is a computable function of k under the reading of Appendix A.1, so φ' is computable.

The instances with $N < 2k$ are handled by the brute-force branch, which costs $O\left(k^2(2k)^k\right)$, a function of k alone, and so joins φ' as well.

The two fixed instances that branch emits are 1-separated, so they satisfy the promise. Both are samples of points of $\mathbb{R}^1$, whose one standard basis vector is e_1. For a yes-answer the branch emits

the sample $((e_1, 1), (e_1, 1))$ with threshold 0. The sample has two copies of one point, so that $n \geq 2$ as assumed in Section 2.1. Its optimum is 0, attained by $w = e_1$ and $b = 0$, and this neuron has $\left|\tilde{w}^\top \tilde{x}_i\right| = 1 = \|\tilde{w}\|$ at both points. For a no-answer the branch emits the sample $((e_1, 1), (e_1, 0))$ with threshold 0. Its optimum is $\min\left\{1, 2^{1-p}\right\} > 0$. The optimum is attained at every preactivation $t^\star \in [0, 1]$ when $p = 1$, and only at $t^\star = 1/2$ when $p > 1$. Choose $t^\star = 1$ when $p = 1$ and $t^\star = 1/2$ when $p > 1$. The neuron with $w = t^\star e_1$ and $b = 0$ is then optimal, and its lifted weight vector $\tilde{w} = (t^\star e_1, 0)$ is nonzero. It satisfies $\left|\tilde{w}^\top \tilde{x}_i\right| = t^\star = \|\tilde{w}\|$ at both points, again at margin 1. The choice $t^\star > 0$ matters, because $t^\star = 0$ is also optimal when $p = 1$ and yields the neuron $(0, 0)$, which Theorem 2.5 declares not to separate. Both fixed instances are output together with the parameter $16k + 9$ of the main branch, so the parameter map is unchanged, and each is γ-separated because it is 1-separated and $\gamma \leq 1$.

A running time of $\varphi'(k) \cdot N^{o(k)}$ for multicolored clique contradicts Theorem A.2 under ETH. The remaining claims of the statement follow from two observations. A running time $f(1/\gamma)\,\mathrm{poly}(n)$ is of the form $\varphi(1/\gamma) \cdot n^{o(1/\gamma^2)}$: take $\mathrm{poly}(n) = O\left(n^{c_1}\right)$ and $g(\lambda) = \max\left\{c_1, \left\lceil\sqrt{\lambda}\right\rceil\right\}$, which is computable, nondecreasing and $o(\lambda)$. Moreover, $\mathsf{W}[1]$-hardness precludes membership in FPT unless $\mathsf{FPT} = \mathsf{W}[1]$. □