THE COMPLEXITY OF BATCHED RELU NETWORK EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We ask whether a batch of inputs to a fixed rectified-linear network can be evaluated asymptotically faster than by a separate forward pass on each input. One input through L layers of width d costs $\Theta(Ld^2)$, and the textbook batch is B such passes; ordinary batching improves locality, not this count. Applying the rectifier costs only linear time in its input size; the main cost lies in the affine maps and in determining which linear region contains each input. Separating evaluation into a geometric part and an algebraic part, we show the algebra amortizes: given the region labels, a batch confined to few regions costs a factor $\Theta(\min(L, d))$ less than pointwise, the cap being the price of reading the labels. Batched evaluation therefore reduces to batched point location, and the converse fails, so location is strictly the harder half. We identify settings in which point location is cheap. Under crossing and margin density assumptions, a concentrated batch occupies few regions through a random network, and a single distance comparison certifies a point's region. For arbitrary weights, exact evaluation has matching matrix-multiplication reductions and upper bounds. For approximation the obstacle is the network's conditioning rather than any flipped sign.

1 INTRODUCTION

Rectified-linear (ReLU) units are the dominant nonlinearity in modern feedforward networks (Nair & Hinton, 2010; Goodfellow et al., 2016), and a network built from affine maps and the rectifier $\sigma(z) = \max(z, 0)$ computes a continuous piecewise-linear function of its input (Arora et al., 2018). Evaluating such a network on a single input of dimension d_0 through L layers of width d costs $\Theta(Ld^2)$ arithmetic operations, dominated by the matrix–vector products in the affine layers. The standard way to evaluate a batch of B inputs runs the same computation B times. Expressing each layer as a matrix–matrix product improves memory locality but still takes $\Theta(BLd^2)$ work. The locality matters in practice, but it does not change the operation count: batching, as usually understood, gives a constant factor, not an asymptotic one. For a depth-50 network of width 1000 and a batch of 10^4 inputs, this is on the order of 10^{12} multiply-adds whether the inputs are unrelated or nearly identical; the structure of the batch is invisible to the count.

We ask when batched evaluation can use asymptotically fewer operations. The starting point is the observation that the rectifier is the wrong place to look for savings. An elementwise rectifier over n scalars must read its n inputs and write its n outputs, so it costs $\Theta(n)$, which is optimal; it cannot be sped up and it is dominated by the affine layer that feeds it. Any saving comes instead from the piecewise-linear geometry: on each region of input space where the sign pattern of all hidden units is fixed, the network restricts to a single affine map, and inputs sharing a region share that map. Region counts are well studied (Montúfar et al., 2014; Telgarsky, 2016; Raghu et al., 2017; Serra et al., 2018; Hanin & Rolnick, 2019a), and exact line and planar-slice partitions can be computed (Sotoudeh & Thakur, 2019; Humayun et al., 2023). Here we ask how that geometry controls the cost of evaluating a batch in the ambient space.

We organize the cost of batched evaluation into two parts. The *geometric* part decides, for each input, which region it belongs to. The *algebraic* part applies the affine map of that region to the input. We show that the algebraic part is cheap and amortizable, so that the entire difficulty of beating pointwise evaluation is concentrated in the geometric part, which is a batched point-location problem. We then chart what the geometric part costs: through a random network a certificate replaces point location

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.

and genuine asymptotic speedups follow; for generic weights it is exactly matrix multiplication; and under approximation it disappears, the speedup being gated by the network's conditioning instead.

1.1 Our Results

Throughout, $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ is an L-layer ReLU network of size T (the total number of weights, equal to the cost of one forward pass up to constants), and a batch is a multiset of B inputs. Formal statements and proofs appear in Section 3 and Appendices A to C; the numbered statements below are informal previews.

Theorem 1.1 (Regional amortization; informal version of Theorem 3.1)**.** *Suppose the B inputs of a batch occupy R distinct linear regions and their region labels are given. Then f can be evaluated on the whole batch using $O\left(BN + Rd_0T + d_0d_LB\right)$ arithmetic operations, where N is the number of hidden units. In the uniform-width case ($d_\ell = d$, $T = \Theta(Ld^2)$), whenever $B \geq RLd$ this is $O\left(Bd\max(d, L)\right)$, a factor of $\Theta\left(\min(L, d)\right)$ below the pointwise cost $\Theta(BLd^2)$, and a factor of $\Theta(L)$ in the regime $L = O(d)$ in which networks are built.*

The labels assumed here are exactly what point location produces, and Theorem 1.2 turns that into a reduction.

Theorem 1.2 (Evaluation reduces to point location; informal version of Theorem 3.2)**.** *If the region labels of a batch can be computed in time $\mathrm{Loc}(f, X)$, then the batch is evaluated in time $\mathrm{Loc}(f, X) + O\left(BN + Rd_0T + d_0d_LB\right)$. At uniform width, a point-location cost of $o(BLd^2)$ therefore gives an evaluation cost of $o(BLd^2)$ along families with $L, d \to \infty$ and $R = o\left(B/d\right)$. The converse fails: when $W_L = b_L = 0$, checking that final layer and writing the constant output costs $O(d_Ld_{L-1} + Bd_L)$, while producing the hidden labels writes BN bits. Point location is therefore strictly the harder of the two, and it is the bottleneck.*

Together these two results concentrate the entire difficulty of beating pointwise evaluation in the geometric part. Each result that follows makes point location cheap in some regime, or says exactly what it costs. We begin with random networks under conditions that keep a batch in few regions and allow its labels to be certified cheaply.

Theorem 1.3 (Average-case region count; informal version of Theorems A.3 and A.6)**.** *Under a bounded zero-crossing density, the expected number of distinct regions occupied by a deterministic batch is at most $1 + c_{\mathrm{d}}N \cdot \mathrm{MST}(X)$, linear in the number of neurons and in the length of a minimum spanning tree of the batch. Independent Gaussian weights at variance $2/$fan-in and Gaussian biases of fixed positive variance satisfy this hypothesis with c_{d} independent of depth and width. For B points spread through a ball of fixed dimension d_0 the count is therefore sublinear in B, and the algebra amortizes once B exceeds a threshold exponential in d_0, but the labels are not produced for free: obtaining them is the point-location problem itself, which the next result removes from the loop.*

Theorem 1.4 supplies those labels without point location, through a certificate instead.

Theorem 1.4 (Average-case labels by certificates; informal version of Theorems A.8 and A.17)**.** *Region membership can be certified by a single distance comparison in input space: a point within the distance-to-nearest-boundary of an evaluated point shares its region and is evaluated by the same affine map. Caching this certificate and recomputing a label only when it fails evaluates a batch presented in an order of length Λ through a random network of $L \geq 2$ layers of width d, with $d_0 = O(d)$, in expected time $O\left(Bd_0d\right)$, once $B = \Omega\left(L^3d^3\rho_{\mathrm{m}}\Lambda + Ld\right)$ with ρ_{m} bounding the density of the margin distribution near zero, a factor of $\Theta\left(Ld/d_0\right)$ below pointwise, with every operation polynomial in the dimension. The number of recomputations is $O\left(1 + \sqrt{BN\rho_{\mathrm{m}}\Lambda}\right)$ in expectation, by charging each one to a disjoint stretch of the order. This closes the loop of Theorem 3.2: it supplies the labels that Theorem 3.1 needs without ever performing point location, so no step of the method is exponential in the dimension, though the batch size at which it wins still is.*

Both of the foregoing exploit randomness. With generic weights and an unstructured batch no shortcut survives, and the problem instead acquires a sharp identity.

Theorem 1.5 (Batched evaluation equals matrix multiplication; informal version of Theorems B.2 and B.5)**.** *For generic dense weights and an arbitrary high-dimensional batch, batched evaluation is equivalent to matrix multiplication. Fast matrix multiplication evaluates a batch of $B \geq d$ inputs*

Table 1: Operation counts for evaluating a batch of B inputs through an L-layer network of uniform width d and size $T = \Theta(Ld^2)$. Here R is the number of distinct regions met, ω, α are the square and dual matrix-multiplication exponents, κ is the forward condition number and ξ the stable-rank profile of the weights, which is dominated by κ/ε when the biases vanish. Every row computes the output exactly except the last, which is randomized and approximate, holding with probability $1-\delta$ and suppressing a $\log(L/\delta)$ factor, and it assumes $B \geq d$ — see Theorem C.4. *The fourth row is an expectation over the random network under Theorem A.10. It assumes at least two layers, $L \geq 2$, and hidden and output widths d and $d_0 = O(d)$. The batch is processed in an order of length Λ, with size $B = \Omega\left(L^3 d^3 \rho_{\mathrm{m}} \Lambda + Ld\right)$ — see Theorem A.17.

Setting	Pointwise	This paper
Arbitrary batch, labels given	$\Theta(BLd^2)$	$O\left(RLd^3 + Bd^2 + BLd\right)$
Arbitrary batch ($B \geq d$), no labels	$\Theta(BLd^2)$	$O\left(LBd^{\omega-1+o(1)}\right)$
Thin batch ($B \leq d^\alpha$), no labels	$\Theta(BLd^2)$	$Ld^{2+o(1)}$
Random network, ordered batch*, no labels	$\Theta(BLd^2)$	$O\left(Bd_0 d\right)$ expected
Arbitrary batch, ε-approximate	$\Theta(BLd^2)$	$O\left(BLd\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$

through a uniform-width network in $O\left(LBd^{\omega-1+o(1)}\right)$ operations, where $\omega < 2.372$ is the matrix-multiplication exponent, beating the pointwise $\Theta\left(BLd^2\right)$; a thin batch of $B \leq d^\alpha$ inputs, with $\alpha > 0.32$ the dual matrix-multiplication exponent, is evaluated for the asymptotic price of a single one. Conversely, any batched-evaluation algorithm that reads the weights yields a matrix-multiplication algorithm of the same cost, so to beat the per-layer product one would have to compute the sign pattern of a matrix product without the product.

All of the above computes f exactly. Permitting a controlled error removes the discrete geometry from the problem altogether.

Theorem 1.6 (Approximate evaluation; informal version of Theorems C.1 and C.4)**.** *For approximation the apparent obstacle is illusory: because the rectifier is Lipschitz, a perturbed pre-activation that flips a sign cannot blow up the forward pass, so the discrete pattern is never needed. Sketching each layer's product and applying exact rectifiers evaluates an* arbitrary *batch of $B \geq d$ inputs to relative error $\varepsilon \leq 1$, with probability $1-\delta$ for every $\delta \in (0, 1/2)$ and up to a $\log(L/\delta)$ factor, in time $O\left(BLd\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$, a factor of $\Theta\left(d/\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$ below pointwise, where κ is the network's forward condition number and $\xi = \sum_\ell \|W_\ell\|_F / \|W_\ell\|_2$ its stable-rank profile, with $\|\cdot\|_F$ the Frobenius norm, $\|\cdot\|_2$ the spectral norm and $\|W\|_F^2 / \|W\|_2^2$ the stable rank of W; when the biases vanish $\kappa \geq \xi$ and the bound is the $O\left(BLd\kappa^2/\varepsilon^2\right)$ that conditioning alone dictates. The obstacle is not sign errors but conditioning: the speedup is real when the layers have low stable rank, and not when they merely preserve norms, which forces $\kappa \geq L\sqrt{d}$ and no speedup at all.*

Table 1 collects the regimes and the corresponding bounds.

1.2 Our Techniques

Most of the results refine the geometry–algebra decomposition, two of them through the randomness of the network; one is purely algebraic, and one leaves exact computation. The technical content rests on a few distinct ideas, which we preview here in the order the paper develops them.

Accumulating the regional map. On a region with sign pattern p the network equals $A_p x + c_p$, with $A_p = W_L \Delta_{L-1} W_{L-1} \cdots \Delta_1 W_1$ and each Δ_ℓ a $0/1$ diagonal mask. Multiplying this product right to left while carrying a width-d_0 right factor forms (A_p, c_p) in $O(d_0 T)$ time, after which applying the map to any input costs only $O(d_0 d_L)$. The gap between the one-time $O(d_0 T)$ of forming the map and the $O(d_0 d_L)$ of using it is the depth factor saved in Theorem 1.1, up to one term that cannot be avoided: an algorithm handed the labels must read them, and the N bits per input cost $\Theta(BN)$, which is what caps the saving at $\Theta(\min(L, d))$.

A connectivity bound for the average case. The number of distinct patterns in a batch is at most one plus the number of edges of a spanning tree whose endpoints disagree on some unit, and the

expected number of such disagreements along an edge is bounded by the one-dimensional boundary-crossing density of the random network, which is linear in the number of neurons. A minimum spanning tree then bounds the expected region count.

Certificates in place of location. A complementary idea avoids point location altogether: the distance from a point to the nearest activation boundary is a radius within which the region, and hence the affine map, is unchanged. A single distance comparison therefore certifies membership, and only a genuine boundary crossing forces a recomputation. Counting those recomputations is the delicate part, since the point that triggers one is selected by the very event being counted; the way out is to charge each recomputation either to a disjoint stretch of the processing order, of length at least the anchor's safe radius, or to a batch point whose margin is small, and to let anti-concentration bound only the second kind.

Matrix multiplication, up and down. A batched forward pass is a sequence of dense matrix products interleaved with the rectifier, so fast matrix multiplication applies layer by layer, and the rectangular case with a sublinear batch makes the products run in time linear in their output. The matching lower bound is the one-line observation that a single linear layer on a batch is itself a matrix product, for an algorithm that is handed the weights rather than allowed to preprocess them.

Lipschitz stability and sketching. Stepping outside exact computation: because the rectifier is Lipschitz, sketching the layer products and applying exact rectifiers approximates the whole forward pass with error that grows only by the layer norms — provided the error is carried through the exact map and the sketch is applied to the already-perturbed activations, and not the other way round, which would leave the unbounded operator $\left\|WSS^\top\right\|_2$ in the recursion. The obstacle to approximation is therefore the network's conditioning rather than any flipped sign.

1.3 Other Related Work

The piecewise-linear structure of ReLU networks is the basis of a large body of work on counting linear regions and relating their number to depth and width (Montúfar et al., 2014; Telgarsky, 2016; Raghu et al., 2017; Serra et al., 2018); Hanin & Rolnick (2019a) proves for random networks, and observes for trained ones, that the number of activation patterns realized is far below the worst-case bound, which is the average-case reason our region-based amortization can help in practice.

The exact piecewise-linear structure is also computed rather than only counted. Sotoudeh & Thakur (2019) compute the exact partition of a line segment into pieces on which the network is affine, for rectified, max-pooling, batch-normalization, fully connected and convolutional layers, and use it to evaluate integrated gradients (Sundararajan et al., 2017) exactly rather than by sampling; Humayun et al. (2023) compute the exact partition of a planar region into cells by traversing the cell-adjacency graph, and characterize the decision boundary on it; Vincent & Schwager (2021) run the same traversal over the whole input space in any dimension, finding a cell's facets by one linear program per constraint, to compute the reachable sets of a network. All three are stated for what they compute; this paper treats the batch in the ambient dimension without enumerating, and the quantity that governs it there, the number of regions a batch meets, is what Appendix A bounds in expectation.

Region-counting in a neighborhood rather than globally is studied by Zhu et al. (2020), who bound the number of linear regions in a ball, and by Patel & Montúfar (2025), whose local complexity measures the density of regions over a data distribution; neither defines the safe radius of Theorem A.7 or uses one to avoid recomputation.

The radius itself is known: Croce et al. (2019) define it as the distance from a point to the boundary of its linear region and use it, with the distance to the decision boundary, as a pointwise robustness certificate; Theorem A.7 is the ℓ_2 case of their lemma, and what is ours is the use, evaluating a second point by the first point's affine map. The closest practical counterpart to that certificate is certified caching (Xiang et al., 2026), where a cache hit is licensed by a radius in a Lipschitz-constrained feature space computed from the classification margin of a lightweight guard network; that radius certifies that the guard network's label is constant on the ball, agreement with the main model being empirical, where Theorem A.7 certifies exact region membership and therefore the exact affine map. Region-by-region reasoning is also the basis of verification (Katz et al., 2017).

On the purely algebraic side, multiplication of a vector by a fixed matrix admits subquadratic algorithms after preprocessing when the matrix and the vector lie in a finite semiring (Williams, 2007), and when the matrix takes values in a finite alphabet and the vector is real (Liberty & Zucker, 2009); for quantized weights these speed up the affine layers irrespective of the rectifier and compose with our results.

Our high-dimensional bounds instead invoke fast dense matrix multiplication (Strassen, 1969; Le Gall, 2014) and its rectangular form (Le Gall & Urrutia, 2018).

The safe-ball certificate of Theorem A.7 is in the spirit of kinetic data structures (Basch et al., 1999), where a cached answer stands until its certificate expires.

Finally, Theorem B.15 sits beside the study of which parameter transformations leave a ReLU network's function unchanged, where positive scaling at a neuron and permutation within a layer are the standard symmetries and the question is what else there is (Grigsby et al., 2023); we need only the elementary characterization, and claim nothing beyond it.

1.4 Discussion

The decomposition identifies where batching can help. For an arbitrary batch through generic dense weights, region labels are expensive; by Theorem 3.3, they can cost more than evaluation itself. Even so, Theorem B.2 expresses the batched pass as a sequence of matrix products. Fast matrix multiplication beats the pointwise baseline by a factor of $\Theta\left(d^{3-\omega-o(1)}\right)$, and a thin batch of up to d^{α} inputs is asymptotically free. This is the best one can hope for per layer without new ideas, since Theorem B.5 shows the problem is matrix multiplication; whether the L layers amortize into one is open (Theorems B.12 and B.14).

Larger, geometry-aware gains require structure that makes point location itself cheap. Locality is the cleanest such structure: nearby inputs tend to share a region, so a batch concentrated in a few regions is evaluated a factor of $\Theta\left(\min(L, d)\right)$ faster once labeled, and through a random network its labels are certified rather than located. These are exactly the batches that arise in local sensitivity analysis, finite differencing, line search, and path-based attribution, where the inputs are deliberately close to one another or to a base point. We view the practical contribution as a recipe: do not recompute the geometry that is already known.

1.5 Open Problems

We close with the problems the paper leaves open and the partial progress made on each along the way.

Sign patterns versus products. Appendix B settles the labels-versus-evaluation question in the arithmetic model at the level of one layer: batched evaluation is matrix multiplication, beatable by exactly the matrix-multiplication factor per layer. Two things remain. The first is whether the L per-layer rounds amortize into fewer. Theorem B.12 shows depth enters linearly at the output rate, Theorem B.14 shows that grouping by pattern strictly loses, and Theorem B.16 gives a structured class whose depth does collapse; Theorem B.13 says why the fine-grained toolkit does not reach a depth-compounding lower bound.

The second is point location proper. By Theorem B.5 the activation pattern of a layer is the positivity pattern of a fixed matrix applied to the batch, which is upper bounded by forming the product. The open question is whether it can be computed strictly faster than the product, in dimension growing with the data. A positive answer would improve the matrix-multiplication upper bound for batched labeling; Theorem B.7 locates the residue in growing dimension, a single layer in fixed dimension being already output-optimal. Theorem B.9 anchors that residue: a single layer's labels compute the Boolean product of its weights and the batch, so output-optimal labeling would give $n^{2+o(1)}$ Boolean matrix multiplication, and with it $n^{2+o(1)}$ triangle detection. We prove that direction only; whether labeling reduces back to a Boolean product, and so whether the sign pattern is strictly cheaper than the product, is the frontier.

Point location without the curse of dimensionality. Appendix A dodges high-dimensional point location altogether in the average case: a safe-ball certificate evaluates concentrated batches in time polynomial in the dimension, a factor of $\Theta(Ld/d_0)$ below pointwise. It rests on the batch being kept away from the boundaries, the margin condition of Theorem A.10. The open task is to remove this condition: an algorithm subexponential in the ambient dimension for a worst-case batch, or a matching conditional lower bound. By Theorem B.5 this is the same frontier as computing the sign pattern of a matrix product without the product. Its average-case form is settled up to one parameter: Theorem A.16 bounds the failures under the bounded margin density of Theorem A.10 with no hypothesis about the anchors, and Theorem A.18 shows that density cannot be dropped, so for a random network the assumption is tight. We bound the constant ρ_{m} in that assumption at every depth for random biases (Theorem A.11) and for bias-free Gaussian networks (Theorem A.14), but not for deterministic biases in general, and Theorem A.19 notes that the exponential in d_0 has moved into the batch size at which the method wins rather than disappeared. Beyond that lies the adversarial, worst-case batch, the same growing-dimension frontier as Theorem B.7.

Approximate evaluation of ill-conditioned networks. Appendix C shows that the apparent obstacle to approximation, the compounding of sign errors through depth, is illusory: the rectifier is Lipschitz, so sketching the layer products and applying exact rectifiers evaluates an arbitrary batch of $B \geq d$ inputs to relative error ε a factor of $\Theta\left(d/\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$ below pointwise, up to a logarithmic factor, gated by the forward condition number κ and, unless the biases vanish, by the stable-rank profile ξ as well.

What remains is to control κ, and Theorem C.5 rules out the first guess: norm preservation forces $\kappa \geq L\sqrt{d}$ and no speedup. Theorem C.6 shows that low stable rank with bounded spectral norm and vanishing biases keeps κ small, and Theorem C.8 shows that the κ in the guarantee cannot be removed by any analysis that controls a layer-wise approximation through its relative Frobenius error alone. Shaving the ε^{-2} is possible for decaying spectra, where Theorem C.9 attains relative error ε with rank logarithmic in $1/\varepsilon$ and in the depth by exact truncation, while on a flat spectrum Theorem C.10 shows sketching cannot help at all. Against every method, Theorem C.11 ties entrywise approximation to Boolean matrix multiplication. What is open is whether the κ dependence is intrinsic or an artifact of sketching; Theorem C.12 explains why the obvious unconditional statement is not merely unproved but false, since Theorem B.3 already evaluates exactly below the pointwise baseline.

1.6 Paper Organization

The main text states every result, fixes the preliminaries and proves the decomposition; the other proofs are in the appendix, which the page limit places there rather than in the body. Section 2 fixes the computational model, the networks, and the definition of linear regions. Section 3 proves the geometry–algebra decomposition: the regional-amortization bound and the equivalence with point location. Appendix A bounds the expected region count of a batch and then supplies its labels by a safe-ball certificate, closing the loop with Theorem 3.1 without any point location. Appendix B confronts the generic high-dimensional worst case, where the problem is matrix multiplication. Appendix C leaves exact computation for approximation, where sketching gives an arbitrary-batch speedup gated by conditioning, not by sign errors.

2 Preliminaries

2.1 The Computational Model

We work in the standard real-arithmetic model: a memory cell holds a real number, and each arithmetic operation in $\{+, -, \times, \div\}$, each comparison, and each value written to the output costs one unit. This is the natural model for numerical linear algebra and computational geometry and is the one in which the pointwise baseline below is stated; every bound below counts operations in it, the arithmetic-circuit statement of Theorem B.5 included.

2.2 ReLU Networks and Their Linear Regions

Every bound below is stated for the following model of a ReLU network. It uses two pieces of notation. For a positive integer m we write $[m]$ for the set $\{1, \dots, m\}$. The rectifier $\sigma\colon \mathbb{R} \to \mathbb{R}$ is the function $\sigma(z) = \max(z, 0)$, which leaves a positive number unchanged and replaces a negative one by 0; applied to a vector, it acts on each entry separately.

Definition 2.1 (ReLU network)**.** An L-layer ReLU network with widths $d_0, d_1, \dots, d_L$ is the function $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ defined by $z_0 = x$ and, for $\ell \in [L]$,

$$h_\ell(x) = W_\ell z_{\ell-1} + b_\ell, \qquad z_\ell = \sigma\left(h_\ell(x)\right) \text{ for } \ell < L, \qquad f(x) = h_L(x), \tag{1}$$

with $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{d_\ell}$. The *size* of f is $T = \sum_{\ell=1}^{L} d_\ell d_{\ell-1}$, the number of weights, and we set $d = \max_\ell d_\ell$ and $N = \sum_{\ell=1}^{L-1} d_\ell$ (the number of hidden units).

We say f has *uniform width* d when $d_0 = d_1 = \cdots = d_L = d$, in which case $T = \Theta\left(Ld^2\right)$ and $N = \Theta(Ld)$; where only the hidden widths are equal we say so and carry d_0 and d_L explicitly.

Vectors are columns. For a vector z we write z^j for its j-th entry. For a matrix W we write W^j for its j-th row. With this notation, the j-th entry of Wz is $(Wz)^j = W^j z$. A quantity that carries both a layer index ℓ and a unit index j writes the two as a pair. For instance, $h_{\ell j}$ denotes the j-th entry of h_ℓ, the pre-activation of unit j in layer ℓ.

We record the cost of the pointwise baseline exactly. At layer ℓ the product $W_\ell z_{\ell-1}$ of a $d_\ell \times d_{\ell-1}$ matrix by a vector uses $d_\ell d_{\ell-1}$ multiplications and $d_\ell\left(d_{\ell-1} - 1\right)$ additions, the bias adds d_ℓ additions, and for $\ell < L$ the rectifier $z_\ell = \sigma\left(h_\ell\right)$ uses d_ℓ comparisons; the layer costs $\Theta\left(d_\ell d_{\ell-1}\right)$ operations. Summing over layers,

$$\sum_{\ell=1}^{L} \Theta\left(d_\ell d_{\ell-1}\right) + \sum_{\ell=1}^{L-1} d_\ell = \Theta\left(T\right) + O(N) = \Theta\left(T\right), \tag{2}$$

the last equality because $T = \sum_{\ell=1}^{L} d_\ell d_{\ell-1} \geq \sum_{\ell=1}^{L-1} d_\ell = N$ once every $d_{\ell-1} \geq 1$. A batch of B inputs evaluated independently costs $\Theta\left(BT\right)$, the baseline we compare against.

The next definition records which units are active. For a statement φ, the indicator $\mathbb{1}(\varphi)$ is 1 when φ holds and 0 otherwise.

Definition 2.2 (Activation pattern and linear region)**.** The *activation pattern* of x is $\pi(x) = \left(\pi_\ell(x)\right)_{\ell=1}^{L-1}$ with $\pi_\ell(x) = \left(\mathbb{1}\left(h_{\ell j}(x) > 0\right)\right)_{j \in [d_\ell]} \in \{0,1\}^{d_\ell}$. For a pattern p, the corresponding *linear region* is $\mathcal{R}_p = \{x \in \mathbb{R}^{d_0} \mid \pi(x) = p\}$, and $\mathcal{R}$ denotes the set of nonempty regions.

For $L = 1$ there are no hidden units and $\pi(x)$ is the empty tuple, so the reductions of Appendix B use networks with one hidden layer, whose pattern is π_1.

Distinct patterns may induce the same affine map, so the number of patterns a batch realizes is at least, and can exceed, the number of distinct affine pieces of f it meets. Every bound below is stated in patterns, which is the quantity our algorithms pay for.

The following standard fact is what the decomposition rests on; we record the explicit form of the affine map because we use it algorithmically.

Lemma 2.3 (Regional affine form)**.** *On the closure of any region with pattern p, the network is affine: $f(x) = A_p x + c_p$, where, with $\Delta_\ell = \operatorname{diag}\left(p_\ell\right)$,*

$$A_p = W_L \Delta_{L-1} W_{L-1} \cdots \Delta_1 W_1, \tag{3}$$

and c_p is given by the recursion $M_1 = W_1$, $v_1 = b_1$, and for $\ell \geq 2$, $M_\ell = W_\ell \Delta_{\ell-1} M_{\ell-1}$ and $v_\ell = W_\ell \Delta_{\ell-1} v_{\ell-1} + b_\ell$, with $A_p = M_L$ and $c_p = v_L$. On that closure every pre-activation is affine as well, $h_\ell(x) = M_\ell x + v_\ell$ for every $\ell \in [L]$. Forming (A_p, c_p) takes $O(d_0 T)$ operations.

The proof is in Appendix D.

The recursion of Theorem 2.3 also describes the region itself, which is what lets Appendix A treat a region as a polyhedron.

Lemma 2.4 (Regions are polyhedra). *Fix a pattern p and let $\tilde{h}_\ell(x) = M_\ell x + v_\ell$ be the affine forms of the recursion in Theorem 2.3. Then*

$$\mathcal{R}_p = \left\{ x \in \mathbb{R}^{d_0} \mid \tilde{h}_{\ell j}(x) > 0 \text{ where } p_{\ell j} = 1, \quad \tilde{h}_{\ell j}(x) \le 0 \text{ where } p_{\ell j} = 0 \right\}, \tag{4}$$

an intersection of N half-spaces. In particular $\mathcal{R}_p$ is convex, its closure is a polyhedron, and every facet of that polyhedron is carried by the zero set of one $\tilde{h}_{\ell j}$.

The proof is in Appendix D.

3 The Geometry–Algebra Decomposition

This section proves the two halves of the decomposition. The algebra amortizes once region labels are known, giving the factor-$\Theta(\min(L, d))$ saving of Theorem 1.1, and producing those labels is then all that remains, a batched point-location problem.

3.1 Forming and Reusing the Regional Map

Theorem 2.3 says that once a region is known, the network on that region is a single affine map costing $O(d_0 d_L)$ to apply, against $O(T)$ for a fresh forward pass. When $d_0 d_L = O(T)$, which holds whenever the input and output are no wider than the network is deep and wide, applying the map is the cheaper of the two, and since forming it costs $O(d_0 T)$ the map pays for itself as soon as it is reused on enough inputs. This is the entire content of regional amortization, and it fails exactly when the map itself is the large object: for $L = 2$, $d_0 = d_2 = n$ and $d_1 = 1$ one has $T = 2n$ while A_p has n^2 entries, and no group size recovers the cost of writing it down.

3.2 Regional Amortization

The costs just described, together with the cost of reading the labels, add up to the following bound for a whole batch whose labels are given.

Theorem 3.1 (Regional amortization). *Let a batch of B inputs $x_1, \dots, x_B$ be given together with their activation patterns, and suppose the inputs occupy $R = |\{\pi(x_i) : i \in [B]\}|$ distinct regions, where $|S|$ is the number of elements of a finite set S. Then f is evaluated on the whole batch in*

$$O\left(BN + Rd_0T + d_0 d_L B\right) \tag{5}$$

operations. In particular, with $d_\ell = d$ and $T = \Theta(Ld^2)$, the bound is $O\left(BLd + RLd^3 + Bd^2\right)$, which is $O\left(Bd^2\right)$ whenever $B \ge RLd$ and $L = O(d)$, a factor of $\Theta(L)$ below the pointwise $\Theta(BLd^2)$. For $B \ge RLd$ and arbitrary depth the bound is $O\left(Bd\max(d, L)\right)$ and the saving is a factor of $\Theta(\min(L, d))$.

Proof. Grouping. Each pattern $p_i = \pi(x_i)$ is the concatenation of the $L - 1$ hidden sign vectors, an N-bit key. Partitioning $[B]$ into the R classes of equal key, for instance by a radix pass over the keys, costs $O(BN)$. We charge this term in full: it is the cost of reading the supplied labels, which no algorithm that consumes them can avoid, so the bound of Eq. (5) is at least the size of its own input. It is always dominated by the pointwise baseline $\Theta(BT)$ because $N \le T$, but it is not dominated by the two terms that follow, which is what limits the saving below.

Forming the maps. For each of the R groups, Theorem 2.3 forms (A_r, c_r) in $O(d_0 T)$, for $O(Rd_0T)$ in total.

Applying the maps. Let group r hold its B_r inputs as the columns of $X_r \in \mathbb{R}^{d_0 \times B_r}$. The product $A_r X_r$, of shapes $d_L \times d_0$ and $d_0 \times B_r$, costs $O(d_L d_0 B_r)$, and broadcasting c_r across the columns costs $O(d_L B_r)$. Summing over groups and using $\sum_r B_r = B$,

$$\sum_{r=1}^{R} O\left(d_L d_0 B_r + d_L B_r\right) = O\left(d_L d_0 B\right), \tag{6}$$

the broadcast term being absorbed since $d_0 \ge 1$. Adding the three contributions gives Eq. (5), that is, $O(BN + Rd_0T + d_0 d_L B)$.

Uniform width. Substituting $T = \Theta\left(Ld^2\right)$, $N = \Theta(Ld)$ and $d_0 = d_L = d$ gives $O\left(BLd + RLd^3 + Bd^2\right)$. The middle term is at most the last exactly when $RLd^3 \le Bd^2$, that is $B \ge RLd$, leaving $O\left(BLd + Bd^2\right) = O\left(Bd\max(d, L)\right)$. Against the pointwise $\Theta\left(BLd^2\right)$ this is a saving of

$$\frac{BLd^2}{Bd\max(d, L)} = \frac{Ld}{\max(d, L)} = \min(L, d), \tag{7}$$

which is $\Theta(L)$ exactly when $L = O(d)$, the regime in which networks are used: reading the N label bits of an input is $\Theta(Ld)$ work, and it overtakes the $\Theta\left(d^2\right)$ of applying the region's map once the network is deeper than it is wide. Algorithm 1 is the procedure just costed. □

Algorithm 1 REGIONALEVAL: evaluate a labeled batch by amortized affine maps

Require: network f; inputs $x_1, \dots, x_B$; patterns $p_1, \dots, p_B$
Ensure: $f(x_1), \dots, f(x_B)$
1: partition $[B]$ into groups $G_1, \dots, G_R$ with equal patterns
2: **for** $r \leftarrow 1$ to R **do**
3: form (A_r, c_r) from the shared pattern by the recursion of Theorem 2.3
4: **for** $i \in G_r$ **do**
5: $y_i \leftarrow A_r x_i + c_r$
6: **return** $y_1, \dots, y_B$

3.3 EVALUATION REDUCES TO POINT LOCATION

Combining the cost of point location with the bound of Theorem 3.1 gives the reduction previewed in Theorem 1.2.

Proposition 3.2 (Evaluation reduces to point location)**.** *Fix a network f and a batch $X = \{x_1, \dots, x_B\}$ of B inputs occupying R regions. Write* $\mathrm{Eval}(f, X)$ *and* $\mathrm{Loc}(f, X)$ *for the least number of operations, in the model of Section 2.1, with which an algorithm given f and X produces respectively the outputs $f(x_i)$ and the patterns $\pi(x_i)$; producing the patterns is what we call* labeling *the batch, and* Loc *is its cost. Then*

$$\mathrm{Eval}(f, X) \le \mathrm{Loc}(f, X) + O\left(BN + Rd_0T + d_0d_LB\right), \qquad \mathrm{Loc}(f, X) = O\left(BT\right). \tag{8}$$

Consequently a batched point-location procedure of cost $o(BT)$ yields a batched evaluation procedure of cost $o(BT)$ whenever the additive algebraic term is also $o(BT)$. At uniform width this holds along families with $L, d \to \infty$ and $R = o\left(B/d\right)$: the three terms of Theorem 3.1, divided by BT, are respectively $O(1/d)$, $O(Rd/B)$, and $O(1/L)$.

The proof is in Appendix E.

The reduction runs one way only, and the failure of the converse is worth stating, because it is what makes point location the harder of the two problems rather than an equal partner.

Remark 3.3 (The converse fails). It is tempting to read the second bound of Eq. (8) as saying that labeling is no harder than evaluation, since a forward pass exposes every pre-activation. What it says is weaker: labeling is no harder than the *baseline*. An evaluator faster than the baseline need not expose any pre-activation at all. Take $W_L = 0$ and $b_L = 0$, so that $f \equiv 0$. An evaluator given the weights can check that the final layer is zero and write the output in $O\left(d_Ld_{L-1} + Bd_L\right)$ operations, whatever the hidden layers are; but the hidden pattern is untouched by this choice, and producing it means writing its BN bits, so $\mathrm{Loc}(f, X) \ge BN$ in the model of Section 2.1, which charges one unit per value written to the output. For uniform width d, $B \ge d$ and $L \to \infty$, this gives a gap of $\Omega\left(BLd/(d^2 + Bd)\right) = \Omega(L)$, which is unbounded. The two problems are not equivalent, and every result of this paper that makes location cheap is a genuine gain rather than a restatement. For one hidden layer, Theorem B.5 identifies the labeling problem with the sign pattern of a matrix product applied to the batch.

AI USE STATEMENT

The authors defined the problem. Claude (Anthropic) helped develop the proofs, identify related work, and draft the manuscript. Codex (OpenAI) assisted with reviewing the mathematical arguments, citations, and prose of the ICLR submission. The authors reviewed the AI-assisted work and take responsibility for the paper and any errors.

Under the ICLR 2027 AI Policy for Authors, theoretical modeling, claim formulation, proof ingredients, and proof writing require disclosure; drafting, literature analysis, and reference formatting are recommended for disclosure. Translation, data, experiments, and research code do not arise in this paper.

REPRODUCIBILITY STATEMENT

All results are analytical; no experiments, datasets, or research code are involved. The cost model and assumptions for the deterministic and random-network bounds are stated in the main text. The proofs, algorithms, and reductions are given in the body and appendices, with cited external results identified where they are used.

REFERENCES

Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2005–2039, 2025.

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

Julien Basch, Leonidas J. Guibas, and John Hershberger. Data structures for mobile data. *Journal of Algorithms*, 31(1):1–28, 1999.

Jillian Beardwood, John H. Halton, and John M. Hammersley. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4):299–327, 1959.

Markus Bläser. Fast matrix multiplication. *Theory of Computing, Graduate Surveys*, 5:1–60, 2013.

Timothy M. Chan and Da Wei Zheng. Hopcroft's problem, $\log^*$ shaving, two-dimensional fractional cascading, and decision trees. *ACM Transactions on Algorithms*, 20(3):24, 2024. doi: 10.1145/3591357.

Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 205–214, 2009.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of ReLU networks via maximization of linear regions. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2057–2066, 2019.

Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Jeff Erickson. New lower bounds for Hopcroft's problem. *Discrete & Computational Geometry*, 16 (4):389–418, 1996.

Esther Ezra, Sariel Har-Peled, Haim Kaplan, and Micha Sharir. Decomposing arrangements of hyperplanes: VC-dimension, combinatorial dimension, and point location. *Discrete & Computational Geometry*, 64(1):109–173, 2020.

L. Few. The shortest path and the shortest road through n points. *Mathematika*, 2(2):141–144, 1955.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

J. Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of ReLU networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 11734–11760, 2023.

Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 359–368, 2019a.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2596–2604, 2019b.

John E. Hopcroft and Jean Musinski. Duality applied to the complexity of matrix multiplication and other bilinear forms. *SIAM Journal on Computing*, 2(3):159–173, 1973.

Ahmed Imtiaz Humayun, Randall Balestriero, Guha Balakrishnan, and Richard G. Baraniuk. SplineCam: Exact visualization and characterization of deep network geometry and decision boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3789–3798, 2023.

Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978.

Daniel M. Kane and Jelani Nelson. Sparser Johnson–Lindenstrauss transforms. *Journal of the ACM*, 61(1):4:1–4:23, 2014.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the 29th International Conference on Computer Aided Verification (CAV)*, pp. 97–117, 2017.

François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pp. 296–303, 2014.

François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith–Winograd tensor. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1029–1046, 2018.

Edo Liberty and Steven W. Zucker. The Mailman algorithm: A note on matrix-vector multiplication. *Information Processing Letters*, 109(3):179–182, 2009.

Grazia Lotti and Francesco Romani. On the asymptotic complexity of rectangular matrix multiplication. *Theoretical Computer Science*, 23(2):171–185, 1983.

Stefan Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106 (2):286–303, 1993.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pp. 2924–2932, 2014.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

Niket Nikul Patel and Guido Montúfar. On the local complexity of linear regions in deep ReLU networks. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 48335–48370, 2025.

Loren K. Platzman and John J. Bartholdi III. Spacefilling curves and the planar travelling salesman problem. *Journal of the ACM*, 36(4):719–737, 1989.

Maithra Raghu, Ben Poole, Jon M. Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2847–2854, 2017.

Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 143–152, 2006.

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

Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4558–4566, 2018.

Matthew Sotoudeh and Aditya V. Thakur. Computing linear restrictions of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3319–3328, 2017.

Matus Telgarsky. Benefits of depth in neural networks. In *Proceedings of the 29th Conference on Learning Theory (COLT)*, pp. 1517–1539, 2016.

Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *Journal of the ACM*, 65(5):27:1–27:38, 2018.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: From alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3792–3835, 2024.

Joseph A. Vincent and Mac Schwager. Reachable polyhedral marching (RPM): A safety verification algorithm for robotic systems with deep neural network components. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9029–9035, 2021.

R. Ryan Williams. On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity. In *Proceedings of the Twenty-Ninth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1207–1215, 2018.

Ryan Williams. Matrix-vector multiplication in sub-quadratic time (some preprocessing required). In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 995–1001, 2007.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Zhengzhe Xiang, Yinlin Chen, Fuli Ying, Binbin Zhou, Hailiang Zhao, and Schahram Dustdar. LipCache: A local inference proxy with certified caching for edge image classification service. *CoRR*, abs/2608.13144, 2026.

Rui Zhu, Bo Lin, and Haixu Tang. Bounding the number of linear regions in local area for neural networks with ReLU activations. *CoRR*, abs/2007.06803, 2020.

A THE AVERAGE CASE

Theorem 3.1 charges the algebra of a batch to the number R of distinct regions it occupies, and Theorem 3.2 leaves the cost of the labels as the open term. Both are worst-case statements, and a generic batch of $B \leq |\mathcal{R}|$ points has $R = B$. We now show that for the random networks studied by Hanin and Rolnick the region count is governed by a one-dimensional crossing density, so geometrically concentrated batches occupy few regions in expectation; the algebra then amortizes, and the difficulty migrates to point location. The certificate of Appendix A.4 then removes point location from the loop: an evaluated point carries a ball inside its region, checkable by one distance comparison, a label is recomputed only when that check fails, and Theorem A.17 shows that for a concentrated batch through a random network the failures are few.

A.1 A CONNECTIVITY BOUND ON DISTINCT PATTERNS

We begin with a deterministic bound that holds for every network and every batch.

Lemma A.1 (Spanning bound)**.** *Let $X = \{x_1, \dots, x_B\}$ be the batch and let $\mathcal{T}$ be any spanning tree on the vertex set X. For an edge $e = (x, x')$ let $\phi(e) = |\{(\ell, j) : \pi_{\ell j}(x) \neq \pi_{\ell j}(x')\}|$ be the number of hidden units whose activation bit differs at x and x'. Then*

$$R = |\{\pi(x_i) : i \in [B]\}| \leq 1 + \sum_{e \in \mathcal{T}} \mathbb{1}\left(\phi(e) \geq 1\right) \leq 1 + \sum_{e \in \mathcal{T}} \phi(e). \tag{9}$$

Proof. Color each point by its pattern $\pi(x_i)$. Call an edge of $\mathcal{T}$ *monochromatic* if its endpoints share a pattern and *bichromatic* otherwise; an edge is bichromatic exactly when $\phi(e) \geq 1$. Contracting the monochromatic edges merges only equally colored points and leaves a tree whose vertices are the contracted groups and whose edges are the bichromatic edges. A tree on V vertices has $V - 1$ edges, so the number of groups is one more than the number of bichromatic edges. The number of distinct patterns is at most the number of groups, which gives the first inequality; the second uses $\mathbb{1}\left(\phi(e) \geq 1\right) \leq \phi(e)$. □

A.2 EXPECTED REGION COUNT UNDER A CROSSING-DENSITY BOUND

The average-case bound of Theorem A.3 below rests on the following hypothesis, which the paragraph after it relates to random networks.

Assumption A.2 (Bounded crossing density)**.** There is a constant c_{d}, depending on the laws of the weights and biases and not on the segment, such that for every line segment S of length ρ the expected number of zero crossings of the hidden pre-activations along S, counted with multiplicity over the units, is at most $c_{\mathrm{d}} N \rho$.

The assumption counts zero crossings rather than units whose endpoint signs differ, and crossings are what the arguments below use: a pre-activation of a deep unit is piecewise affine along a segment and may cross zero several times while agreeing in sign at the two ends. Hanin & Rolnick (2019b) study boundaries where the network gradient changes, and Hanin & Rolnick (2019a) count activation regions in a cube. Neither cited theorem states the all-unit zero-crossing count with multiplicity required by Theorem A.2; a hidden unit's crossing can leave the output gradient unchanged. For independent random biases, Theorem A.12 below proves Theorem A.2 directly for all hidden units.

At the independent Gaussian initialization of Theorem A.11, the resulting constant is $c_{\mathrm{d}} = \sqrt{2}/(s_{\mathrm{b}}\sqrt{2\pi})$, independent of depth and width. This supplies a concrete random-network regime for the theorem without equating region boundaries with all pre-activation zero crossings. We take the assumption as given and draw out its consequence for batches.

Theorem A.3 (Expected region count of a batch)**.** *Under Theorem A.2, for every deterministic batch X with Euclidean minimum spanning tree of length $\mathrm{MST}(X)$, the number R of distinct regions the batch occupies satisfies*

$$\mathbb{E}\left[R\right] \leq 1 + c_{\mathrm{d}} N \cdot \mathrm{MST}(X), \tag{10}$$

the expectation taken over the random weights and biases.

Proof. Fix a minimum spanning tree $\mathcal{T}$ of X; the batch being deterministic, $\mathcal{T}$ is determined by X alone and does not depend on the random weights and biases. For an edge $e = (x, x')$, a unit has opposite signs at the endpoints only if it changes sign along the straight segment e, so $\phi(e)$ is at most the number of such changes. Write $\|v\|$ for the Euclidean norm of a vector v, so that $\|x - x'\|$ is the length of the edge e. Then Theorem A.2 gives $\mathbb{E}[\phi(e)] \le c_{\mathrm{d}} N \|x - x'\|$. Taking expectations in Eq. (9) and summing over the edges, $\mathbb{E}[R] \le 1 + c_{\mathrm{d}} N \sum_{e \in \mathcal{T}} \|e\| = 1 + c_{\mathrm{d}} N \cdot \mathrm{MST}(X)$. □

The hypothesis in Theorem A.3 that X is fixed before the weights are drawn is not a simplification. The following remark exhibits a batch, chosen after the draw, for which the bound fails.

Remark A.4 (The batch must not depend on the network). The batch in Theorem A.3 is fixed before the network is drawn, and the bound fails for a batch chosen afterwards. Let the weights have a continuous law, so that $W_1^1 \neq 0$ with probability one, and fix $\delta > 0$ with $2c_{\mathrm{d}} N\delta < 1$. On the event $W_1^1 \neq 0$ put $w = \left(W_1^1\right)^\top / \left\|W_1^1\right\|$, pick a point y with $h_{11}(y) = 0$, and take $X = \{y - \delta w, y + \delta w\}$; on the complementary event take two points at distance 2δ. On the first event $h_{11}(y \pm \delta w) = \pm\delta \left\|W_1^1\right\|$, so the two points differ in the bit of the unit $(1, 1)$ and $R = 2$. Hence $\mathbb{E}[R] = 2$, while $\mathrm{MST}(X) = 2\delta$ on every draw and the right-hand side of Eq. (10) is $1 + 2c_{\mathrm{d}} N\delta < 2$. The proof breaks where it applies Theorem A.2 to an edge of the tree. The assumption bounds the expected crossings of a segment fixed before the draw, whereas this segment crosses the zero set of the unit $(1, 1)$ on every draw of the first event, so its expected number of crossings is at least $1 > c_{\mathrm{d}} N \cdot 2\delta$.

A.3 Concentrated Batches in a Ball

The region count of Theorem A.3 is linear in the length of a spanning tree, so what a concentrated batch gains depends on how short a tree it admits.

Lemma A.5 (A short order through a ball)**.** *Let X be a set of B points in a ball of radius r in $\mathbb{R}^{d_0}$, and call the length of an order $x_{\tau(1)}, \ldots, x_{\tau(B)}$ of X the sum $\sum_{i<B} \left\|x_{\tau(i+1)} - x_{\tau(i)}\right\|$ of its consecutive distances. Let K be the least integer with $2^{Kd_0} > B$. Consider the order of the points by their index along the Hilbert curve of a cube of side $2r$ containing the ball, the curve resolved to depth K: that is, by the position along the curve of the dyadic subcube of side $2r2^{-K}$ containing each point, the points of one such subcube taken in any order. This order has length $\Lambda = O\left(rB^{1-1/d_0}\right)$, where the constant depends on d_0 alone, and it is computed in $O(Bd_0 \log B)$. The same bound holds for $\mathrm{MST}(X)$, and for the shortest closed tour through X.*

Proof. The Hilbert curve of the cube visits each dyadic subcube of side $2r2^{-j}$ in one contiguous run of the index, for every $j \ge 0$; that is the recursive property that defines it. Write $J = K - 1$, so that $2^{Jd_0} \le B < 2^{(J+1)d_0}$ by the choice of K. Consider the $B - 1$ consecutive pairs of the order, and charge each pair to the largest $j \le K$ for which both its points lie in one subcube of side $2r2^{-j}$; every pair is charged, since $j = 0$ qualifies for all of them, and a pair of coincident points is charged to K. A pair charged to j has both points in a common subcube of that side, so its length is at most $2r\sqrt{d_0}2^{-j}$.

At every level $j \le K$, two points in different subcubes of side $2r2^{-j}$ are separated in the order by every point of every subcube the curve visits between them: the order sorts the depth-K subcubes along the curve, and a coarser subcube is a contiguous run of them. A pair charged to $j < K$ therefore has its two points in subcubes of side $2r2^{-(j+1)}$ that the curve visits consecutively among those containing points of X. There are at most $2^{(j+1)d_0}$ such subcubes in all, hence at most that many consecutive pairs of them, and the order leaves each subcube once, so at most $2^{(j+1)d_0}$ pairs are charged to j.

Pairs charged to some $j \le J$ contribute at most $\sum_{j \le J} 2^{(j+1)d_0} \cdot 2r\sqrt{d_0}\, 2^{-j} = O\left(r2^{J(d_0-1)}\right) = O\left(rB^{1-1/d_0}\right)$, the geometric sum being dominated by its last term for $d_0 \ge 2$. Pairs charged to K number at most $B - 1$, and each has length at most $2r\sqrt{d_0}2^{-K} \le 2r\sqrt{d_0}2^{-J} = O\left(rB^{-1/d_0}\right)$, since $B < 2^{(J+1)d_0}$ gives $2^{-J} < 2B^{-1/d_0}$; together they contribute a further $O\left(rB^{1-1/d_0}\right)$.

For $d_0 = 1$ the order is the sorted order up to the arrangement of the points within a subcube of side $2r2^{-K}$, so each consecutive pair is at most that side longer than the gap between the left ends of its

two subcubes, and these gaps telescope to at most $2r$; hence $\Lambda \le 2r + (B-1)\cdot 2r2^{-K} < 4r$, which is the same bound.

The depth-K subcube of a point is found level by level, one comparison per coordinate against the center of the current subcube, and the curve's recursion turns the choice at each level into a position in $O(d_0)$, so the K levels cost $O(d_0 K) = O(d_0 + \log B)$ per point, because $(K-1)d_0 \le \log_2 B$. Sorting the B positions costs $O(B \log B)$ comparisons, so the order costs $O(B(d_0 + \log B)) = O(Bd_0 \log B)$ in all.

Finally $\mathrm{MST}(X)$ is at most the length of any spanning path, hence at most Λ, and the shortest closed tour is at most $2 \cdot \mathrm{MST}(X)$: doubling every edge of a minimum spanning tree gives a closed walk of length $2 \cdot \mathrm{MST}(X)$ through every point, and shortcutting its repeated visits does not lengthen it, by the triangle inequality. □

The bound is classical in substance. That some path through n points of a unit cube in $\mathbb{R}^d$ has length $O\left(n^{1-1/d}\right)$ is Few's theorem (Few, 1955), and that ordering the points along a space-filling curve attains it in the plane, in $O(n \log n)$ time, is due to Platzman & III (1989); Theorem A.5 restates that argument for the Hilbert curve in $\mathbb{R}^{d_0}$, with the depth fixed by B, which is the form the certificate needs.

Remark A.6 (Batches in a ball and the residual bottleneck). For B points spread through a ball of radius r in $\mathbb{R}^{d_0}$ with d_0 fixed, the minimum spanning tree has length $O\left(rB^{1-1/d_0}\right)$ by Theorem A.5. That is a deterministic bound for an arbitrary batch, which is what Theorem A.3 needs; the matching almost-sure limit for the tour through i.i.d. points is due to Beardwood et al. (1959). Theorem A.3 therefore gives $\mathbb{E}[R] \le 1 + O\left(c_{\mathrm{d}} N r B^{1-1/d_0}\right)$, which is sublinear in B for fixed parameters: the algebra does amortize, and Theorem 3.1 beats the pointwise baseline in expectation once $B \ge \mathbb{E}[R]\, Ld$, that is once $B = \Omega\left(Ld + (c_{\mathrm{d}} N r L d)^{d_0}\right)$, *provided the labels are known*. The two terms in the threshold respectively absorb the leading 1 and the $c_{\mathrm{d}} N r B^{1-1/d_0}$ term in the region bound. It may be taken in expectation because Eq. (5) is affine in R: the expectation passes through it, and the term RLd^3 is dominated by Bd^2 in expectation exactly when $B \ge \mathbb{E}[R]\, Ld$. Two features stand out. The threshold is exponential in d_0, the curse of dimensionality; and the labels are not produced for free, so obtaining them here is precisely the high-dimensional point-location problem of Section 1.5. The average-case analysis thus shrinks the algebra but relocates the difficulty onto the geometry, reaffirming point location as the crux.

The regions are few, but Theorem 3.1 still needs the labels, and locating the batch in the arrangement to obtain them is the point-location problem of Section 1.5, whose known structures need storage exponential in the dimension (Meiser, 1993; Ezra et al., 2020). We now close the loop with a mechanism that never performs point location at all: each evaluated point carries a *certificate*, a ball inside its region checkable by one distance comparison, and a new label is computed only when the certificate fails.

A.4 The Safe-Ball Certificate

Theorem A.8 needs a radius, computable from one point's pre-activations and their gradients, inside which that point's pattern is guaranteed unchanged. The following lemma supplies it.

Lemma A.7 (Safe ball; the ℓ_2 case of (Croce et al., 2019, Lemma 3.1))**.** *Let y be a point with pattern $p = \pi(y)$, let $h_{\ell j}$ denote the pre-activations and $g_{\ell j}$ their gradients on $\mathcal{R}_p$, so that $g_{\ell j}^{\top}$ is the j-th row of the matrix M_ℓ in the recursion of Theorem 2.3, and set*

$$m(y) = \min_{\ell, j : g_{\ell j} \neq 0} \frac{|h_{\ell j}(y)|}{\|g_{\ell j}\|}, \tag{11}$$

the minimum being over the hidden units with $g_{\ell j} \neq 0$, whose affine form is not constant, and $m(y) = \infty$ if there are none. Then the open ball $\mathbb{B}(y, m(y))$ lies in $\mathcal{R}_p$, so every x with $\|x - y\| < m(y)$ satisfies $\pi(x) = p$ and $f(x) = A_p x + c_p$.

A conservative radius $m'(y) = \left(\min_{\ell, j\,:\, g_{\ell j} \neq 0} |h_{\ell j}(y)|\right)/G \le m(y)$, with $G \ge \max_{\ell j} \|g_{\ell j}\|$ a global bound, is computable from the pre-activation values and the list of nonzero-gradient units when some $g_{\ell j} \neq 0$ and $G > 0$; if all the gradients vanish, take $m'(y) = \infty$ as for $m(y)$.

Proof. The first claim is the ℓ_2 case of Lemma 3.1 of Croce et al. (2019), who show that the ℓ_p-distance from y to the boundary of its region is exactly $\min_{\ell,j} |h_{\ell j}(y)| / \|g_{\ell j}\|_q$ for $1/p + 1/q = 1$; in our terms the region is the polyhedron of Theorem 2.4, cut out by the forms $h_{\ell j}(x) = g_{\ell j}^\top x + v_{\ell j}$, and $m(y)$ is the least distance from y to one of their zero sets, a unit with $g_{\ell j} = 0$ having a constant form that changes sign nowhere. The open ball of radius $m(y)$ meets none of these zero sets, so on it every form keeps the sign it has at y, and the ball lies in $\mathcal{R}_p$ by Theorem 2.4. On $\mathcal{R}_p$ the network is $A_p x + c_p$ by Theorem 2.3. For the conservative radius, $g_{\ell j}^\top$ is the j-th row of $M_\ell = W_\ell \Delta_{\ell-1} \cdots \Delta_1 W_1$ and every $\|\Delta_k\|_2 \le 1$, where $\|\cdot\|_2$ is the spectral norm, the largest singular value, so $\|g_{\ell j}\| \le \prod_{k \le \ell} \|W_k\|_2$ whatever the pattern; hence $G = \max_{\ell < L} \prod_{k \le \ell} \|W_k\|_2$ is a valid global bound, computed once from the weights and independent of the batch. □

Testing the certificate is a single distance comparison in input space, costing $O(d_0)$ with no dependence on N, on L, or on any arrangement; the comparison is made between squared quantities, $\|x - y\|^2$ against $m(y)^2$ with $m(y)^2 = \min h_{\ell j}(y)^2 / \|g_{\ell j}\|^2$, so that no square root is taken and the model of Section 2.1 suffices. This comparison is the operation that dodges the curse of dimensionality of point location, and a failing certificate is the analogue of an expiring certificate in a kinetic data structure (Basch et al., 1999).

A.5 Certificate-Cached Evaluation

The batch is processed in a given order — the order in which the inputs arrive, which in the applications of Section 1.4 is already a short one — and, as in Theorem A.5, we write Λ for the length of that order, $\sum_{i<B} \|x_{i+1} - x_i\|$. The algorithm keeps a current *anchor*: a point y, its affine map (A_p, c_p), and its safe radius $m(y)$ of Theorem A.7. Before the first point there is no anchor, so the first point always fails and becomes the first anchor. For each point in turn, if it lies within the anchor's radius, we evaluate it by the anchor's map; otherwise it *fails*, and we recompute its pattern, its map and its radius through Theorem 2.3 and make it the new anchor. Making every failing point the anchor, rather than only those in regions not already cached, is what lets the analysis below charge failures to disjoint stretches of the order. Figure 1 draws a run in the plane, with the charges of that analysis.

Theorem A.8 (Certificate-cached cost)**.** *Run certificate-cached evaluation on a batch of B points and let ν be the number of points that fail the certificate. The batch is evaluated in*

$$O(Bd_0 d_L + \nu d_0 T) \tag{12}$$

time. Along every sequence of networks and batches with $\nu d_0 = o(B)$, this is below the pointwise $\Theta(BT)$ whenever $d_0 d_L = o(T)$ along the sequence, and of the same order when $d_0 d_L = \Theta(T)$; without the condition on ν it is never above pointwise by more than a factor d_0.

Proof. First, correctness. A point passing the test lies in the open ball $\mathbb{B}(y, m(y))$ of the current anchor y, so by Theorem A.7 it has the anchor's pattern and the anchor's map returns its exact value; a failing point is evaluated from the map recomputed for it. The count below is therefore the cost of an exact evaluation.

A certified point costs the $O(d_0)$ distance test and $O(d_0 d_L)$ to apply the anchor's map, for $O(Bd_0 d_L)$ over the batch. A failing point y costs one forward pass interleaved with the recursion of Theorem 2.3: that recursion needs $\Delta_{\ell-1} = \operatorname{diag}(p_{\ell-1})$ before it can form M_ℓ, and $p_{\ell-1}$ is read off in $d_{\ell-1}$ comparisons from $h_{\ell-1}(y) = M_{\ell-1} y + v_{\ell-1}$, which the previous step has already produced. Run layer by layer, the forward pass and the recursion together therefore yield the pattern $p = \pi(y)$, the affine map $(A_p, c_p) = (M_L, v_L)$ and the pre-activation values $h_{\ell j}(y)$, at the $O(d_0 T)$ of Theorem 2.3; the row norms $\|g_{\ell j}\|$ that Eq. (11) needs cost a further $\sum_\ell d_\ell d_0 = O(N d_0)$, which the $O(d_0 T)$ absorbs because $N \le T$, as at Eq. (2). There are ν such points, for $O(\nu d_0 T)$ in all. Under $\nu d_0 = o(B)$ the second term is $o(BT)$, and the first is $o(BT)$ or $\Theta(BT)$ according as $d_0 d_L = o(T)$ or $d_0 d_L = \Theta(T)$, which are the two middle claims. Without that condition, $\nu \le B$ bounds the second term by $Bd_0 T$, and $d_L \le d_L d_{L-1} \le T$ bounds the first by the same, so the whole is at most $O(Bd_0 T)$, the pointwise $\Theta(BT)$ times d_0, which is the last claim. □

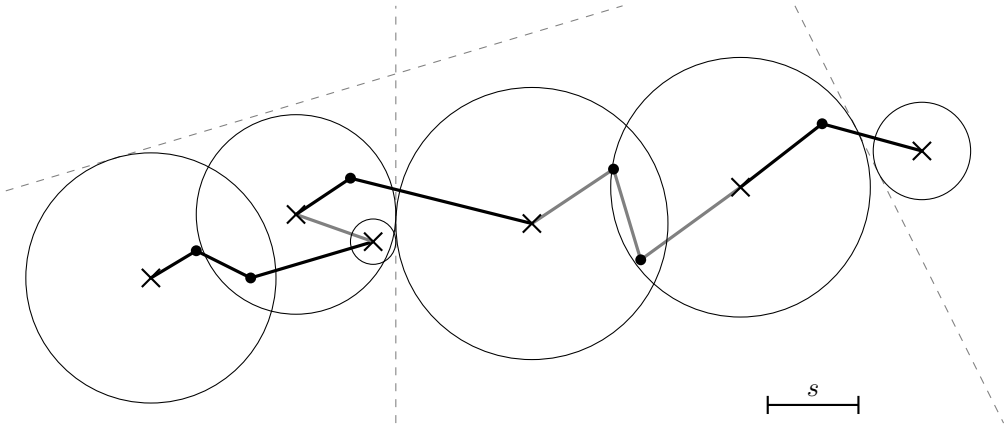


Figure 1: A run of certificate-cached evaluation in the plane, for a network with one hidden layer of three units whose zero sets are the dashed lines. The batch is processed along the polyline, starting at the leftmost cross. Each cross is a failing point that becomes the new anchor, and its circle is the safe ball of Theorem A.7, touching the nearest boundary; the first point fails because there is no anchor yet. Each dot is a certified point, inside the current anchor's ball. The stretches, alternately black and gray, run from each anchor to the next failure, sharing no step. Theorem A.9 charges each failure after the first to its anchor when the anchor's safe radius is below the bar's length s, and otherwise to the stretch leading to it, which then has length at least s. The second anchor lies near the vertical boundary with radius below s, so the failure ending the short gray stretch after it is charged to that anchor; the other four stretches exceed s and pay for their own failures. Only the fourth and sixth crosses, in processing order, lie outside the anchor's region: the certificate fails whenever a point leaves the current ball, so Theorem A.8 counts failures rather than regions.

A.6 Few Failures and the Speedup

The number of failures is controlled by a purely deterministic accounting, which is what makes the randomness enter in one place only.

Lemma A.9 (Failures pay for the order)**.** *Let certificate-cached evaluation process a batch $x_1, \dots, x_B$ in an order of length Λ, let ν be the number of points that fail the certificate, and let m be the safe radius of Theorem A.7. Then for every $s > 0$,*

$$\nu \le 1 + \frac{\Lambda}{s} + |\{i \in [B] : m(x_i) < s\}| \,. \tag{13}$$

Proof. We charge each failure after the first either to its anchor, when the anchor's safe radius is below s, or to the stretch of the order between the anchor and the failure; the two charges give the last two terms of Eq. (13). The first point fails, which is the leading 1. Write $i_1 < \cdots < i_\nu$ for the failure indices. The anchor changes only at a failure, so the failure at x_{i_k} has as its anchor the point $x_{i_{k-1}}$ at which the previous failure occurred, and $k \mapsto i_{k-1}$ is injective on $k \ge 2$: distinct failures have distinct anchors.

Fix $k \ge 2$, and write $x_{i'} = x_{i_k}$ for the failing point and $x_i = x_{i_{k-1}}$ for its anchor. If $m(x_i) < s$ the failure is charged to x_i, one of the points counted by the last term. Otherwise $m(x_i) \ge s$, and since $x_{i'}$ left the anchor's ball, $\|x_{i'} - x_i\| \ge m(x_i) \ge s$; the stretch of the order from x_i to $x_{i'}$ has length at least $\|x_{i'} - x_i\| \ge s$ by the triangle inequality. The stretch charged at k consists of the steps $x_j \to x_{j+1}$ for $i_{k-1} \le j < i_k$; consecutive stretches share the endpoint x_{i_k} but no step, so the charged step sets are pairwise disjoint subsets of the $B-1$ steps of the order. Their lengths therefore sum to at most Λ, and each charged stretch has length at least s, so at most Λ/s failures are charged this way. □

The one thing Theorem A.9 does not control is how many batch points sit close to a boundary, and that is exactly what an anti-concentration condition supplies.

Assumption A.10 (Bounded margin density)**.** There is a constant ρ_{m} such that for every unit (ℓ, j), every batch point x and every $t \geq 0$ the margin $m_{\ell j}(x) = |h_{\ell j}(x)| \,/\, \|g_{\ell j}\|$, with $g_{\ell j}$ the gradient of $h_{\ell j}$ on the region of x as in Theorem A.7 and the margin read as ∞ on the event $g_{\ell j} = 0$, satisfies $\mathbf{Pr}\left[m_{\ell j}(x) < t\right] \leq \rho_{\mathrm{m}} t$, the probability taken over the random network.

The event $g_{\ell j} = 0$, on which the unit's form is constant, is exactly the event Eq. (11) excludes from its minimum, and for $\ell \geq 2$ it has positive probability, occurring whenever all units of layer $\ell - 1$ are off at x; at $\ell = 1$ it is the event $W_1^j = 0$, of probability zero under any continuous law of the weights.

At the first layer, and for Gaussian weights, the constant is explicit. There $g_{1j} = W_1^j$, so writing $W_1^j = \left\|W_1^j\right\| u$ with u uniform on the unit sphere of $\mathbb{R}^{d_0}$ and independent of $\left\|W_1^j\right\|$, as it is for a Gaussian row, the margin is $m_{1j}(x) = \left|\langle u, x\rangle + b_1^j / \left\|W_1^j\right\|\right|$ and the scale cancels outright, with no concentration step. For $d_0 \geq 3$, and conditionally on $\left\|W_1^j\right\|$ and b_1^j, the variable $\langle u, x\rangle$ has density $c_{d_0}\left(1 - t^2 / \|x\|^2\right)^{(d_0-3)/2} / \|x\|$ on $(-\|x\|, \|x\|)$, with $c_{d_0} = \Gamma\left(d_0/2\right) / \left(\sqrt{\pi}\,\Gamma\left((d_0-1)/2\right)\right)$ the reciprocal of the Beta integral $\int_{-1}^{1}\left(1 - s^2\right)^{(d_0-3)/2} ds$. Since $\Gamma(z+1/2)/\Gamma(z) \sim \sqrt{z}$ by Stirling's formula, $c_{d_0} = \Theta\left(\sqrt{d_0}\right)$, so the density is bounded by $O\left(\sqrt{d_0} / \|x\|\right)$ everywhere, the remaining factor being at most 1. The restriction to $d_0 \geq 3$ is real, the exponent making the density unbounded at $d_0 = 2$ and the law atomic at $d_0 = 1$. Hence $\mathbf{Pr}\left[m_{1j}(x) < t \mid \left\|W_1^j\right\|, b_1^j\right] = O\left(\sqrt{d_0}\, t / \|x\|\right)$, and averaging over $\left\|W_1^j\right\|$ and b_1^j gives $\rho_{\mathrm{m}} = O\left(\sqrt{d_0} / \min_i \|x_i\|\right)$, which carries the batch's distance from the origin as well as the dimension.

Beyond the first layer the calculation above, which conditions on the bias and uses only the randomness of the weight row, is not available. Conditioning on the lower layers can make the margin of a deeper unit deterministic. Consider the example $d_0 = d_1 = d_2 = 1$ and $b_2 = 0$. On the event $h_1 > 0$ one has $h_2 = w_2 h_1$ and $g_2 = w_2 w_1$. Hence $m_2 = |h_1| \,/\, |w_1| = m_1$ is fixed by the first layer. Theorem A.10 asks for less than a conditional bound, namely a bound on the law of each margin at each fixed input, and two sources of randomness supply one at every depth. The first is the unit's own bias. The bias enters $h_{\ell j}(x)$ additively and does not enter $g_{\ell j}$ at all. The density of the bias and the norm of the gradient are the two quantities that govern Theorem 3 of Hanin & Rolnick (2019b), and Theorem A.11 applies them at a single input and a single unit. The second, for a network without biases, is the rotation invariance of Gaussian weights, which makes the margin of every unit at every depth, on the event that its form is not constant, distributed as a first-layer margin (Theorem A.14).

Proposition A.11 (Margin density with random biases)**.** *Let all entries of $W_1, \ldots, W_L$ and of $b_1, \ldots, b_L$ be independent. For each $k \in [L]$, let the entries of W_k have a law symmetric about 0 with variance $\theta_k^2 < \infty$. Let the entries of b_k have a law symmetric about 0 with a density bounded by $\beta_k < \infty$. Set $\mu_1^2 = d_0 \theta_1^2$ and $\mu_\ell^2 = \mu_{\ell-1}^2 d_{\ell-1} \theta_\ell^2 / 2$ for $2 \leq \ell \leq L$. Then for every unit (ℓ, j), every $x \in \mathbb{R}^{d_0}$ and every $t \geq 0$,*

$$\mathbb{E}\left[\|g_{\ell j}(x)\|^2\right] = \mu_\ell^2 \qquad \text{and} \qquad \mathbf{Pr}\left[m_{\ell j}(x) < t\right] \leq 2\beta_\ell \mu_\ell t, \tag{14}$$

where $g_{\ell j}(x)$ is the gradient of $h_{\ell j}$ on the region of x and $m_{\ell j}(x)$ is the margin of Theorem A.10. Hence Theorem A.10 holds, for every batch fixed before the network is drawn, with $\rho_{\mathrm{m}} = 2 \max_{\ell \in [L]} \beta_\ell \mu_\ell$. In particular, consider Gaussian weights of variance $\theta_k^2 = 2/d_{k-1}$ and Gaussian biases of variance $s_{\mathrm{b}}^2 > 0$. Then every μ_ℓ^2 equals 2 and every β_ℓ equals $1/\left(s_{\mathrm{b}}\sqrt{2\pi}\right)$. Hence $\rho_{\mathrm{m}} = 2/\left(\sqrt{\pi} s_{\mathrm{b}}\right)$, whatever the depth, the widths, the input dimension and the batch.

Proof. Fix $x \in \mathbb{R}^{d_0}$ and write $p = \pi(x)$. Write $\Delta_k = \operatorname{diag}(p_k)$, and let M_k be the matrices of the recursion of Theorem 2.3 with this pattern. Then $g_{ki}^\top$ is the i-th row of M_k for every unit (k, i). We first record which entries each quantity depends on. For $k \in [L]$, the pre-activation is $h_k(x) = W_k z_{k-1}(x) + b_k$. By induction on k from $z_0 = x$, the pre-activation $h_k(x)$ depends only on the entries of $W_1, \ldots, W_k$ and $b_1, \ldots, b_k$. So does the pattern p_k. Hence $M_1 = W_1$ and $M_k = W_k \Delta_{k-1} M_{k-1}$ depend only on the entries of $W_1, \ldots, W_k$ and $b_1, \ldots, b_{k-1}$. In particular,

g_{ki} does not depend on b_k. Every weight entry of layer k has mean 0, its law being symmetric with finite variance, and so has second moment θ_k^2.

The second moment. We prove the first identity of Eq. (14) by induction on ℓ, for all j at once. For $\ell = 1$ the gradient is $g_{1j} = \left(W_1^j\right)^\top$. So $\mathbb{E}\left[\|g_{1j}\|^2\right] = \sum_{i=1}^{d_0} \mathbb{E}\left[\left(W_1^{ji}\right)^2\right] = d_0\theta_1^2 = \mu_1^2$, where W_1^{ji} is the (j,i) entry of W_1. Let $\ell \geq 2$ and assume $\mathbb{E}\left[\|g_{\ell-1,i}\|^2\right] = \mu_{\ell-1}^2$ for every $i \in [d_{\ell-1}]$. Taking the j-th row of $M_\ell = W_\ell \Delta_{\ell-1} M_{\ell-1}$ gives

$$g_{\ell j} = \sum_{i=1}^{d_{\ell-1}} W_\ell^{ji} p_{\ell-1,i} g_{\ell-1,i}, \qquad \|g_{\ell j}\|^2 = \sum_{i=1}^{d_{\ell-1}} \sum_{i'=1}^{d_{\ell-1}} W_\ell^{ji} W_\ell^{ji'} p_{\ell-1,i} p_{\ell-1,i'} \left\langle g_{\ell-1,i}, g_{\ell-1,i'} \right\rangle .$$

The factor $X_{ii'} = p_{\ell-1,i} p_{\ell-1,i'} \left\langle g_{\ell-1,i}, g_{\ell-1,i'} \right\rangle$ depends only on the entries of layers $1, \ldots, \ell-1$, so it is independent of the row W_ℓ^j, and it is integrable, since by the Cauchy–Schwarz inequality twice $\mathbb{E}\left[|X_{ii'}|\right] \leq \mathbb{E}\left[\|g_{\ell-1,i}\| \, \|g_{\ell-1,i'}\|\right] \leq \mu_{\ell-1}^2$. By independence, $\mathbb{E}\left[W_\ell^{ji} W_\ell^{ji'} X_{ii'}\right] = \mathbb{E}\left[W_\ell^{ji} W_\ell^{ji'}\right] \mathbb{E}\left[X_{ii'}\right]$. Here $\mathbb{E}\left[W_\ell^{ji} W_\ell^{ji'}\right]$ equals $\mathbb{E}\left[W_\ell^{ji}\right] \mathbb{E}\left[W_\ell^{ji'}\right] = 0$ for $i \neq i'$, and equals θ_ℓ^2 for $i = i'$. Since $p_{\ell-1,i}^2 = p_{\ell-1,i}$,

$$\mathbb{E}\left[\|g_{\ell j}\|^2\right] = \theta_\ell^2 \sum_{i=1}^{d_{\ell-1}} \mathbb{E}\left[p_{\ell-1,i} \|g_{\ell-1,i}\|^2\right].$$

Fix i and write $h = h_{\ell-1,i}(x)$ and $Y = \|g_{\ell-1,i}\|^2$, so that $p_{\ell-1,i} = \mathbb{1}(h > 0)$ by Theorem 2.2. Let Φ be the map on the entries of the network that replaces the row $W_{\ell-1}^i$ by $-W_{\ell-1}^i$ and the entry $b_{\ell-1}^i$ by $-b_{\ell-1}^i$ and leaves every other entry unchanged. The entries are independent and each has a law symmetric about 0, so Φ preserves their joint law. The vector $z_{\ell-2}(x)$ and the matrices $\Delta_{\ell-2}$ and $M_{\ell-2}$ depend only on layers $1, \ldots, \ell-2$. Read $z_0 = x$ when $\ell = 2$. The map Φ does not touch these layers. Hence Φ sends $h = W_{\ell-1}^i z_{\ell-2}(x) + b_{\ell-1}^i$ to $-h$. Also Φ sends $g_{\ell-1,i}$ to $-g_{\ell-1,i}$. Indeed $g_{\ell-1,i}^\top = W_{\ell-1}^i \Delta_{\ell-2} M_{\ell-2}$ for $\ell \geq 3$, and $g_{1i}^\top = W_1^i$ for $\ell = 2$. So Φ leaves Y unchanged. Therefore $\mathbb{E}\left[\mathbb{1}(h > 0) Y\right] = \mathbb{E}\left[\mathbb{1}(-h > 0) Y\right] = \mathbb{E}\left[\mathbb{1}(h < 0) Y\right]$. Moreover $\mathbf{Pr}\left[h = 0\right] = 0$. Indeed, condition on all entries other than $b_{\ell-1}^i$, which are independent of it; then $h = h' + b_{\ell-1}^i$ with h' fixed, and a variable with a density takes the value $-h'$ with probability 0. So $\mathbb{E}\left[Y\right] = \mathbb{E}\left[\mathbb{1}(h > 0) Y\right] + \mathbb{E}\left[\mathbb{1}(h < 0) Y\right] = 2\mathbb{E}\left[\mathbb{1}(h > 0) Y\right]$, that is $\mathbb{E}\left[p_{\ell-1,i} \|g_{\ell-1,i}\|^2\right] = \mu_{\ell-1}^2/2$ by the induction hypothesis. Summing over the $d_{\ell-1}$ values of i gives $\mathbb{E}\left[\|g_{\ell j}\|^2\right] = \theta_\ell^2 d_{\ell-1} \mu_{\ell-1}^2/2 = \mu_\ell^2$, which completes the induction.

The small-ball bound. Fix (ℓ, j) and $t \geq 0$. Let $\mathcal{O}$ be the collection of all entries of the network other than b_ℓ^j. Write $G = \|g_{\ell j}\|$ and $a = W_\ell^j z_{\ell-1}(x)$, so that $h_{\ell j}(x) = a + b_\ell^j$. By the first paragraph, G and a are functions of $\mathcal{O}$, and b_ℓ^j is independent of $\mathcal{O}$. If $G = 0$, the margin is ∞ by the convention of Theorem A.10, and the event $m_{\ell j}(x) < t$ is empty. If $G > 0$, the event $m_{\ell j}(x) < t$ is the event $\left|a + b_\ell^j\right| < tG$. That is, $b_\ell^j \in (-a - tG, -a + tG)$, an interval of length $2tG$ that is fixed once $\mathcal{O}$ is. The density of b_ℓ^j is at most β_ℓ, so conditionally on $\mathcal{O}$ this event has probability at most $2\beta_\ell tG$. In both cases $\mathbf{Pr}\left[m_{\ell j}(x) < t \mid \mathcal{O}\right] \leq 2\beta_\ell tG$, and taking expectations,

$$\mathbf{Pr}\left[m_{\ell j}(x) < t\right] \leq 2\beta_\ell t \mathbb{E}\left[G\right] \leq 2\beta_\ell t \left(\mathbb{E}\left[G^2\right]\right)^{1/2} = 2\beta_\ell \mu_\ell t,$$

the middle step by the Cauchy–Schwarz inequality and the last by the first identity of Eq. (14).

The constant. Theorem A.10 concerns every unit (ℓ, j) and every point x of a batch fixed before the draw, and for each of them the bound just proved gives $\mathbf{Pr}\left[m_{\ell j}(x) < t\right] \leq 2\beta_\ell \mu_\ell t \leq \rho_{\mathrm{m}} t$. For the Gaussian case, $\mu_1^2 = d_0 \cdot 2/d_0 = 2$, and if $\mu_{\ell-1}^2 = 2$ then $\mu_\ell^2 = 2 \cdot d_{\ell-1} \cdot (2/d_{\ell-1})/2 = 2$. The Gaussian density of variance s_{b}^2 is at most its value $1/\left(s_{\mathrm{b}}\sqrt{2\pi}\right)$ at 0. Hence $2 \cdot \sqrt{2}/\left(s_{\mathrm{b}}\sqrt{2\pi}\right) = 2/\left(\sqrt{\pi} s_{\mathrm{b}}\right)$. □

The gradient estimate just proved also bounds every zero crossing on a fixed segment, which supplies the hypothesis used by Theorem A.3.

Proposition A.12 (Crossing density with random biases)**.** *Under the hypotheses of Theorem A.11, and with $L \geq 2$ so that $N > 0$, Theorem A.2 holds with*

$$c_{\mathrm{d}} = \frac{1}{N} \sum_{\ell=1}^{L-1} d_\ell \beta_\ell \mu_\ell \leq \max_{\ell < L} \beta_\ell \mu_\ell.$$

For the Gaussian initialization in that proposition, $c_{\mathrm{d}} = 1/(s_{\mathrm{b}}\sqrt{\pi})$, independent of the depth and widths.

Proof. Fix a segment $x(t) = x_0 + tv$ for $0 \leq t \leq 1$, of length $\rho = \|v\|$, and a hidden unit (ℓ, j). Condition on all network entries except its own bias b_ℓ^j. Along the segment its pre-activation is $a(t) + b_\ell^j$, where $a(t) = W_\ell^j z_{\ell-1}(x(t))$ is a continuous piecewise-affine function independent of that bias. Partition $[0, 1]$ into its finitely many affine pieces. On a piece with endpoint values a_0, a_1, a zero in its interior requires the bias to lie between $-a_0$ and $-a_1$, an interval of length $|a_1 - a_0|$. Thus the conditional expected number of zero crossings, counted across the pieces, is at most $\beta_\ell \sum |a_1 - a_0| = \beta_\ell \int_0^1 |a'(t)|\, dt$. The density of the bias makes roots at the finitely many piece endpoints and identically zero pieces null events. Almost everywhere, $a'(t) = \langle g_{\ell j}(x(t)), v \rangle$; the unit's own bias does not affect this gradient. Taking expectations, applying Cauchy–Schwarz and then Eq. (14) at each fixed $x(t)$, gives

$$\mathbb{E}[\text{crossings of } (\ell, j)] \leq \beta_\ell \int_0^1 \|v\|\, \mathbb{E}\, \|g_{\ell j}(x(t))\|\, dt \leq \beta_\ell \mu_\ell \rho.$$

Summing over the d_ℓ units of each hidden layer proves the first claim. At the stated Gaussian initialization $\mu_\ell = \sqrt{2}$ and $\beta_\ell = 1/(s_{\mathrm{b}}\sqrt{2\pi})$, so their product is $1/(s_{\mathrm{b}}\sqrt{\pi})$ at every layer. □

The bias is what Theorem A.11 spends: the bound grows like $1/s_{\mathrm{b}}$ as the biases shrink, and at $s_{\mathrm{b}} = 0$ it says nothing, which is the situation of the example above. For Gaussian weights and no biases at all, a different argument gives a bound at every depth that does not depend on the depth.

Lemma A.13 (A cap of the sphere)**.** *Let ξ be a standard Gaussian vector in $\mathbb{R}^{d_0}$ and e a unit vector. Then $\mathbf{Pr}\left[|\langle \xi, e \rangle| < \tau \|\xi\|\right] \leq 2\sqrt{d_0}\tau$ for every $\tau \geq 0$.*

Proof. Let Q be an orthogonal matrix with Qe the first standard basis vector. Then $\langle \xi, e \rangle = \langle Q\xi, Qe \rangle = (Q\xi)_1$ and $\|\xi\| = \|Q\xi\|$, and $Q\xi$ is again a standard Gaussian vector, so we may take e to be the first basis vector and $\langle \xi, e \rangle = \xi_1$. If $\tau \geq 1/2$, the probability is at most $1 \leq 2\tau \leq 2\sqrt{d_0}\tau$. Let $0 \leq \tau < 1/2$. If $d_0 = 1$, then $\|\xi\| = |\xi_1|$ and the event $|\xi_1| < \tau\, |\xi_1|$ is empty. If $d_0 \geq 2$, write $\varrho^2 = \sum_{i=2}^{d_0} \xi_i^2$, which is independent of ξ_1. The event $|\xi_1| < \tau \|\xi\|$ is the event $\xi_1^2 < \tau^2 \left(\xi_1^2 + \varrho^2\right)$, that is $\xi_1^2 \left(1 - \tau^2\right) < \tau^2 \varrho^2$, that is $|\xi_1| < \tau\varrho/\sqrt{1 - \tau^2}$. Moreover $\sqrt{1 - \tau^2} > \sqrt{3}/2$, because $\tau < 1/2$. So the event lies in $|\xi_1| < 2\tau\varrho/\sqrt{3}$. Conditionally on ϱ, the variable ξ_1 is standard normal with density at most $1/\sqrt{2\pi}$, so the conditional probability of that event is at most $2 \cdot 2\tau\varrho/\left(\sqrt{3}\sqrt{2\pi}\right) = 4\tau\varrho/\sqrt{6\pi}$. Taking expectations and using $\mathbb{E}\left[\varrho\right] \leq \left(\mathbb{E}\left[\varrho^2\right]\right)^{1/2} = \sqrt{d_0 - 1}$ by the Cauchy–Schwarz inequality, the probability is at most $4\tau\sqrt{d_0 - 1}/\sqrt{6\pi}$, which is below $\sqrt{d_0}\tau$ because $\sqrt{6\pi} > 4$. □

Theorem A.11 needed the biases to be random, since it bounds the margin through their density. The next result instead covers a Gaussian network with no biases at all, at every depth. It reduces the margin of every unit to that of a first-layer unit and bounds the latter with the cap bound of Theorem A.13, just proved, in part (iii) below.

Theorem A.14 (Margins of a bias-free Gaussian network)**.** *Let $b_k = 0$ for every $k \in [L]$. Let $W_1, \ldots, W_L$ be independent, the entries of W_k independent normal with mean 0 and variance $\theta_k^2 > 0$. Fix $x \in \mathbb{R}^{d_0}$ with $x \neq 0$ and a unit (ℓ, j). Then:*

- *(i) there are a random variable $\lambda \geq 0$ and a standard Gaussian vector ξ in $\mathbb{R}^{d_0}$, independent of λ, such that $g_{\ell j}(x)$ has the law of $\lambda\xi$;*
- *(ii) $h_{\ell j}(x) = \langle g_{\ell j}(x), x \rangle$;*

(iii) for every $t \geq 0$, $\mathbf{Pr}\left[m_{\ell j}(x) < t\right] = \mathbf{Pr}\left[g_{\ell j}(x) \neq 0\right] \mathbf{Pr}\left[m_{11}(x) < t\right] \leq 2\sqrt{d_0}t/\left\|x\right\|$.

Hence Theorem A.10 holds, for every batch of nonzero points fixed before the network is drawn, with $\rho_{\mathrm{m}} = 2\sqrt{d_0}/\min_i \left\|x_i\right\|$.

Proof. Three facts about a network of this form hold for every input $u \neq 0$ and every unit (k, i). The proof of the first claim uses them for networks other than the given one, so they are stated for any network with zero biases. Write $g_{ki}(u)$ for the gradient of h_{ki} on the region of u and $M_k(u)$ for the matrices of Theorem 2.3 with the pattern $\pi(u)$.

(F1) Linearity. In the recursion of Theorem 2.3, $v_1 = b_1 = 0$. Also $v_{k-1} = 0$ gives $v_k = W_k\Delta_{k-1}v_{k-1} + b_k = 0$, so every v_k vanishes and $h_k(u) = M_k(u)u$. Taking the i-th entry, $h_{ki}(u) = \langle g_{ki}(u), u\rangle$. With $k = \ell$, $i = j$ and $u = x$ this is the second claim.

(F2) Scaling. For $\vartheta > 0$, one has $h_1(\vartheta u) = \vartheta W_1 u = \vartheta h_1(u)$. Suppose $h_{k-1}(\vartheta u) = \vartheta h_{k-1}(u)$. Since $\sigma(\vartheta s) = \vartheta\sigma(s)$ for $\vartheta > 0$, this gives $z_{k-1}(\vartheta u) = \sigma\left(\vartheta h_{k-1}(u)\right) = \vartheta z_{k-1}(u)$, and so $h_k(\vartheta u) = \vartheta h_k(u)$. Positive scaling does not change the sign of any pre-activation, so $\pi(\vartheta u) = \pi(u)$, the matrices M_k agree at ϑu and u, and $g_{ki}(\vartheta u) = g_{ki}(u)$.

(F3) Rotation. Write $g_{ki}\left[V\right](u)$ for the gradient computed with the first-layer matrix W_1 replaced by V and every other matrix unchanged. For an orthogonal matrix Q, consider the first-layer pre-activation with W_1Q at u. It equals W_1Qu, which is also the first-layer pre-activation with W_1 at Qu. Every later pre-activation is a function of the first-layer one and of $W_2, \ldots, W_L$, so the two patterns agree. By the recursion, $M_k\left[W_1Q\right](u) = M_k\left[W_1\right](Qu)Q$. Taking rows, $g_{ki}\left[W_1Q\right](u) = Q^\top g_{ki}\left[W_1\right](Qu)$.

A consequence of the first claim. Suppose the first claim holds for the unit (k, i) of some network of this form at every nonzero input. Then the law of $g_{ki}(u)$ is invariant under every orthogonal map and is the same for every $u \neq 0$. Indeed, the law of $\lambda\xi$ is invariant under every orthogonal Q, because $Q\xi$ is again a standard Gaussian vector independent of λ. The rows of W_1Q are independent with law $N\left(0, \theta_1^2Q^\top Q\right) = N\left(0, \theta_1^2 I\right)$. So W_1Q has the law of W_1 and is independent of $W_2, \ldots, W_L$. By (F3), $g_{ki}(Qu) = Qg_{ki}\left[W_1Q\right](u)$. Since W_1Q has the law of W_1, this has the law of $Qg_{ki}(u)$, which is the law of $g_{ki}(u)$. Given $u, u' \neq 0$, take an orthogonal Q with $Q\left(u/\left\|u\right\|\right) = u'/\left\|u'\right\|$. Then $u' = \vartheta Qu$ with $\vartheta = \left\|u'\right\|/\left\|u\right\| > 0$. By (F2), $g_{ki}(u') = g_{ki}(Qu)$, which has the law of $g_{ki}(u)$.

The first claim. We prove it by induction on ℓ, simultaneously for all depths, widths, variances, units of layer ℓ and nonzero inputs. For $\ell = 1$ the gradient $g_{1j} = \left(W_1^j\right)^\top$ is normal with mean 0 and covariance $\theta_1^2 I$. This is the law of $\lambda\xi$ with $\lambda = \theta_1$.

Let $\ell \geq 2$ and assume the claim for every unit of layer $\ell - 1$ of every network of this form. Put $\hat{x} = x/\left\|x\right\|$. If $d_0 \geq 2$, let U be a $d_0 \times (d_0 - 1)$ matrix whose columns are an orthonormal basis of the orthogonal complement of $\hat{x}$. If $d_0 = 1$, the terms with U below are absent. Set $a = W_1\hat{x} \in \mathbb{R}^{d_1}$ and $Y = W_1U \in \mathbb{R}^{d_1 \times (d_0 - 1)}$. The matrix $\left[\hat{x} \quad U\right]$ is orthogonal. Hence, by the rotation invariance just used, the matrix $W_1\left[\hat{x} \quad U\right] = \left[a \quad Y\right]$ has independent entries, normal with mean 0 and variance θ_1^2. Hence the vector a and the matrix Y are independent of each other and of $W_2, \ldots, W_L$. Multiplying W_1 on the right by $\left[\hat{x} \quad U\right]\left[\hat{x} \quad U\right]^\top = I$ gives $W_1 = a\hat{x}^\top + YU^\top$.

The first-layer pre-activation is $h_1(x) = \left\|x\right\| a$, so the first-layer pattern is $p_1 = \left(\mathbb{1}\left(a_i > 0\right)\right)_{i \in [d_1]}$ and $z_1(x) = \left\|x\right\| \sigma(a)$. Every later pre-activation and pattern is a function of $z_1(x)$ and $W_2, \ldots, W_L$, hence of a and $W_2, \ldots, W_L$. Define the row vector $c^\top = W_\ell^j\Delta_{\ell-1}W_{\ell-1}\cdots\Delta_2W_2\Delta_1 \in \mathbb{R}^{1 \times d_1}$, which for $\ell = 2$ reads $c^\top = W_2^j\Delta_1$. It is a function of a and $W_2, \ldots, W_\ell$, hence independent of Y. By the recursion of Theorem 2.3, $g_{\ell j}^\top$ is the j-th row of $M_\ell = W_\ell\Delta_{\ell-1}\cdots W_2\Delta_1W_1$, that is $g_{\ell j}^\top = c^\top W_1$, and so

$$g_{\ell j} = W_1^\top c = \left(a^\top c\right)\hat{x} + U\left(Y^\top c\right). \tag{15}$$

Claim 1. Conditionally on a and $W_2, \ldots, W_\ell$, the vector $Y^\top c$ is normal with mean 0 and covariance $\theta_1^2\left\|c\right\|^2 I_{d_0 - 1}$. Indeed c is then fixed, and $Y^\top c = \sum_{i=1}^{d_1} c_i\left(Y^i\right)^\top$. The rows Y^i are independent of

the conditioning and of one another, each normal with covariance $\theta_1^2 I_{d_0-1}$. A sum of independent centered normal vectors is normal with the sum of the covariances, here $\sum_i c_i^2 \theta_1^2 I_{d_0-1}$.

Claim 2. There is a standard normal variable ω, independent of $\|c\|$, such that the pair $\left(a^\top c, \|c\|\right)$ has the law of $(\theta_1 \|c\| \omega, \|c\|)$. Let $S = \{i \in [d_1] : a_i > 0\}$ be the set of units of the first layer active at x. Condition on $S = S_0$ for a fixed $S_0 \subseteq [d_1]$ of positive probability. It suffices to prove the claim conditionally on each such event, with ω standard normal and independent of $\|c\|$ under each. This suffices because the law of ω does not depend on S_0, so mixing over S_0 then gives the claim with ω independent of $(S, \|c\|)$. If $S_0 = \emptyset$, then $\Delta_1 = 0$, so $c = 0$ and $a^\top c = 0$. The pair is then $(0, 0)$, which is $(\theta_1 \cdot 0 \cdot \omega, 0)$ for every ω. Let $S_0 \neq \emptyset$ and $k = |S_0|$. The entries of a are independent, so conditionally on $S = S_0$ the entries $a_{S_0} = (a_i)_{i \in S_0}$ are independent, each with the law of a normal variable of variance θ_1^2 conditioned to be positive, and they remain independent of $W_2, \ldots, W_L$. On this event, $\sigma(a)$ and Δ_1 are determined by a_{S_0}. The entries of $\sigma(a)$ outside S_0 are 0, and those in S_0 are a_{S_0}.

Let f_{S_0} be the network with input dimension k and zero biases, whose first-layer matrix is $W_1' = W_2 [\cdot, S_0]$, the columns of W_2 indexed by S_0. Its later matrices are $W_{k'}' = W_{k'+1}$ for $2 \leq k' \leq L-1$. It is a network of the form of the theorem, with variances $\theta_2^2, \ldots, \theta_L^2$, and its weights are independent of a. Write h', Δ' and g' for its pre-activations, masks and gradients at the input a_{S_0}. Since $z_1(x) = \|x\| \sigma(a)$ and $\sigma(a)$ vanishes outside S_0, the base case is $h_2(x) = W_2 z_1(x) = \|x\| W_1' a_{S_0} = \|x\| h_1'(a_{S_0})$. For the inductive step, if $h_{k'+1}(x) = \|x\| h_{k'}'(a_{S_0})$ then $z_{k'+1}(x) = \|x\| z_{k'}'(a_{S_0})$ and $h_{k'+2}(x) = \|x\| h_{k'+1}'(a_{S_0})$. So $h_{k'+1}(x) = \|x\| h_{k'}'(a_{S_0})$ for every $k' \in [L-1]$, the patterns agree, and $\Delta_{k'+1} = \Delta_{k'}'$. The entries of c outside S_0 vanish, because Δ_1 is the last factor of $c^\top$, and the entries in S_0 are

$$c_{S_0}^\top = W_\ell^j \Delta_{\ell-1} \cdots \Delta_2 W_2 [\cdot, S_0] = W_{\ell-1}'^j \Delta_{\ell-2}' \cdots \Delta_1' W_1' = g_{\ell-1,j}' \left(a_{S_0}\right)^\top,$$

the gradient of the unit $(\ell - 1, j)$ of f_{S_0} at a_{S_0} (for $\ell = 2$ the product reduces to $W_1'^j$). Write $g' = g_{\ell-1,j}' (a_{S_0})$. Then $\|c\| = \|g'\|$ and $a^\top c = a_{S_0}^\top c_{S_0} = \langle g', a_{S_0} \rangle$.

By the induction hypothesis, the first claim holds for the unit $(\ell-1, j)$ of f_{S_0} at every nonzero input. Hence, by the consequence proved above, the law of $g_{\ell-1,j}'(u)$ is the same for every $u \neq 0$ and is invariant under every orthogonal map of $\mathbb{R}^k$. The input a_{S_0} is nonzero, since its entries are positive, and it is independent of the weights of f_{S_0}; so conditionally on a_{S_0} the law of g' is that common law, whatever the value of a_{S_0}. Hence g' is independent of a_{S_0}, and its law is invariant under every orthogonal map of $\mathbb{R}^k$. Let D be a diagonal matrix whose diagonal entries are independent uniform signs, independent of everything else. For each fixed sign matrix D_0, the pair $(D_0 g', a_{S_0})$ has the law of (g', a_{S_0}). This holds because g' and a_{S_0} are independent and $D_0 g'$ has the law of g'. So the pair (Dg', a_{S_0}) has the law of (g', a_{S_0}) as well. Since D is orthogonal and symmetric, $\langle Dg', a_{S_0} \rangle = \langle g', D a_{S_0} \rangle$ and $\|Dg'\| = \|g'\|$, so

$$(\langle g', a_{S_0} \rangle, \|g'\|) \text{ has the law of } (\langle Dg', a_{S_0} \rangle, \|Dg'\|) = (\langle g', D a_{S_0} \rangle, \|g'\|).$$

The entries of $D a_{S_0}$ are independent, each a positive half of a normal variable of variance θ_1^2 multiplied by an independent uniform sign, which is a normal variable of mean 0 and variance θ_1^2. Moreover $D a_{S_0}$ is independent of g'. Conditionally on g', the variable $\langle g', D a_{S_0} \rangle$ is therefore normal with mean 0 and variance $\theta_1^2 \|g'\|^2$. So it has the law of $\theta_1 \|g'\| \omega$ jointly with $\|g'\|$, for a standard normal ω independent of g'. This is Claim 2 on the event $S = S_0$, since $\|c\| = \|g'\|$ and $a^\top c = \langle g', a_{S_0} \rangle$.

Conclusion of the induction step. Let η be a standard Gaussian vector in $\mathbb{R}^{d_0-1}$, independent of everything else. For a bounded measurable function ϕ on $\mathbb{R}^{d_0}$, put $\Psi(s, r) = \mathbb{E}\left[\phi\left(s\hat{x} + \theta_1 r U \eta\right)\right]$ for real s and $r \geq 0$. By Eq. (15) and Claim 1, $\mathbb{E}\left[\phi\left(g_{\ell j}\right) \mid a, W_2, \ldots, W_\ell\right] = \Psi\left(a^\top c, \|c\|\right)$, because $U\left(Y^\top c\right)$ has, given the conditioning, the law of $\theta_1 \|c\| U\eta$ and $a^\top c$ is fixed. Taking expectations and applying Claim 2,

$$\mathbb{E}\left[\phi\left(g_{\ell j}\right)\right] = \mathbb{E}\left[\Psi\left(a^\top c, \|c\|\right)\right] = \mathbb{E}\left[\Psi\left(\theta_1 \|c\| \omega, \|c\|\right)\right] = \mathbb{E}\left[\phi\left(\theta_1 \|c\| \left(\omega\hat{x} + U\eta\right)\right)\right],$$

with ω, η and $\|c\|$ independent. The vector $\xi = \omega\hat{x} + U\eta = \begin{bmatrix}\hat{x} & U\end{bmatrix}(\omega, \eta)$ is the image of the standard Gaussian vector (ω, η) of $\mathbb{R}^{d_0}$ under an orthogonal matrix, so it is a standard Gaussian vector, independent of $\|c\|$. Hence $g_{\ell j}$ has the law of $\lambda\xi$ with $\lambda = \theta_1 \|c\|$, which is the first claim at ℓ.

The third claim. Define $\Upsilon(v) = |\langle v, x\rangle| / \|v\|$ for $v \neq 0$ and $\Upsilon(0) = \infty$. By the second claim and the definition of the margin, $m_{\ell j}(x) = \Upsilon\left(g_{\ell j}(x)\right)$, so by the first claim $m_{\ell j}(x)$ has the law of $\Upsilon(\lambda\xi)$. The vector ξ is nonzero with probability one; if $\lambda = 0$ then $\Upsilon(\lambda\xi) = \infty$, and if $\lambda > 0$ then $\Upsilon(\lambda\xi) = \Upsilon(\xi)$. Hence, by the independence of λ and ξ,

$$\mathbf{Pr}\left[m_{\ell j}(x) < t\right] = \mathbf{Pr}\left[\lambda > 0\right] \mathbf{Pr}\left[\Upsilon(\xi) < t\right],$$

and $\mathbf{Pr}\left[\lambda > 0\right] = \mathbf{Pr}\left[\lambda\xi \neq 0\right] = \mathbf{Pr}\left[g_{\ell j}(x) \neq 0\right]$. At $\ell = 1$ the variable $\lambda = \theta_1$ is positive, so $\mathbf{Pr}\left[\Upsilon(\xi) < t\right] = \mathbf{Pr}\left[m_{11}(x) < t\right]$, which is the equality of the third claim. Finally $\Upsilon(\xi) < t$ is the event $|\langle \xi, \hat{x}\rangle| < (t/\|x\|)\|\xi\|$, whose probability is at most $2\sqrt{d_0}t/\|x\|$ by Theorem A.13 with $e = \hat{x}$ and $\tau = t/\|x\|$.

For the last sentence, apply the third claim to every hidden unit and every point x_i of the batch, and bound $1/\|x_i\|$ by $1/\min_i \|x_i\|$. □

Theorem A.14 reads the example above correctly. In a bias-free Gaussian network, conditioning on the lower layers can fix a deep margin. But its unconditional law is exactly that of a first-layer margin on the event that the unit's form is not constant, the rest of its mass sitting at ∞. The shape of $\Delta_{\ell-1}M_{\ell-1}$, which a conditional calculation sees, does not enter. The two results also combine when the biases sit in the first layer only.

Corollary A.15 (Gaussian biases in the first layer)**.** *Let the weights be as in Theorem A.14, let the entries of b_1 be independent normal with mean 0 and variance $s_{\mathrm{b}}^2 > 0$, independent of the weights, and let $b_k = 0$ for $k \geq 2$. Then for every unit (ℓ, j), every $x \in \mathbb{R}^{d_0}$ and every $t \geq 0$,* $\mathbf{Pr}\left[m_{\ell j}(x) < t\right] \leq 2\sqrt{d_0 + 1}t / \left(\|x\|^2 + s_{\mathrm{b}}^2/\theta_1^2\right)^{1/2}$.

Proof. Put $r_{\mathrm{b}} = s_{\mathrm{b}}/\theta_1 > 0$, $\tilde{x} = (x, r_{\mathrm{b}}) \in \mathbb{R}^{d_0+1}$, and $\tilde{W}_1 = \begin{bmatrix} W_1 & b_1/r_{\mathrm{b}} \end{bmatrix} \in \mathbb{R}^{d_1 \times (d_0+1)}$. The entries of $\tilde{W}_1$ are independent normal with mean 0 and variance θ_1^2, since those of b_1/r_{b} have variance $s_{\mathrm{b}}^2/r_{\mathrm{b}}^2 = \theta_1^2$. Also $\tilde{W}_1\tilde{x} = W_1 x + b_1 = h_1(x)$. So the network $\tilde{f}$, with first-layer matrix $\tilde{W}_1$, the later matrices $W_2, \ldots, W_L$, and zero biases, satisfies the hypotheses of Theorem A.14 in input dimension $d_0 + 1$. At $\tilde{x}$ it has the same pre-activations and the same pattern as the given network at x. Its matrices are $\tilde{M}_1 = \tilde{W}_1$ and $\tilde{M}_k = W_k\Delta_{k-1}\tilde{M}_{k-1}$, whose first d_0 columns are M_k. So its gradient $\tilde{g}_{\ell j}$ at $\tilde{x}$ has $g_{\ell j}(x)$ as its first d_0 entries. Hence $\|g_{\ell j}(x)\| \leq \|\tilde{g}_{\ell j}\|$. If $g_{\ell j}(x) \neq 0$, then $m_{\ell j}(x) = |h_{\ell j}(x)| / \|g_{\ell j}(x)\| \geq |h_{\ell j}(x)| / \|\tilde{g}_{\ell j}\| = \tilde{m}_{\ell j}(\tilde{x})$, the margin of $\tilde{f}$; if $g_{\ell j}(x) = 0$, then $m_{\ell j}(x) = \infty$. In both cases the event $m_{\ell j}(x) < t$ lies in the event $\tilde{m}_{\ell j}(\tilde{x}) < t$, whose probability is at most $2\sqrt{d_0 + 1}t/\|\tilde{x}\|$ by the third claim of Theorem A.14, since $\tilde{x} \neq 0$; and $\|\tilde{x}\|^2 = \|x\|^2 + s_{\mathrm{b}}^2/\theta_1^2$. □

At $\theta_1^2 = 2/d_0$ the bound of Theorem A.15 is at most $2\sqrt{2(d_0+1)/d_0}t/s_{\mathrm{b}} \leq 4t/s_{\mathrm{b}}$ at every input, the origin included. It is also at most $2\sqrt{d_0+1}t/\|x\|$. So for a batch of nonzero points, Theorem A.10 holds with $\rho_{\mathrm{m}} = 2\sqrt{d_0+1}/\min_i \|x_i\|$ for every $s_{\mathrm{b}} > 0$. This is a constant that does not grow as $s_{\mathrm{b}} \to 0$, unlike that of Theorem A.11. What these results leave open is a bound uniform in the scale of the biases when the biases of the deeper layers are nonzero. Also open is any bound for deterministic nonzero biases, or for weights whose law is not rotation invariant and biases that are not random. The constant enters the threshold of Theorem A.17 linearly: at the Gaussian initialization of Theorem A.11 that threshold reads $B = \Omega\left(L^3 d^3 \Lambda/s_{\mathrm{b}} + Ld\right)$, and for a bias-free Gaussian network $B = \Omega\left(L^3 d^3 \sqrt{d_0}\Lambda/\min_i \|x_i\| + Ld\right)$.

Proposition A.16 (Failures are square-root in the order length)**.** *Under Theorem A.10, the expected number of certificate failures of certificate-cached evaluation on a batch of B points, processed in an order of length Λ through a network with N hidden units, satisfies*

$$\mathbb{E}\left[\nu\right] \leq 1 + 2\sqrt{BN\rho_{\mathrm{m}}\Lambda}. \tag{16}$$

Proof. Fix $s > 0$. For a single batch point x the margin $m(x)$ of Eq. (11) is a minimum of the $m_{\ell j}(x)$, so Theorem A.10 and a union bound over the N units give $\mathbf{Pr}\left[m(x) < s\right] \leq N\rho_{\mathrm{m}}s$, and summing over the batch, $\mathbb{E}\left[|\{i : m(x_i) < s\}|\right] \leq BN\rho_{\mathrm{m}}s$. Taking expectations in Eq. (13), $\mathbb{E}\left[\nu\right] \leq 1 + \Lambda/s + BN\rho_{\mathrm{m}}s$ for every $s > 0$; choosing $s = \sqrt{\Lambda/(BN\rho_{\mathrm{m}})}$ gives Eq. (16) when

$\Lambda > 0$ and $BN\rho_{\mathrm{m}} > 0$. If $\Lambda = 0$ that choice degenerates, and the bound follows from Eq. (13) instead: it holds for every $s > 0$, so letting $s \to 0$ in $\mathbb{E}[\nu] \le 1 + BN\rho_{\mathrm{m}} s$ gives $\mathbb{E}[\nu] \le 1$. If $BN\rho_{\mathrm{m}} = 0$, the general inequality $\mathbb{E}[\nu] \le 1 + \Lambda/s + BN\rho_{\mathrm{m}} s$ reduces to $\mathbb{E}[\nu] \le 1 + \Lambda/s$ for every $s > 0$, so letting $s \to \infty$ gives $\mathbb{E}[\nu] \le 1$ as well. □

The randomness enters only through Theorem A.10, which is applied to every batch point and never to the anchor, whose identity is determined by the very failures being counted. That separation is what Theorem A.9 provides.

Corollary A.17 (Average-case speedup without point location)**.** *Let the network have $L \ge 2$ layers and hidden widths $d_\ell = d$ for $1 \le \ell \le L-1$, with $d_0 = O(d)$ and $d_L = d$, so that $T = \Theta\left(Ld^2\right)$ and $N = \Theta(Ld)$, and assume Theorem A.10. For a batch of B points processed in an order of length Λ, with $B = \Omega\left(L^3 d^3 \rho_{\mathrm{m}} \Lambda + Ld\right)$, certificate-cached evaluation runs in expected time $O\left(Bd_0 d\right)$, a factor of $\Theta\left(Ld/d_0\right)$ below the pointwise $\Theta\left(BLd^2\right)$. Every operation is polynomial in the dimension, and the failing points, the only ones located, each by one forward pass, number $O\left(B/(Ld)\right)$ in expectation.*

Proof. Eq. (12) is affine in ν, so the expectation passes through it and Eq. (16) may be substituted directly. By Theorem A.8 and Eq. (16) the expected cost is

$$O\left(Bd_0 d + \left(1 + \sqrt{BN\rho_{\mathrm{m}}\Lambda}\right) d_0 T\right) = O\left(Bd_0 d + d_0 Ld^2 + d_0 Ld^2 \sqrt{BLd\rho_{\mathrm{m}}\Lambda}\right), \tag{17}$$

using $N = \Theta(Ld)$ and $T = \Theta\left(Ld^2\right)$. The middle term is at most the first once $B = \Omega(Ld)$. The last is at most the first exactly when $Ld^2\sqrt{BLd\rho_{\mathrm{m}}\Lambda} = O\left(Bd\right)$. Dividing by Ld^2 and squaring turns this into $BLd\rho_{\mathrm{m}}\Lambda = O\left(B^2/\left(L^2 d^2\right)\right)$, and rearranging gives $B = \Omega\left(L^3 d^3 \rho_{\mathrm{m}} \Lambda\right)$. Under the same condition, $\rho_{\mathrm{m}}\Lambda = O\left(B/\left(L^3 d^3\right)\right)$, so Eq. (16) gives $\mathbb{E}[\nu] \le 1 + 2\sqrt{BN\rho_{\mathrm{m}}\Lambda} = 1 + O\left(B/(Ld)\right) = O\left(B/(Ld)\right)$, the last step by $B = \Omega(Ld)$; that is the failure count of the statement. The certified per-point work $O\left(d_0 d\right)$ replaces a forward pass $O\left(Ld^2\right)$, a factor of Ld/d_0. The condition $B = \Omega\left(L^3 d^3 \rho_{\mathrm{m}} \Lambda\right)$ constrains the order as much as the batch size, since Λ is a sum over the $B-1$ steps of the order and in general grows with B; which orders meet it is taken up in Theorem A.19. □

The condition cannot be dropped: without it a single boundary already forces a linear number of failures.

Proposition A.18 (Anti-concentration is necessary)**.** *There is a network and a batch on which certificate-cached evaluation makes $\nu \ge B - 1$ failures under every order, so the bounded margin density of Theorem A.10 cannot be dropped from Theorem A.16.*

Proof. Take a network whose first hidden unit is $h_1^1(x) = w^\top x$ with $\|w\| = 1$, so that its zero set is a genuine hyperplane H, take $d_0 \ge 2$ so that a line parallel to H exists, and place $B \ge 4$ points on such a line at distance $\delta > 0$ from H, with successive positions spaced exactly 3δ apart. The statement is existential in the batch, so its size $B \ge 4$ and the distance δ are ours to choose; δ is fixed below, after ρ_{m}.

Every point y lies at distance exactly δ from H, and $g_{11} = w \ne 0$ whatever the pattern, since $M_1 = W_1$, so the unit $(1,1)$ enters the minimum of Eq. (11) and the safe radius satisfies $m(y) \le m_{11}(y) = \left|w^\top y\right| = \delta$ by Theorem A.7. Any two distinct points are at least 3δ apart, hence outside each other's safe balls, so under every order every step leaves the current anchor's safe ball and triggers a recomputation; thus $\nu \ge B - 1$, whatever δ is.

This network and batch violate Theorem A.10 at $t = 2\delta$: every point has margin exactly δ at the unit $(1,1)$, so $\mathbf{Pr}\left[m_{11}(x) < 2\delta\right] = 1$, against the $2\rho_{\mathrm{m}}\delta$ the assumption allows, which is below 1 once $\delta < 1/(2\rho_{\mathrm{m}})$. The pair defeats the conclusion Eq. (16) and not only the hypothesis, for every value of ρ_{m} one might put there.

Take the order along the line, of length $\Lambda = 3\delta(B-1) < 3\delta B$ — one order suffices, since Eq. (16) is asserted of the order used, while $\nu \ge B - 1$ holds under all of them. The right-hand side of Eq. (16) is then below $1 + 2B\sqrt{3N\rho_{\mathrm{m}}\delta}$. For $\delta < 1/\left(48N\rho_{\mathrm{m}}\right)$ this is below $1 + B/2$, since then $3N\rho_{\mathrm{m}}\delta < 3/48 = 1/16$ and the square root is below $1/4$. Since $1 + B/2 \le B - 1$ exactly when

$B \geq 4$, the bound is below the true $\nu \geq B - 1$. No bound of the form Eq. (16) can therefore hold without an anti-concentration hypothesis. □

The two propositions together bracket the average case: certificate failures are $O\left(\sqrt{BN\rho_{\mathrm{m}}\Lambda}\right)$ when margins are anti-concentrated and $\Omega(B)$ when they pile at zero.

Remark A.19 (How the curse is dodged, and what is left). The worst-case wall of Appendix B comes from locating a point against an adversarial arrangement. The certificate sidesteps location entirely: a point is evaluated by checking one distance, and a label is recomputed only when the next point leaves the current anchor's safe ball, which for a concentrated batch through a random network is rare. Every step is polynomial in the dimension, which is the sense in which the curse is dodged: no arrangement is built and nothing is enumerated, and the labels the method needs are the ones it produces itself, one per failure, rather than labels it is handed.

The curse does not disappear; it moves out of the per-operation cost and into the batch size at which the method wins. The threshold $B = \Omega\left(L^3 d^3 \rho_{\mathrm{m}} \Lambda\right)$ of Theorem A.17 is met only by orders with $\Lambda = O\left(B/\left(L^3 d^3 \rho_{\mathrm{m}}\right)\right)$. For B points spread through a ball of radius r in $\mathbb{R}^{d_0}$, the arrival order, the order in which the points are given, meets it only when $r = O\left(1/\left(L^3 d^3 \rho_{\mathrm{m}}\right)\right)$, an order blind to position having expected length $\Theta(rB)$ for points uniform in the ball. Sorting the batch along a space-filling curve first gives an order of length $\Lambda = O\left(rB^{1-1/d_0}\right)$ by Theorem A.5, at a cost $O\left(Bd_0 \log B\right)$ that the $O\left(Bd_0 d\right)$ of Theorem A.17 absorbs once $\log B = O(d)$. The threshold then reads $B = \Omega\left(Ld + \left(c_{\mathrm{o}} L^3 d^3 \rho_{\mathrm{m}} r\right)^{d_0}\right)$, where c_{o} is the constant of Theorem A.5 and the constant in the Ω is the d_0-th power of the one in the threshold of Theorem A.17: the same exponential in d_0 as Theorem A.6, alongside the base requirement $B = \Omega(Ld)$.

What the certificate gives is that the exponential sits in the regime of validity rather than in the running time, and that the condition producing it, Theorem A.10, is a joint property of the batch and the random network rather than of a fixed arrangement. The batch of Theorem A.18 is placed against one hyperplane of one network, whereas what the assumption asks is a bound on the margin's density with the network random and the batch fixed — a bound we establish at every depth for random biases and for bias-free Gaussian weights (Theorems A.11 and A.14), and not for deterministic biases in general.

This is the smoothed-analysis (Spielman & Teng, 2004) reading of the question: worst-case-hard point location becomes easy when the inputs are not placed adversarially on the boundaries, Theorem A.16 showing that bounded margin density suffices and Theorem A.18 that it cannot be dropped — the pair is a sufficient condition and one counterexample, not a characterization. The adversarial, worst-case batch remains.

B The Worst Case: Batched Evaluation Is Matrix Multiplication

The regime of Appendix A makes point location cheap through average-case structure. We now confront the generic high-dimensional case, the one Theorem 3.2 identified as the bottleneck, and show that in the arithmetic model of Section 2.1 it is neither hopeless nor mysterious: batched evaluation is matrix multiplication, so fast matrix multiplication already beats the pointwise baseline, and no *layer* can be beaten by more than the matrix-multiplication factor. What the L layers together admit is a separate question, and Theorem B.6 keeps it open.

B.1 The Batched Pass as Layered Matrix Multiplication

We record the whole batch's forward pass as one sequence of matrix products, the form that Theorem B.2 costs by fast matrix multiplication.

Lemma B.1 (The batched pass is a sequence of matrix products)**.** *Let $X \in \mathbb{R}^{d_0 \times B}$ have the B inputs $x_1, \ldots, x_B$ of a batch as its columns, and set $Z_0 = X$ and, for $\ell \in [L]$,*

$$H_\ell = W_\ell Z_{\ell-1} + b_\ell \mathbf{1}^\top, \qquad Z_\ell = \sigma\left(H_\ell\right) \text{ for } \ell < L, \qquad \text{output } H_L, \tag{18}$$

where $\mathbf{1} \in \mathbb{R}^B$ is the all-ones vector. Then column i of H_ℓ is $h_\ell\left(x_i\right)$ for every $\ell \in [L]$ and $i \in [B]$, so Eq. (18) *is the forward pass run on all inputs at once. All activation patterns are read off from*

the signs of the columns of the H_ℓ with $\ell < L$. The only superlinear work is the L matrix products $W_\ell Z_{\ell-1}$.

Proof. We argue by induction on ℓ that column i of $Z_{\ell-1}$ equals $z_{\ell-1}(x_i)$ for every $\ell \in [L]$. At $\ell = 1$ this is $Z_0 = X$. If it holds at ℓ, then column i of $H_\ell = W_\ell Z_{\ell-1} + b_\ell \mathbf{1}^\top$ is $W_\ell z_{\ell-1}(x_i) + b_\ell = h_\ell(x_i)$ by Eq. (1), whence column i of Z_ℓ is $\sigma(h_\ell(x_i)) = z_\ell(x_i)$ for $\ell < L$. The claim therefore holds for every $\ell \in [L]$, and the displayed identity then gives column i of H_ℓ as $h_\ell(x_i)$ for every $\ell \in [L]$, including $\ell = L$, where H_L is the output and no rectifier is applied. So Eq. (18) is the pointwise recurrence run on all inputs at once. The bias broadcast $b_\ell \mathbf{1}^\top$, the entrywise rectifier, and reading $\pi_\ell = \mathbb{1}(H_\ell > 0)$ each cost $O(d_\ell B)$, which is linear in the size of H_ℓ; all remaining work is the product $W_\ell Z_{\ell-1}$. □

B.2 Beating the Baseline by Fast Matrix Multiplication

Let $\mathrm{MM}(a, b, c)$ be the number of arithmetic operations needed to multiply an $a \times b$ matrix by a $b \times c$ matrix. Let ω be the square exponent, the infimum of the τ for which $\mathrm{MM}(n, n, n) = O(n^\tau)$, so that $\mathrm{MM}(n, n, n) = n^{\omega + o(1)}$ with $2 \le \omega < 2.372$ (Le Gall, 2014; Alman et al., 2025) (subcubic since Strassen (1969)). The $n^{o(1)}$ is the one in the definition of ω as an infimum, and we carry it rather than assume an algorithm attaining the exponent; it is suppressed wherever the context makes the reading plain. Let α be the dual exponent, the largest β for which $\mathrm{MM}\left(n, n, n^\beta\right) = n^{2+o(1)}$. The largest such β exists: the set of such β contains 0 and is an initial segment of $[0, \infty)$, since MM is nondecreasing in its third argument by zero-padding; it is bounded by 1, since the product has $n^{1+\beta}$ entries; and its supremum belongs to it because the exponent of $\mathrm{MM}\left(n, n, n^\beta\right)$ is a convex, hence continuous, function of $\beta > 0$ (Lotti & Romani, 1983). Moreover $\alpha > 0.32$ (Le Gall & Urrutia, 2018; Vassilevska Williams et al., 2024).

Theorem B.2 (Evaluation and point location by fast matrix multiplication)**.** *A batch of B inputs to an L-layer ReLU network with widths $d_0, \ldots, d_L$ is evaluated, and all its activation patterns are computed, in*

$$O\left(\sum_{\ell=1}^{L} \mathrm{MM}(d_\ell, d_{\ell-1}, B) + B \sum_{\ell=1}^{L} d_\ell\right) \tag{19}$$

arithmetic operations.

Proof. By Theorem B.1 the work splits into the L products $W_\ell Z_{\ell-1}$, of cost $\mathrm{MM}(d_\ell, d_{\ell-1}, B)$ each, and the entrywise bias, rectifier, and sign extraction, of cost $O(d_\ell B)$ per layer. Summing over layers gives Eq. (19). □

Specializing Eq. (19) to uniform width gives two regimes, set by whether the batch size is at least d or at most d^α, with α the dual exponent.

Corollary B.3 (Uniform width: two regimes)**.** *Let the network have uniform width d, so that $T = \Theta\left(Ld^2\right)$. Every $o(1)$ below is taken along a sequence of such networks and batches with $d \to \infty$ and the depth L arbitrary.*

- *(i) If $B \ge d$, the bound of Eq.* (19) *is $O\left(LBd^{\omega-1+o(1)}\right)$, a factor of $\Theta\left(d^{3-\omega-o(1)}\right)$ below the pointwise $\Theta\left(BLd^2\right)$; with $\omega < 2.372$ this factor exceeds $d^{0.6}$.*
- *(ii) If $B \le d^\alpha$, the bound of Eq.* (19) *is $O\left(Ld^{2+o(1)}\right)$, the cost of a single forward pass up to a $d^{o(1)}$ factor, hence smaller than the pointwise $\Theta\left(BLd^2\right)$ by a factor of $B/d^{o(1)}$, which is a polynomial factor once $B = d^{\Omega(1)}$.*

Proof. For (i), split the $d \times B$ matrix $Z_{\ell-1}$ into $\lceil B/d \rceil$ blocks of d columns. Each block gives a $d \times d$ by $d \times d$ product of cost $O\left(d^{\omega+o(1)}\right)$, so $\mathrm{MM}(d, d, B) = O\left(\lceil B/d \rceil d^{\omega+o(1)}\right) = O\left(Bd^{\omega-1+o(1)}\right)$; the entrywise term $O(LdB)$ is dominated because $\omega \ge 2$. For (ii), by the definition of α, $\mathrm{MM}(d, d, \lceil d^\alpha \rceil) = d^{2+o(1)}$. Moreover MM is nondecreasing in its third argument, because a narrower right factor is padded with zero columns. So $\mathrm{MM}(d, d, B) = d^{2+o(1)}$ for every $B \le d^\alpha$. The

entrywise term $O(LdB)$ is $O\left(Ld^2\right)$, because $B \le d^\alpha$ and $\alpha \le 1$. Summing over the L layers gives $O\left(Ld^{2+o(1)}\right)$, against the pointwise $\Theta\left(BLd^2\right)$. □

Part (ii) of the corollary just proved deserves its own statement, since it says a whole batch is evaluated for close to the asymptotic price of one input.

Remark B.4 (Thin batches are asymptotically free). Part (ii) of Theorem B.3 is the paper's sharpest answer to the question that opened it: up to $d^{0.32}$ inputs are evaluated for the asymptotic price of a single one. Both parts are oblivious to the geometry, using no region structure at all, so they apply to arbitrary unlabeled high-dimensional batches; they supersede the pessimistic reading of Theorem 3.2, under which no asymptotic gain would have been possible.

B.3 Matrix Multiplication Is Also Necessary

Theorem B.2 evaluates the batch by matrix multiplication, and the following proposition shows that, under its hypotheses, matrix multiplication is also needed.

Proposition B.5 (The batched problem is matrix multiplication)**.** *The product AC of an $m \times k$ matrix A by a $k \times n$ matrix C is computed by evaluating a one-layer ReLU network ($L = 1$, which has no rectifier) of width m on a batch of n inputs in dimension k. Hence any algorithm that, given the weights and the batch as input, evaluates batches of ReLU networks in the arithmetic-circuit model multiplies matrices at the same asymptotic cost, and the per-layer optimum in Theorem B.2 is exactly* MM*. The exponents agree layer by layer, and for the whole network only when the depth is negligible: the lower bound is one product and the upper bound is L of them, so the two coincide for $L = n^{o(1)}$ and are separated by the depth otherwise. Moreover, for a network with one hidden layer, computing the hidden activation pattern is the problem of determining the sign of every entry of a fixed matrix applied to the batch, the bias being absorbed by appending it to the weight matrix as a column and appending to the batch a row of ones.*

Proof. A one-layer network ($L = 1$) is $f(x) = W_1 x + b_1$ with no rectifier, the degenerate member of the class. Taking $W_1 = A$, $b_1 = 0$, and the batch equal to the columns of C produces the outputs AC, so a batched evaluation solves the matrix product; combined with Theorem B.2 the per-layer cost is MM up to constants. The qualification that the weights are part of the input is needed, since after preprocessing the network a batch confined to a planar slice is evaluated below MM by locating it in the cell enumeration of (Humayun et al., 2023).

For the last claim, take $L = 2$, so that the pattern is π_1 and $z_0 = x$ is already affine. Then $\pi_1^i(x) = \mathbb{1}\left((W_1 x + b_1)^i > 0\right)$ for every x, and the batch pattern is the entrywise positivity pattern of $MX + u\mathbf{1}^\top$ with $M = W_1$ and $u = b_1$. Since $MX + u\mathbf{1}^\top = \begin{pmatrix} M & u \end{pmatrix}\begin{pmatrix} X \\ \mathbf{1}^\top \end{pmatrix}$, this is the positivity pattern of a single product of the weight matrix augmented by the bias column and the batch augmented by a row of ones, which is the absorption the statement describes.

At one hidden layer no restriction on the batch is needed. We do not claim the same for a deeper layer.

The vector $z_{\ell-1}$ is affine in x only where the units below layer ℓ do not switch, and no fixed bias keeps a unit with weight row $W^j \neq 0$ active on an unbounded batch, since a point far enough along $-W^j$ deactivates it. That route would also have to measure the bias against the image of the batch under the layers beneath rather than against the batch itself, and we do not take it. □

The following remark collects what Theorem B.5 leaves open, across depth and across dimension.

Remark B.6 (What is left open). In the high-dimensional generic case the labels-versus-evaluation question of Section 1.5 is therefore settled in the arithmetic model up to the per-layer matrix-multiplication factor: the pointwise baseline is beatable, by $\Theta\left(d^{3-\omega-o(1)}\right)$ for $B \ge d$, and no *single* layer can be computed below MM. Whether the L rounds themselves amortize is a separate question, left open by Theorem B.12 and Theorem B.14, and one layer's lower bound does not close it: with $W_\ell = I$ and $b_\ell = 0$ for $2 \le \ell \le L - 1$ every hidden activation equals z_1, the network collapses to $f(X) = W_L \sigma\left(W_1 X + b_1 \mathbf{1}^\top\right) + b_L \mathbf{1}^\top$, and the batch is evaluated in two products rather than L.

The other question left open is point location proper. By Theorem B.5 it is the sign pattern of a matrix product, which is upper bounded by forming the product but might be cheaper. Computing all signs of MX is point location of the batch against the hyperplane arrangement of the rows of M.

On a line and on a planar slice, the exact enumerations of Sotoudeh & Thakur (2019); Humayun et al. (2023) compute the cells that the batch meets. In fixed dimension the product itself already gives all signs in output-linear time, as Theorem B.7 explains.

In dimension growing with the data no algorithm beating matrix multiplication is known, and even deciding whether *some* entry of MX vanishes is Hopcroft's problem for hyperplanes through the origin: solvable in $O\left(n^{2d_0/(d_0+1)}\right)$ time in every fixed dimension (Chan & Zheng, 2024), and for inputs with $O(\log n)$-bit integer entries requiring $n^{2-o(1)}$ time under the strong exponential time hypothesis once $d_0 = \omega(\log \log n)$ (Williams, 2018); its unconditional complexity in growing dimension is open (Erickson, 1996). The precise residue of the question is therefore whether the sign pattern of a matrix product can be computed faster than the product.

The open question just stated is not visible at every input dimension, and the following remark locates where it lives.

Remark B.7 (The residue is a growing-dimension phenomenon). In fixed input dimension $d_0 = O(1)$ there is no gap at a single layer: the sign pattern of $W_1 X$ is read from the product in $O\left(d_0 B d\right) = O(Bd)$, the size of the output itself, so the labels cost no more than reading them. The residue of Theorem B.6 therefore lives entirely in growing dimension, where the question is whether the n^2 sign bits of an $n \times n$ product can be produced output-optimally in $n^{2+o(1)}$ rather than in the $n^{\omega+o(1)}$ of forming the product. No method below the matrix-multiplication exponent is known for the sign pattern in growing dimension. The only unconditional lower bound for the full pattern is the output bound $\Omega\left(n^2\right)$ of Section 2.1; Theorem B.9 supplies the conditional side. Closing the gap amounts to deciding whether the combinatorial sign information is strictly cheaper than the algebraic product. This we leave open.

B.4 The Fine-Grained Anchor Is Boolean Matrix Multiplication

The residue of Theorem B.7 has a precise fine-grained identity, and two nearby candidates do not fit it. The all-pairs shortest paths problem is the min-plus product, a different algebra, and bipartite point–hyperplane incidence is Hopcroft's problem, solvable in the plane in $O\left(n^{4/3}\right)$ (Chan & Zheng, 2024) and hence below the quadratic 3SUM threshold. The anchor is Boolean matrix multiplication.

Fact B.8 (Itai & Rodeh, 1978; Vassilevska Williams & Williams, 2018)**.** *Detecting a triangle in an n-vertex graph reduces to one Boolean product of two $n \times n$ matrices (Itai & Rodeh, 1978). Conversely, Boolean matrix multiplication and triangle detection have truly subcubic combinatorial algorithms together or not at all (Vassilevska Williams & Williams, 2018, Theorem 6.1). The Boolean product is computable in $n^{\omega+o(1)}$ operations by forming an integer product and testing which entries are positive; no $n^{2+o(1)}$ algorithm is known. The min-plus product is likewise subcubic-equivalent to all-pairs shortest paths, a classical equivalence recorded as items 1 and 5 of (Vassilevska Williams & Williams, 2018, Theorem 1.1).*

Theorem B.8 records the fine-grained standing of Boolean matrix multiplication. The following proposition transfers that hardness to labeling, which anchors the residue there.

Proposition B.9 (Labeling is Boolean-matrix-multiplication-hard)**.** *Computing the activation pattern of a hidden layer of width n on a batch of n points in dimension d is at least as hard as the Boolean product of an $n \times d$ matrix by a $d \times n$ matrix. Its best known bound is the $O\left(\mathrm{MM}(n, d, n)\right)$ of Theorem B.2, which is $O\left(n^{\omega}\right)$ for $d \leq n$. At $d = n$, and by zero-padding at every $d \geq n$, it is therefore triangle-detection-hard, and an output-optimal $n^{2+o(1)}$ labeling at $d = n$ would give $n^{2+o(1)}$ Boolean matrix multiplication, an open problem.*

Proof. Let $A \in \{0,1\}^{n \times d}$ and $C \in \{0,1\}^{d \times n}$ be a Boolean-product instance, and write $c_1, \ldots, c_n$ for the columns of C. Take a network with one hidden layer, $W_1 = A$ and zero bias, and the batch $X = C$; the output layer is immaterial. Then $h_1^i\left(c_j\right) = \sum_k A_{ik} C_{kj} = (AC)_{ij}$, a non-negative integer, so

$$\pi_1^i\left(c_j\right) = \mathbb{1}\left((AC)_{ij} > 0\right) = \mathbb{1}\left(\exists k \ \left(A_{ik} = C_{kj} = 1\right)\right),$$

which is exactly the (i, j) entry of the Boolean product $A \odot C$; the pattern bit $\mathbb{1}(h > 0)$ loses nothing at zero here because the pre-activations are non-negative. The instance is written down in $O(nd)$ time. Labeling the batch therefore computes $A \odot C$. At $d = n$ this is the square Boolean product, so labeling inherits the hardness recorded in Theorem B.8, whose first reduction takes triangle detection to one such product; for $d > n$ the square product is the instance in which the last $d - n$ columns of A and rows of C vanish. The upper bound is Theorem B.2, and an $n^{2+o(1)}$ labeling at $d = n$ in the model of Section 2.1 would compute $A \odot C$ in $n^{2+o(1)}$ arithmetic operations. □

Theorem B.9 ties labeling to triangle detection at input dimension $d \geq n$. The following corollary shows that labeling is output-optimal up to $d = n^{\alpha}$, so any possible super-output labeling cost lies above the dual exponent α.

Corollary B.10 (The gap opens above the dual exponent)**.** *Consider a hidden layer of width n on a batch of n points in dimension d. If $d \leq n^{\alpha}$ for the dual matrix-multiplication exponent α of Appendix B.2, with $\alpha > 0.32$ (Le Gall & Urrutia, 2018; Vassilevska Williams et al., 2024), then $\mathrm{MM}(n, d, n) = n^{2+o(1)}$ and labeling is output-optimal. For power-law dimensions $d = n^{\beta}$, any possible super-output labeling cost requires $\beta > \alpha$; the square Boolean-product reduction of Theorem B.9 applies when $d \geq n$.*

Proof. The dual exponent is defined in Appendix B.2 through $\mathrm{MM}\left(n, n, n^{\beta}\right)$, with the small dimension last; the exponent of rectangular matrix multiplication is invariant under permuting its three arguments (Hopcroft & Musinski, 1973; Lotti & Romani, 1983). Write $\langle a, b, c\rangle$ for the tensor of the product of an $a \times b$ by a $b \times c$ matrix, whose rank determines the exponent of that product's cost. Then the tensor $\langle b, c, a\rangle$ is obtained from $\langle a, b, c\rangle$ by cyclically permuting its three factors, $\langle c, b, a\rangle$ is obtained from it by transposition, and tensor rank is invariant under both (Bläser, 2013, Lemma 5.5). So $\mathrm{MM}\left(n, n^{\alpha}, n\right) = n^{2+o(1)}$ as well, and MM is nondecreasing in each argument by zero-padding. Forming $W_1 X$ with $d \leq n^{\alpha}$ therefore costs $n^{2+o(1)}$, which is the size of the batch of patterns and hence optimal up to sub-polynomial factors by the output bound of Section 2.1. For $d = n^{\beta}$ with $\beta > \alpha$ a constant the product is not $n^{2+o(1)}$: since $\beta > \alpha$, the maximality of α in its definition in Appendix B.2 and the permutation invariance just used show that the exponent of $\mathrm{MM}\left(n, n^{\beta}, n\right)$ is not 2; it is at least 2 because the product has n^2 entries, so it exceeds 2 by a constant depending on β. Whether the signs alone are cheaper is the open Boolean-multiplication question of Theorem B.9. □

The results since Theorem B.5 now compose into one picture of where the difficulty sits.

Remark B.11 (The landscape, by dimension and by depth)*.* The complexity turns on the input dimension. In fixed dimension a single layer's labels are read from the product in the size of the output (Theorem B.7); at input dimension $d \geq n$, *computing* the full sign pattern is at least as hard as Boolean matrix multiplication (Theorem B.9), between n^2 and n^{ω} when $d = n$; we prove one direction only, since for real weights the sign pattern is not itself a Boolean product. A depth-L network labels a batch by L sequential single-layer labelings, each inheriting this hardness: the instance of Theorem B.9 is planted at layer ℓ by taking $W_1, \ldots, W_{\ell-1}$ to be identity matrices with zero biases, which act as the identity on the non-negative batch that reduction uses, so that layer ℓ receives the batch unchanged. Depth adds a factor linear in L at the output rate (Theorem B.12), and whether it compounds the per-layer hardness further is a question we delimit below (Theorem B.13). The point-location term of the paper is thus, at one layer, at least as hard as Boolean matrix multiplication.

The following bound shows how the cost of labeling grows with the depth L.

Proposition B.12 (Depth enters linearly, at the output rate)**.** *Labeling a depth-L network of uniform width d, with $L \geq 2$, on a batch of B points requires $\Omega\left(LdB\right)$ operations in the model of Section 2.1, and Theorem B.2 attains $O\left(L \cdot \mathrm{MM}(d, d, B)\right)$. Both scale linearly in L, so depth contributes a factor exactly linear at the output rate, and the only room for compounding is whether the per-layer overhead $\mathrm{MM}(d, d, B)/(dB)$ recurs L times or amortizes to once.*

Proof. The output is the $N = (L - 1)d = \Theta(Ld)$ hidden bits at each of the B points, using $L \geq 2$, and the model of Section 2.1 charges one unit per value written, so producing it costs

$\Omega(NB) = \Omega(LdB)$. The upper bound is Theorem B.2 with L layers of width d, whose MM term dominates the $O(LdB)$ elementwise work. □

Whether a lower bound can compound with depth beyond the linear rate of Theorem B.12 is open, and the following remark explains why the standard fine-grained toolkit does not reach that question.

Remark B.13 (Why depth resists the fine-grained toolkit). A super-linear compounding would need a fine-grained problem whose hardness grows with circuit depth. The depth-parameterized problems in the toolkit, k-orthogonal-vectors and k-clique, are hard in their *search* dimension — the number of quantified tuples — which for labeling is the batch size and the input dimension, not the network depth; as threshold circuits both are of constant depth. Moreover, a fixed batch of B points visits at most B of the $d^{\Theta(d_0 L)}$ regions the network admits, so depth cannot enlarge the batch's effective search. The reductions of this section therefore act at a single layer, and a depth-compounding lower bound lies beyond them; we leave it open.

The barrier just described concerns lower bounds. The following remark takes up the upper-bound question, whether the L labeling rounds can be amortized.

Remark B.14 (Amortization is an evaluation phenomenon, not a labeling one). On the upper side, one asks whether the L labeling rounds can be amortized below the $L \cdot \mathrm{MM}(d, d, B)$ upper bound of Theorem B.2. The region amortization of Theorem 3.1 does collapse the L applications of a *labeled* batch into one, forming a region's composed map A_p once and applying it once, independent of depth. Labeling does not collapse this way: splitting a group of inputs by their layer-ℓ signs needs their layer-ℓ pre-activations, so the natural labeling algorithm applies maps at every layer and runs all L sign-computing rounds.

Grouping by pattern moreover fragments the batch on which the fast matrix multiplication of Theorem B.2 depends. With $R = \Theta(B)$ singleton regions in the worst case and $B \geq d$, so that Theorem B.3 (i) applies, the per-layer cost rises from the $O\left(Bd^{\omega-1+o(1)}\right)$ of one product to the $\Theta\left(Bd^2\right)$ of B matrix–vector products, a factor $d^{3-\omega-o(1)}$, so grouping strictly loses. For a thin batch $B \leq d^{\alpha}$ the comparison is $d^{2+o(1)}$ against Bd^2 and the loss is $B/d^{o(1)}$ instead.

The algebra amortizes once labels are known; this natural pattern-grouping method does not amortize their computation. Whether a worst-case batch admits any sub-L labeling is the open upper-bound question, and we have no algorithm for it.

The average-case counterpart is already answered, with one qualification: the certificate method of Appendix A labels a concentrated batch through a random network in the sub-L cost of Theorem A.17, but a certified point's label exists there only as a pointer to its anchor's pattern, and writing the B patterns out costs a further $\Theta(BLd)$, which stays inside the stated $O(Bd_0 d)$ only for $L = O(d_0)$; since Theorem A.17 assumes only $d_0 = O(d)$ and allows $d_0 \ll d$, this is a stronger restriction than $L = O(d)$. Some structured weights do collapse depth, as we now show.

One might hope that weights cheap to multiply — circulant, sparse, or low rank — let the L products compose and collapse the rounds. Generically they do not, and the reason is that the rectifier commutes with almost no linear map: only with those that are non-negative and have at most one nonzero entry per row, a class that includes singular maps such as $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ and so is wider than the monomial matrices.

Lemma B.15 (The rectifier commutes only with non-negative row-monomial maps)**.** *A linear map $M \in \mathbb{R}^{m \times n}$ satisfies $\sigma(Mz) = M\sigma(z)$ for all $z \in \mathbb{R}^n$ if and only if $M \geq 0$ and each row of M has at most one nonzero entry.*

Proof. If $M \geq 0$ and row k has its only nonzero entry in column $\iota(k)$, then $(Mz)_k = M_{k\iota(k)} z_{\iota(k)}$, and since $M_{k\iota(k)} \geq 0$,

$$\sigma\left(M_{k\iota(k)} z_{\iota(k)}\right) = M_{k\iota(k)} \sigma\left(z_{\iota(k)}\right) = (M\sigma(z))_k .$$

If row k is zero, both $(Mz)_k$ and $(M\sigma(z))_k$ vanish, and $\sigma(0) = 0$, so the two sides agree at that row as well. Conversely, taking $z = e_j$ gives $\sigma(M_{kj}) = M_{kj}$, so $M \geq 0$; and taking $z = e_i - te_j$ with $i \neq j$ and $t > 0$ gives $(M\sigma(z))_k = M_{ki}$ while $\sigma(Mz)_k = \sigma(M_{ki} - tM_{kj})$, which vanishes as soon as $t > M_{ki}/M_{kj}$ when $M_{kj} > 0$. Equating the two at such a t gives $M_{ki} = 0$. So if some entry of row k is nonzero, every other entry of that row vanishes: each row has at most one nonzero entry. □

Theorem B.15 characterizes the weights that commute with the rectifier, and the following proposition works out what they give for depth.

Proposition B.16 (A structured depth collapse)**.** *Suppose $W_2, \dots, W_{L-1}$ commute with the rectifier and the biases $b_2, \dots, b_{L-1}$ vanish. Then $z_\ell = (W_\ell \cdots W_2)\,\sigma(W_1 x + b_1)$ for $1 \le \ell \le L-1$, the product being empty at $\ell = 1$; every hidden pre-activation past the first layer, that is h_ℓ for $2 \le \ell \le L-1$, is non-negative; and the whole activation pattern is a function of π_1 computable from it in $O(N)$ operations. Labeling therefore reduces to the single layer $\sigma(W_1 x + b_1)$.*

Among maps that commute with the rectifier on every input, these are the only weights: with the stated bias condition, they collapse the network to one nonlinearity. The bias condition matters: in a three-layer network with $W_1 = W_2 = [1]$, $b_1 = 0$ and $b_2 = -1$, the layer-2 unit is inactive at $x = 1/2$ and active at $x = 2$ while $\pi_1 = 1$ at both points, so π_2 is not a function of π_1 and a bias alone restores the second round.

This is a sufficient condition, not a characterization of depth collapse. Even a non-negative matrix with several positive entries per row makes the next rectifier redundant on a non-negative hidden activation when its bias vanishes. Circulant, sparse, or low-rank structure alone does not guarantee such a collapse: negative weights or later biases can create new sign decisions.

Proof. By Theorem B.15 each $W_\ell \ge 0$ for $2 \le \ell \le L-1$. Then $z_1 = \sigma(W_1 x + b_1) \ge 0$, and inductively, using the vanishing biases, $z_\ell = \sigma(W_\ell z_{\ell-1}) = W_\ell \sigma(z_{\ell-1}) = W_\ell z_{\ell-1} = (W_\ell \cdots W_2)\, z_1$, by commutation, the idempotence $\sigma(z_{\ell-1}) = z_{\ell-1}$ of a non-negative vector, and $z_\ell \ge 0$ from $W_\ell \ge 0$. Hence $h_\ell = W_\ell z_{\ell-1} \ge 0$ for $2 \le \ell \le L-1$; the output pre-activation h_L is excluded, b_L being unconstrained. The pattern bit of such an h is $\mathbb{1}(h > 0)$, which is 0 where $h = 0$, and $h = 0$ holds on a set of positive measure whenever the first-layer row it traces back to is nonzero: if W_ℓ has its row-k entry in column $\iota(k)$, then $h_\ell^k = 0$ exactly where $z_{\ell-1}^{\iota(k)} = 0$, which traces back to a first-layer unit being inactive. A zero first-layer row makes the bit constant instead. So the bit is not identically one, but it is determined by π_1: unrolling the row-monomial maps, h_ℓ^k is a non-negative multiple of z_1^j for a single j, and its bit is that of π_1^j whenever the multiple is positive and 0 otherwise. Composing the $L-1$ index maps costs $O(N)$, and only $\sigma(W_1 x + b_1)$ carries independent information. □

C Approximate Evaluation and the Conditioning Obstacle

Every speedup so far is exact, and either galactic (Appendix B) or confined to concentrated batches through random networks (Appendix A). We turn to approximation, the one regime that can serve an arbitrary high-dimensional batch with practical constants, and we begin with the apparent obstacle, which is not the real one.

This appendix measures matrices by two norms. For a matrix W we write $\|W\|_2$ for the spectral norm, its largest singular value. We write $\|W\|_F$ for the Frobenius norm, the Euclidean norm of the vector of its entries, whose square is the sum of the squares of the singular values. The ratio $\|W\|_F^2 / \|W\|_2^2$ is the stable rank of W. Two facts about these norms are used without comment. The first is $\|W\|_2 \le \|W\|_F$, since the spectral norm is the largest of the singular values, whose squares sum to $\|W\|_F^2$. The second is $\|WZ\|_F \le \|W\|_2 \|Z\|_F$, obtained by applying $\|Wz\| \le \|W\|_2 \|z\|$ to each column z of Z and summing the squares. The first fact says that the stable rank of a nonzero W is at least 1. The stable rank is also at most the rank r of W. Each of the r nonzero singular values has square at most $\|W\|_2^2$, so $\|W\|_F^2 \le r \|W\|_2^2$. Equality holds exactly when all r of them are equal. An orthogonal $d \times d$ matrix has singular values all equal to 1, so it attains the upper bound with stable rank d. A matrix of rank one attains the lower bound with stable rank 1.

C.1 The Sign-Compounding Concern Is Illusory

It is natural to expect approximation to fail because a perturbed pre-activation may cross zero, flip a rectifier, and corrupt the computation downstream. For forward evaluation no such corruption occurs.

Lemma C.1 (Rectified forward error is Lipschitz)**.** *For pre-activation vectors h and $\tilde{h}$ we have $\left\|\sigma(\tilde{h}) - \sigma(h)\right\| \le \left\|\tilde{h} - h\right\|$ regardless of any sign change, and for pre-activation matrices H and $\tilde{H}$ of the same shape likewise $\left\|\sigma(\tilde{H}) - \sigma(H)\right\|_F \le \left\|\tilde{H} - H\right\|_F$. Consequently, applying exact rectifiers to approximate pre-activations propagates error through the network by the layer operator norms alone, and the discrete activation pattern is never needed.*

Proof. Fix a coordinate, writing a for $\tilde{h}^j$ and b for h^j; we check $|\max(a,0) - \max(b,0)| \le |a - b|$ in three cases. If $a, b \ge 0$ the left side equals $|a - b|$; if $a, b \le 0$ it is 0. If they straddle zero, say $a > 0 \ge b$, the left side is a while the right is $a - b = |a| + |b| \ge a$; the case $b > 0 \ge a$ is symmetric. Summing the squares of the coordinate inequalities gives the Euclidean bound. Since σ acts entrywise and the Frobenius norm of a matrix is the Euclidean norm of the vector of its entries, the same inequality holds for matrices in the Frobenius norm, which is the form Theorems C.3, C.4 and C.9 use. The error of a forward pass therefore obeys a recursion in the layer norms, with no term created by a flipped sign. □

The error of an approximate forward pass is thus numerical, not combinatorial; what governs it is how the layers amplify it.

C.2 Sketched Forward Evaluation

We approximate each batched product $W_\ell Z_{\ell-1}$ by the sketch $(W_\ell S_\ell)\left(S_\ell^\top Z_{\ell-1}\right)$, where $S_\ell \in \mathbb{R}^{d_{\ell-1} \times s}$ is a matrix of independent $N(0, 1/s)$ entries drawn independently of everything else, and where the data-independent factor $W_\ell S_\ell$ is precomputed once. The sketch is unbiased, $\mathbb{E}\left[S_\ell S_\ell^\top\right] = I_{d_{\ell-1}}$. The (i, j) entry of $S_\ell S_\ell^\top$ is the inner product of rows i and j of S_ℓ, a sum of s products of two entries. For $i \ne j$ each product has mean 0, its two factors being independent with mean 0, and for $i = j$ each product is the square of an entry, of mean $1/s$, so the s of them sum to mean 1. By linearity of expectation, $\mathbb{E}\left[(W_\ell S_\ell)\left(S_\ell^\top Z\right)\right] = W_\ell \mathbb{E}\left[S_\ell S_\ell^\top\right] Z = W_\ell Z$ for every fixed matrix Z: the estimator routes the product through the s rows of $S_\ell^\top Z$ and is right on average, and what remains to control is how far one draw strays from its mean. The one result this section imports is the guarantee such a sketch carries.

Fact C.2 (Approximate matrix multiplication; Kane & Nelson, 2014; Clarkson & Woodruff, 2009)**.** *Let $S \in \mathbb{R}^{d \times s}$ have independent $N(0, 1/s)$ entries. There is an absolute constant C such that for every fixed $W \in \mathbb{R}^{d' \times d}$ and $Z \in \mathbb{R}^{d \times B}$ and every $\varepsilon_{\mathrm{s}}, \delta \in (0, 1/2)$, once $s \ge C\varepsilon_{\mathrm{s}}^{-2} \log(1/\delta)$,*

$$\left\|(WS)\left(S^\top Z\right) - WZ\right\|_F \le \varepsilon_{\mathrm{s}} \|W\|_F \|Z\|_F$$

with probability at least $1 - \delta$.

The bound is Woodruff (2014, Theorem 13), due to Kane & Nelson (2014), applied to a Gaussian matrix, which has the moment property that theorem asks for at this s. For a matrix of independent $\pm 1/\sqrt{s}$ entries the bound is Theorem 2.2 of Clarkson & Woodruff (2009); Sarlós (2006) gives it with an extra logarithmic factor, and Drineas et al. (2006) gives it for the column-sampling estimator.

Assuming $H_L \ne 0$, we define the forward condition number of f on the batch X to be

$$\kappa = \frac{1}{\|H_L\|_F} \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \|W_j\|_2 \right) \|W_\ell\|_F \|Z_{\ell-1}\|_F . \tag{20}$$

In words, the summand of index ℓ is the scale of the error a sketch of the layer-ℓ product can make, carried to the output by the operator norms of the later layers. The quantity κ measures their sum against the size of the output.

The smallest case shows what κ measures. For one layer with no bias, $L = 1$ and $b_1 = 0$, the output is $H_1 = W_1 X$, and Eq. (20) has a single summand, whose product over j is empty, so $\kappa = \|W_1\|_F \|X\|_F / \|W_1 X\|_F$ for every batch with $W_1 X \ne 0$. Since $\|W_1 X\|_F \le \|W_1\|_2 \|X\|_F$, this κ is at least $\|W_1\|_F / \|W_1\|_2$, the square root of the stable rank of W_1. Equality holds exactly when every column of X lies in the span of the right singular vectors of W_1 for its largest singular value.

To see this, write a column as $x = \sum_i \alpha_i v_i$ in an orthonormal basis of right singular vectors v_i of W_1, with singular values σ_i, zero on the kernel. Then $\|W_1 x\|^2 = \sum_i \sigma_i^2 \alpha_i^2$, which equals $\|W_1\|_2^2 \|x\|^2 = \sigma_1^2 \sum_i \alpha_i^2$ exactly when $\alpha_i = 0$ wherever $\sigma_i < \sigma_1$, and summing over the columns, the inequality $\|W_1 X\|_F \le \|W_1\|_2 \|X\|_F$ is an equality exactly when it is one column by column.

In the other direction, when W_1 has a nontrivial kernel, κ is unbounded as the batch tilts toward it. Suppose the kernel contains a unit vector u, and let v_1 be a unit right singular vector for the largest singular value, which is orthogonal to u because the kernel is orthogonal to every right singular vector of nonzero singular value. The single input $x = v_1 + tu$ has $W_1 x = W_1 v_1$, of norm $\|W_1\|_2$, and $\|x\| = \sqrt{1+t^2}$, so $\kappa = \sqrt{1+t^2}\,\|W_1\|_F / \|W_1\|_2$, which grows without bound with t.

For deeper networks, the summand of index ℓ in Eq. (20), divided by $\|H_L\|_F$, bounds the relative output error left by an error of Frobenius norm $\|W_\ell\|_F \|Z_{\ell-1}\|_F$ in the product $W_\ell Z_{\ell-1}$ alone, which is the scale on which Theorem C.2 measures the error of a sketch. The rectifiers do not enlarge that error, by Theorem C.1, and each later layer j multiplies it by at most $\|W_j\|_2$, because $\|W_j D\|_F \le \|W_j\|_2 \|D\|_F$ for every matrix D.

Lemma C.3 (Vanishing biases bound the activations)**.** *Suppose every bias vanishes, $b_\ell = 0$ for all $\ell \in [L]$. Then $\|Z_\ell\|_F \le \|W_\ell\|_2 \|Z_{\ell-1}\|_F$ for every $\ell < L$ and $\|H_L\|_F \le \|W_L\|_2 \|Z_{L-1}\|_F$, so that for every $\ell \in [L]$*

$$\|Z_{\ell-1}\|_F \le \left(\prod_{j<\ell} \|W_j\|_2\right) \|X\|_F \qquad \text{and} \qquad \|H_L\|_F \le \left(\prod_{j\ge\ell} \|W_j\|_2\right) \|Z_{\ell-1}\|_F . \tag{21}$$

If moreover $H_L \neq 0$ and every $W_\ell \neq 0$, then $\kappa \ge \sum_{\ell=1}^{L} \|W_\ell\|_F / \|W_\ell\|_2$.

Proof. For $\ell < L$, Eq. (18) with $b_\ell = 0$ reads $Z_\ell = \sigma(W_\ell Z_{\ell-1})$, and $\sigma(0) = 0$, so Theorem C.1 in its Frobenius form, applied to the pre-activations $W_\ell Z_{\ell-1}$ and 0, gives $\|Z_\ell\|_F = \|\sigma(W_\ell Z_{\ell-1}) - \sigma(0)\|_F \le \|W_\ell Z_{\ell-1}\|_F \le \|W_\ell\|_2 \|Z_{\ell-1}\|_F$. At $\ell = L$ there is no rectifier and $H_L = W_L Z_{L-1}$, so $\|H_L\|_F \le \|W_L\|_2 \|Z_{L-1}\|_F$ directly. Chaining the first inequality from $Z_{\ell-1}$ down to $Z_0 = X$ gives the first bound of Eq. (21); chaining it from $Z_{\ell-1}$ up to Z_{L-1} and appending the second gives the other. For the last claim, the second bound of Eq. (21) divided by $\|W_\ell\|_2 \neq 0$ gives $\left(\prod_{j>\ell} \|W_j\|_2\right) \|Z_{\ell-1}\|_F \ge \|H_L\|_F / \|W_\ell\|_2$; multiplying by $\|W_\ell\|_F$, the summand of index ℓ in Eq. (20) is at least $\|W_\ell\|_F \|H_L\|_F / \|W_\ell\|_2$, and summing over ℓ and dividing by $\|H_L\|_F \neq 0$ gives the claim. □

The following theorem states the guarantee of the sketched forward pass, measured by κ and by the stable-rank profile ξ.

Theorem C.4 (Approximate batched evaluation)**.** *Let $H_L \neq 0$, let every $W_\ell \neq 0$, and let $\varepsilon \le 1$ and $\delta \in (0, 1/2)$; write $\xi = \sum_\ell \|W_\ell\|_F / \|W_\ell\|_2$ for the total stable-rank profile of the weights. Applying exact rectifiers to sketched pre-activations with sketch dimension $s = O\left(\left(\kappa^2\varepsilon^{-2} + \xi^2\right)\log(L/\delta)\right)$ yields, with probability at least $1-\delta$, the bound $\left\|\tilde{H}_L - H_L\right\|_F \le \varepsilon \|H_L\|_F$, in time*

$$O\left(\left(LdB + Ld^2\right)\left(\frac{\kappa^2}{\varepsilon^2} + \xi^2\right)\right) \tag{22}$$

for uniform width, the second term a one-time precomputation and the $\log(L/\delta)$ suppressed. For $B \ge d$ this is $O\left(LdB\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$, a factor of $\Theta\left(d/\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$ below the pointwise $\Theta\left(BLd^2\right)$, which, along a sequence of such networks, batches and accuracies with $d \to \infty$, is a saving when $\kappa^2/\varepsilon^2 + \xi^2 = o(d)$, a loss when it is $\omega(d)$, and a constant factor either way when it is $\Theta(d)$. When the biases vanish the second term is free: $\kappa \ge \xi$, so $\kappa^2/\varepsilon^2 \ge \xi^2$ at $\varepsilon \le 1$ and the sketch dimension is the $O\left(\kappa^2\varepsilon^{-2}\log(L/\delta)\right)$ that conditioning alone dictates. The sketch dimension is a parameter of the algorithm rather than a quantity it must know in advance: run with any $s \ge 5C\xi^2 \log(L/\delta)$, where C is the constant of Theorem C.2, and with probability at least $1-\delta$ the relative error is $O\left(\kappa\sqrt{\log(L/\delta)/s}\right)$.

Proof. We bound the layer errors by a recursion in ℓ and unroll it. Let $E_\ell = \left\| \tilde{Z}_\ell - Z_\ell \right\|_F$ for $\ell < L$ and $E_L = \left\| \tilde{H}_L - H_L \right\|_F$, with $E_0 = 0$. Given the sketch dimension s, set $\varepsilon_{\mathrm{s}} = \sqrt{C \log(L/\delta)/s}$, where C is the constant of Theorem C.2: this is the accuracy the Fact affords at dimension s, and $s = C\varepsilon_{\mathrm{s}}^{-2} \log(L/\delta)$ exactly. Every demand the argument makes on s is a demand on ε_{s}, and each is made at the line that forces it. Condition on every layer's sketch succeeding, which by a union bound has probability at least $1 - \delta$ once each succeeds with probability $1 - \delta/L$, that is, by Theorem C.2, once $s \geq C\varepsilon_{\mathrm{s}}^{-2} \log(L/\delta)$, which our s satisfies with equality. The Fact asks for δ/L and ε_{s} in $(0, 1/2)$: the first holds because $\delta < 1/2$, and the second is a demand on s, met where the demands are collected below.

At layer ℓ, add and subtract $W_\ell \tilde{Z}_{\ell-1}$, not $(W_\ell S_\ell)\left(S_\ell^\top Z_{\ell-1}\right)$: by Theorem C.1,

$$\begin{aligned} E_\ell &\leq \left\| W_\ell \tilde{Z}_{\ell-1} - W_\ell Z_{\ell-1} \right\|_F + \left\| (W_\ell S_\ell)\left(S_\ell^\top \tilde{Z}_{\ell-1}\right) - W_\ell \tilde{Z}_{\ell-1} \right\|_F \\ &\leq \|W_\ell\|_2 \, E_{\ell-1} + \varepsilon_{\mathrm{s}} \|W_\ell\|_F \left\| \tilde{Z}_{\ell-1} \right\|_F , \end{aligned} \tag{23}$$

the second term by Theorem C.2, applied conditionally on $S_1, \ldots, S_{\ell-1}$: given them, $\tilde{Z}_{\ell-1}$ is a fixed matrix, being a function of them and of the network and batch alone, while S_ℓ, drawn independently of them, still has independent $N(0, 1/s)$ entries; so the Fact's bound fails with conditional probability at most δ/L, hence with probability at most δ/L outright. The order of the two subtractions matters. Sketching the exact $Z_{\ell-1}$ instead leaves the operator $\left\| W_\ell S_\ell S_\ell^\top \right\|_2$ in the first term, and that is not bounded by $\|W_\ell\|_2$. For $W_\ell = I_d$ it is $\left\| S_\ell S_\ell^\top \right\|_2 = \|S_\ell\|_2^2 \geq \|S_\ell\|_F^2 / s$, since $S_\ell S_\ell^\top$ has rank at most s, and $\mathbb{E}\left[\|S_\ell\|_F^2 \right] = d$ for a sketch with $N(0, 1/s)$ entries; so the factor is at least d/s in expectation, a hundred at $d = 1000$ and $s = 10$, against $\|I_d\|_2 = 1$, and the difference compounds over L layers. Using $\left\| \tilde{Z}_{\ell-1} \right\|_F \leq \|Z_{\ell-1}\|_F + E_{\ell-1}$,

$$E_\ell \leq \left(\|W_\ell\|_2 + \varepsilon_{\mathrm{s}} \|W_\ell\|_F \right) E_{\ell-1} + \varepsilon_{\mathrm{s}} \|W_\ell\|_F \, \|Z_{\ell-1}\|_F . \tag{24}$$

The unrolling below needs $\varepsilon_{\mathrm{s}}\xi \leq 1$, that is $s \geq C\xi^2 \log(L/\delta)$, and this is what the ξ^2 term in the sketch dimension is for. The requirement $\varepsilon_{\mathrm{s}}\xi \leq 1$ is not implied by $s = \Omega\left(\kappa^2 \varepsilon^{-2} \log(L/\delta)\right)$, and no enlargement of that by a constant factor secures it: κ is a property of the batch as well as the weights, and with $L = 5$, $d = 100$, $W_\ell = I$, vanishing biases except a large constant b_L, and $X = e_1$, the quantity κ tends to 0 as b_L grows while $\xi = L\sqrt{d}$ stays fixed, so the two requirements are unrelated in general. With vanishing biases, by contrast, the requirement $\varepsilon_{\mathrm{s}}\xi \leq 1$ follows from the other one: Theorem C.3 gives $\kappa \geq \xi$, so at $\varepsilon \leq 1$ the demand $s \geq e^2 C \kappa^2 \varepsilon^{-2} \log(L/\delta)$ made below already gives $s \geq e^2 C \xi^2 \log(L/\delta)$. Granting $\varepsilon_{\mathrm{s}}\xi \leq 1$, we have $\prod_{j>\ell} \left(\|W_j\|_2 + \varepsilon_{\mathrm{s}} \|W_j\|_F \right) \leq e \prod_{j>\ell} \|W_j\|_2$: since every $W_j \neq 0$, factor $\|W_j\|_2$ out of each term, and then $\prod_{j>\ell} \left(1 + \varepsilon_{\mathrm{s}} \|W_j\|_F / \|W_j\|_2\right) \leq \exp\left(\varepsilon_{\mathrm{s}} \sum_{j>\ell} \|W_j\|_F / \|W_j\|_2\right) \leq \exp(\varepsilon_{\mathrm{s}}\xi) \leq e$ by $1 + x \leq e^x$. Unrolling Eq. (24) from $E_0 = 0$, that is $E_L \leq \sum_\ell \left(\prod_{j>\ell} \left(\|W_j\|_2 + \varepsilon_{\mathrm{s}} \|W_j\|_F \right) \right) \varepsilon_{\mathrm{s}} \|W_\ell\|_F \, \|Z_{\ell-1}\|_F$, gives

$$E_L \leq e\varepsilon_{\mathrm{s}} \sum_\ell \left(\prod_{j>\ell} \|W_j\|_2 \right) \|W_\ell\|_F \, \|Z_{\ell-1}\|_F = e\varepsilon_{\mathrm{s}} \kappa \, \|H_L\|_F . \tag{25}$$

We now collect the two demands on s. Take any $s \geq 5C\xi^2 \log(L/\delta)$. Then $\varepsilon_{\mathrm{s}} \leq 1/(\sqrt{5}\xi)$. The right side is at most $1/\xi$, which is the unrolling's requirement. It is also below $1/2$, which is the Fact's, because $\xi \geq L \geq 1$, as $\|W\|_F \geq \|W\|_2$ for every $W \neq 0$. So Eq. (25) holds with probability at least $1 - \delta$ and reads $E_L \leq e\kappa\sqrt{C \log(L/\delta)/s} \, \|H_L\|_F$, which is the last claim of the statement. For the error bound $E_L \leq \varepsilon \|H_L\|_F$ it remains to make that right side at most $\varepsilon \|H_L\|_F$, that is $s \geq e^2 C \kappa^2 \varepsilon^{-2} \log(L/\delta)$; so any $s \geq C \max\left(5\xi^2, e^2\kappa^2\varepsilon^{-2}\right) \log(L/\delta)$ serves, and this is $O\left(\left(\kappa^2\varepsilon^{-2} + \xi^2\right) \log(L/\delta)\right)$. With vanishing biases, $\xi \leq \kappa$ (Theorem C.3) and $\varepsilon \leq 1$ give $5\xi^2 \leq 5\kappa^2\varepsilon^{-2} \leq e^2\kappa^2\varepsilon^{-2}$, so the sketch dimension is $O\left(\kappa^2\varepsilon^{-2} \log(L/\delta)\right)$. For the cost, per layer $S_\ell^\top Z_{\ell-1}$ costs $O(d_{\ell-1} s B)$ and $(W_\ell S_\ell)\left(S_\ell^\top Z_{\ell-1}\right)$ costs $O(d_\ell s B)$, while precomputing $W_\ell S_\ell$ costs $O(d_\ell d_{\ell-1} s)$; summing over layers and substituting s gives Eq. (22). □

C.3 The Real Obstacle, and What Remains

Theorem C.4 located the cost in the conditioning κ and the profile ξ rather than in sign flips, and the following remark says what that means for which networks are easy.

Remark C.5 (Conditioning, not signs). The speedup is governed by the conditioning κ and, unless the biases vanish, by the stable-rank profile ξ, not by sign errors, and it is worth being exact about which networks have small κ, because the obvious guess is wrong. Norm preservation is not the condition: an orthogonal W_ℓ has $\|W_\ell\|_2 = 1$ but $\|W_\ell\|_F = \sqrt{d}$ exactly, so with vanishing biases the bound $\kappa \geq \sum_\ell \|W_\ell\|_F / \|W_\ell\|_2$ of Theorem C.3 reads $\kappa \geq L\sqrt{d}$, at which $s \geq \kappa^2/\varepsilon^2 \gg d$ and sketching is a loss. The hypothesis on the biases is needed for this lower bound on κ, which is the direction that matters here: a large output bias makes $\|H_L\|_F$ as large as one likes and drives κ to zero without changing a single weight. What controls κ is a small ratio $\|W_\ell\|_F / \|W_\ell\|_2$, that is low stable rank, which is incompatible with norm preservation; Theorem C.6 makes this precise.

The open questions are correspondingly relocated, from sign errors to conditioning. First, characterize the networks and inputs with small κ, and in particular decide whether training, or explicit spectral control, provably bounds it. Second, reduce the κ^2/ε^2 dependence, for instance by iterative refinement that sketches the residual of a first pass. Third, and sharpest, decide whether large κ is a real barrier — whether some family of ill-conditioned networks admits no approximate batched speedup at all.

We note finally that predicting a point's pattern from a nearby point's is also stable, since a mispredicted sign can occur only near a boundary, where the two affine pieces of the continuous map f agree; but its error across depth is more delicate to track, and the sketched forward pass is the route that carries an arbitrary-batch guarantee depending on conditioning alone.

The first and the third of these questions admit partial answers; the second is taken up after them.

Proposition C.6 (Low stable rank controls the conditioning)**.** *Suppose every layer satisfies $\|W_\ell\|_2 \leq 1$ and has stable rank $\|W_\ell\|_F^2 / \|W_\ell\|_2^2 \leq r$, the biases vanish, and the output does not collapse, $\|H_L\|_F \geq c\,\|X\|_F$ for some $c > 0$, with $X \neq 0$ so that κ is defined. Then $\kappa \leq L\sqrt{r}/c$. Consequently, along every sequence of such networks of uniform width, batches with $B \geq d$ and accuracies $\varepsilon \leq 1$, the sketched evaluation of Theorem C.4, with its $\log(L/\delta)$ suppressed as there, beats pointwise whenever $r = o\left(c^2\varepsilon^2 d/L^2\right)$ along the sequence. The hypothesis on the biases is not cosmetic: with $L = 3$, scalar layers $W_\ell = [1]$ and biases $(100, -100, 0)$ at $x = 1$, every other hypothesis holds with $r = c = 1$, yet $\kappa = 103$, against the $L\sqrt{r}/c = 3$ that the bound would give.*

Proof. In each summand of Eq. (20) the operator-norm product $\prod_{j>\ell} \|W_j\|_2$ is at most one, and $\|W_\ell\|_F \leq \sqrt{r}\,\|W_\ell\|_2 \leq \sqrt{r}$ because the stable rank of W_ℓ is at most r and $\|W_\ell\|_2 \leq 1$. With the biases vanishing, the first bound of Eq. (21) gives $\|Z_{\ell-1}\|_F \leq \|X\|_F$ for every ℓ, as every $\|W_j\|_2 \leq 1$. The sum is therefore at most $L\sqrt{r}\,\|X\|_F$, and dividing by $\|H_L\|_F \geq c\,\|X\|_F$ gives $\kappa \leq L\sqrt{r}/c$. Theorem C.3 also gives $\kappa \geq \xi$ for the ξ of Theorem C.4; the hypotheses $H_L \neq 0$ and $W_\ell \neq 0$, which the lemma and the theorem share, hold because $c > 0$ with $X \neq 0$ and because the stable rank of W_ℓ is defined. So the sketch dimension of Theorem C.4, with the $\log(L/\delta)$ suppressed, is $s = O\left(\kappa^2/\varepsilon^2\right) = O\left(L^2 r/(c^2\varepsilon^2)\right)$. At $B \geq d$ the precomputation $O\left(Ld^2 s\right)$ is dominated by the query term, and the cost $O\left(BLds\right)$ falls below the pointwise $O\left(BLd^2\right)$ once $s = o(d)$, that is $r = o\left(c^2\varepsilon^2 d/L^2\right)$. The restriction $B \geq d$ is not cosmetic either: at $B = 1$ the precomputation alone exceeds a single pointwise evaluation for every $s > 1$, whatever the stable rank. □

The proposition gives a sufficient condition for Theorem C.4 to save; with vanishing biases, low stable rank is also necessary for it, in total over the layers.

Corollary C.7 (Low total stable rank is necessary for the bound to save)**.** *Suppose the biases vanish, $H_L \neq 0$, every $W_\ell \neq 0$, and $\varepsilon > 0$. Then*

$$\frac{\kappa^2}{\varepsilon^2} + \xi^2 \geq \frac{\kappa^2}{\varepsilon^2} \geq \frac{\xi^2}{\varepsilon^2} \geq \frac{1}{\varepsilon^2} \sum_{\ell=1}^{L} \frac{\|W_\ell\|_F^2}{\|W_\ell\|_2^2}. \tag{26}$$

Consequently, along every sequence of networks of uniform width, batches with $B \geq d$ and accuracies $\varepsilon \leq 1$ that satisfy these hypotheses, the factor $\Theta\left(d/\left(\kappa^2/\varepsilon^2 + \xi^2\right)\right)$ by which the cost of Theorem C.4 falls below pointwise evaluation grows without bound only if the total stable rank $\sum_\ell \|W_\ell\|_F^2 / \|W_\ell\|_2^2$ is $o\left(\varepsilon^2 d\right)$, and it tends to 0 whenever that total is $\omega\left(\varepsilon^2 d\right)$. Without the hypothesis that the biases vanish, the other hypotheses kept, the factor still grows without bound only if the total stable rank is $o(d)$.

Proof. The first inequality of Eq. (26) holds because $\xi^2 \geq 0$. The second is the bound $\kappa \geq \xi$ of Theorem C.3, whose hypotheses are the ones assumed here, squared, both sides being non-negative, and divided by ε^2. For the third, write $a_\ell = \|W_\ell\|_F / \|W_\ell\|_2 > 0$, so that $\xi = \sum_\ell a_\ell$ and a_ℓ^2 is the stable rank of W_ℓ. Expanding the square, $\xi^2 = \sum_\ell a_\ell^2 + \sum_{\ell \neq m} a_\ell a_m \geq \sum_\ell a_\ell^2$, since every cross term $a_\ell a_m$ is positive; dividing by ε^2 gives the third inequality. For the consequences, write $K = \kappa^2/\varepsilon^2 + \xi^2$ and $\varrho = \sum_\ell \|W_\ell\|_F^2 / \|W_\ell\|_2^2$, so that Eq. (26) reads $\varrho \leq \varepsilon^2 K$. By Theorem C.4 at $B \geq d$ the cost falls below pointwise by a factor $\Theta(d/K)$, which grows without bound exactly when $K/d \to 0$ and tends to 0 exactly when $K/d \to \infty$. If $K/d \to 0$, then $\varrho/(\varepsilon^2 d) \leq K/d \to 0$, that is $\varrho = o\left(\varepsilon^2 d\right)$. If $\varrho/(\varepsilon^2 d) \to \infty$, then $K/d \geq \varrho/(\varepsilon^2 d) \to \infty$, and the factor tends to 0. Without vanishing biases the bound $\kappa \geq \xi$ is not available, but $K \geq \xi^2 \geq \varrho$ still holds, by the expansion of the square above, which uses only $W_\ell \neq 0$. The factor of Theorem C.4 is $\Theta(d/K)$ whatever the biases, since that theorem does not assume they vanish. So if the factor grows without bound, then $K/d \to 0$ and $\varrho/d \leq K/d \to 0$, that is $\varrho = o(d)$. □

With Theorem C.6 this brackets the saving. Per-layer stable rank $r = o\left(c^2\varepsilon^2 d/L^2\right)$ suffices, and it makes the total at most $Lr = o\left(c^2\varepsilon^2 d/L\right)$, which is $o\left(\varepsilon^2 d\right)$ because $L \geq 1$ and $c \leq 1$ there, the latter by the second bound of Eq. (21) at $\ell = 1$, which gives $\|H_L\|_F \leq \|X\|_F$ when every $\|W_j\|_2 \leq 1$. Total stable rank $o\left(\varepsilon^2 d\right)$ is necessary. The corollary concerns the bound of Theorem C.4 and not the sketch itself, which Theorem C.10 addresses, nor any other method.

Proposition C.8 (The condition number is attained)**.** *For every $a > 1$ there is a one-hidden-layer network on a scalar input whose forward condition number is $\kappa = \Theta(a)$ and which, for every $\varepsilon_s > 0$, admits a perturbation δZ of its hidden activations with $\|\delta Z\|_F = \varepsilon_s \|Z\|_F$ whose effect on the output is $\|\delta H_L\|_F = \Theta(\varepsilon_s \kappa) \|H_L\|_F$, the constants uniform in a and ε_s. Hence no analysis that controls a layer-wise approximation only through $\left\|\tilde{Z}_\ell - Z_\ell\right\|_F / \|Z_\ell\|_F$ can promise a relative output error better than $\Theta(\varepsilon_s \kappa)$; in particular the bound $E_L \leq e\varepsilon_s \kappa \|H_L\|_F$ of Eq. (25), which is all the proof of Theorem C.4 extracts from its layer-wise errors, is tight up to a constant on this family, and the factor κ cannot be removed from that analysis.*

Proof. On a scalar input x, let the network compute $f(x) = \sigma(ax) - \sigma((a-1)x)$, a single hidden layer of two units and no biases, so that $W_1 = (a, a-1)^\top$, $b_1 = 0$ and $W_2 = (1, -1)$; on $x \geq 0$ it computes $f(x) = x$ as the difference of the two activations ax and $(a-1)x$. At $x = 1$ with $a > 1$ both units are active, $Z = (a, a-1)^\top = W_1$ and $H_L = 1$. By Eq. (20), since $\|W_1\|_F = \|Z\|_F$ and $\|W_2\|_2 = \|W_2\|_F = \sqrt{2}$,

$$\kappa = \|W_2\|_2 \|W_1\|_F \|Z_0\|_F + \|W_2\|_F \|Z_1\|_F = 2\sqrt{2}\, \|Z\|_F .$$

For every $a > 1$, the quantity $\|Z\|_F^2 = 2a^2 - 2a + 1$ lies between $a^2/2$ and $2a^2$. So κ lies between $2a$ and $4a$, and is $\Theta(a)$. Take $\delta Z = \varepsilon_s \|Z\|_F e_1$, whose norm is exactly $\varepsilon_s \|Z\|_F$. Then $\|\delta H_L\|_F = |W_2 \delta Z| = \varepsilon_s \|Z\|_F = \varepsilon_s \kappa/(2\sqrt{2})$ while $\|H_L\|_F = 1$; the constant is exact, and uniform in a and ε_s. □

The proposition is about the analysis, not about any particular estimator, and the distinction matters here. On this family the input is scalar, so the layer-1 sketch $S_1 \in \mathbb{R}^{1\times s}$ acts as the scalar $S_1 S_1^\top = \|S_1\|^2$: the sketched hidden activation is $\|S_1\|^2 Z$, both units staying active because $\|S_1\|^2 \geq 0$, and when the output layer is computed exactly the output error is $\left|\|S_1\|^2 - 1\right| |H_L|$, a relative error that κ does not amplify. Computing the output layer exactly is the natural choice, as the hidden width is 2 and at $s \geq 2$ the exact product costs no more than the sketched one; run that way, sketching on this family is pointless rather than harmful. What the family shows is that the output is not stable

to a general perturbation of the hidden activations of relative size ε_{s}, which is all that the proof of Theorem C.4 knows about $\tilde{Z}_\ell$. Whether the approximate-matrix-multiplication estimator itself is forced to dimension $\Omega\left(\kappa^2/\varepsilon^2\right)$ is a separate question. Theorem C.10 answers it for one family: there $L = 1$, the bias vanishes and $\kappa = \|W\|_F \|Z\|_F / \|WZ\|_F = \sqrt{d}$, so the dimension it forces, $(d+1)/\varepsilon^2$, is $(\kappa^2+1)/\varepsilon^2$.

Together with Theorem C.6 the picture is that the sketch wins when the layers have low stable rank and the output does not collapse. For the bound of Theorem C.4 with vanishing biases, the stable-rank half of the converse holds in total over the layers, by Theorem C.7. For evaluation in general no converse is claimed, since Theorems C.8 and C.10 speak against one analysis and one estimator rather than against every method. The two conditions are independent: low stable rank bounds how much error the sketch introduces, and the non-collapse hypothesis $\|H_L\|_F \geq c\,\|X\|_F$ bounds how much of it survives to the output.

The remaining question, the second of Theorem C.5, is whether the ε^{-2} dependence itself can be shaved. For weights with a decaying spectrum, the deterministic counterpart of low stable rank, it can be, and by an exact truncation rather than a random sketch. The rank-r truncation $W^{(r)}$ of a matrix W keeps the r largest singular values of its singular value decomposition; it is the best rank-r approximation of W in spectral norm (Eckart & Young, 1936; Mirsky, 1960), though the proof below uses only the identity $\left\|W - W^{(r)}\right\|_2 = \sigma_{r+1}(W)$, which it reads off the decomposition.

Proposition C.9 (Deterministic low-rank evaluation for decaying spectra)**.** *Suppose every layer ℓ satisfies $\|W_\ell\|_2 \leq 1$ with geometrically decaying singular values $\sigma_i(W_\ell) \leq \eta^{i-1}$ for some $\eta < 1$ bounded away from 1, and that its activations do not grow, $\|Z_\ell\|_F \leq \|X\|_F$; suppose also that the output does not collapse, $\|H_L\|_F \geq c\,\|X\|_F$ for some $c > 0$.*

Replacing each W_ℓ by its rank-r truncation $W_\ell^{(r)}$ and applying exact rectifiers evaluates the batch to relative Frobenius error ε with

$$r = O\left(\frac{\log\left(L/(c\varepsilon)\right)}{\log(1/\eta)}\right) = O\left(\log\left(L/(c\varepsilon)\right)\right), \tag{27}$$

in time $O\left(LdBr\right)$ after a one-time singular value decomposition of the weights, a data-independent precomputation of $O\left(Ld^3\right)$ floating-point operations that lies outside the model of Section 2.1, which has no square root; this replaces the ε^{-2} dependence of Theorem C.4 by $\log(1/\varepsilon)$. The rank is $O\left(\log(1/\varepsilon)\right)$ outright when $L/c \leq \varepsilon^{-O(1)}$, and not in general: at $\varepsilon = 1/2$ and $c = 1$ it grows as $\log L$. The constant hidden in Eq. (27) is $1/\log(1/\eta)$, which is $\Theta(d)$ at $\eta = 1 - 1/d$; the hypothesis that η is bounded away from 1 is what makes the saving real.

Proof. Let Z_ℓ and $\widetilde{Z}_\ell$ be the exact and truncated activations, with $Z_0 = \widetilde{Z}_0 = X$, and set $E_\ell = \left\|\widetilde{Z}_\ell - Z_\ell\right\|_F$ for $\ell < L$ and $E_L = \left\|\widetilde{H}_L - H_L\right\|_F$. Since σ is 1-Lipschitz in the Frobenius norm (Theorem C.1), for $\ell < L$

$$E_\ell = \left\|\sigma\left(W_\ell^{(r)}\widetilde{Z}_{\ell-1} + b_\ell \mathbf{1}^\top\right) - \sigma\left(W_\ell Z_{\ell-1} + b_\ell \mathbf{1}^\top\right)\right\|_F \leq \left\|W_\ell^{(r)}\widetilde{Z}_{\ell-1} - W_\ell Z_{\ell-1}\right\|_F, \tag{28}$$

and at $\ell = L$, where there is no rectifier, the same bound holds with equality. Adding and subtracting $W_\ell^{(r)} Z_{\ell-1}$ and applying the triangle inequality, with $\left\|W_\ell^{(r)}\right\|_2 = \sigma_1(W_\ell) \leq 1$ and $\left\|W_\ell^{(r)} - W_\ell\right\|_2 = \sigma_{r+1}(W_\ell) \leq \eta^r$, gives

$$E_\ell \leq \left\|W_\ell^{(r)}\right\|_2 E_{\ell-1} + \left\|W_\ell^{(r)} - W_\ell\right\|_2 \|Z_{\ell-1}\|_F \leq E_{\ell-1} + \eta^r \|X\|_F,$$

using the hypothesis $\|Z_{\ell-1}\|_F \leq \|X\|_F$. The two spectral norms $\left\|W_\ell^{(r)}\right\|_2$ and $\left\|W_\ell^{(r)} - W_\ell\right\|_2$ are read off the singular value decomposition $W_\ell = \sum_i \sigma_i(W_\ell) u_i v_i^\top$: the truncation keeps the terms with $i \leq r$, so $W_\ell^{(r)}$ has largest singular value $\sigma_1(W_\ell)$, and $W_\ell - W_\ell^{(r)} = \sum_{i>r} \sigma_i(W_\ell) u_i v_i^\top$ is itself a singular value decomposition with largest singular value $\sigma_{r+1}(W_\ell)$, which is at most η^r by the decay hypothesis. Unrolling from $E_0 = 0$, $E_L \leq L\eta^r \|X\|_F$, so the relative error is $E_L / \|H_L\|_F \leq L\eta^r/c$, which is at most ε once $\eta^r \leq c\varepsilon/L$, that is once $r \geq \log\left(L/(c\varepsilon)\right) / \log(1/\eta)$, which is

Eq. (27). Writing $W_\ell^{(r)} = U_\ell \Sigma_\ell V_\ell^\top$ and associating as $U_\ell \left(\Sigma_\ell \left(V_\ell^\top \widetilde{Z}_{\ell-1}\right)\right)$ applies the map to the $d \times B$ activation in $O(rdB)$, for $O(LrdB)$ over the network; the truncations are computed once, from the weights alone, by a singular value decomposition, in $O\left(Ld^3\right)$ floating-point operations outside the model of Section 2.1, which has no square root. □

The decaying case is thus settled. Two residues remain from the discussion after Theorem C.8: whether the sketch itself is forced to large dimension on a flat spectrum, and whether any bound holds against every method. We settle the first and delimit the second.

Proposition C.10 (Sketching cannot beat a flat spectrum)**.** *Let $W \in \mathbb{R}^{d\times d}$ be a square layer with a flat spectrum, $\sigma_i(W) = \varsigma$ for every $i \in [d]$, so that $\|W\|_F = \varsigma\sqrt{d}$ and $\|WZ\|_F = \varsigma \|Z\|_F$ for every Z. Then there is a batch on which the Gaussian approximate-matrix-multiplication sketch of Theorem C.4 attains expected squared relative Frobenius error ε^2 only with sketch dimension $s \geq (d+1)/\varepsilon^2$, which is at least d for every $\varepsilon \leq 1$, at which point $(WS)\left(S^\top Z\right)$ costs no less than the direct product WZ. The same holds with high probability: if $d \geq 2$ and $\varepsilon < 1/2$, then on the same batch every sketch dimension $s < (d-1)/\left(4\varepsilon^2\right)$ attains relative Frobenius error at most ε with probability at most $8/(d-1)$. Sketching therefore gives no speedup on flat spectra.*

Proof. A flat spectrum means $W = \varsigma Q$ with Q orthogonal, so $\|WZ\|_F = \varsigma \|Z\|_F$. Take the batch whose first column is e_1 and whose others are zero, so $\|Z\|_F = 1$ and $\|WZ\|_F = \varsigma$, and let $S \in \mathbb{R}^{d\times s}$ have independent $N(0, 1/s)$ entries. Writing $w = S^\top e_1 \in \mathbb{R}^s$ for the first row of S, the estimator's error on that column is $\varsigma \|Sw - e_1\|$, since Q is orthogonal. The first coordinate is $(Sw)_1 = \|w\|^2$, of mean 1 and variance $2/s$. For $i \neq 1$, the coordinate $(Sw)_i = \sum_k S_{ik}S_{1k}$ has mean 0 and variance $1/s$. Hence

$$\mathbb{E}\left[\left\|(WS)\left(S^\top Z\right) - WZ\right\|_F^2\right] = \varsigma^2\left(\frac{2}{s} + \frac{d-1}{s}\right) = \frac{(d+1)\varsigma^2}{s},$$

and an expected squared relative error $\varepsilon^2 \|WZ\|_F^2 = \varepsilon^2\varsigma^2$ forces $s \geq (d+1)/\varepsilon^2$, which is at least d for $\varepsilon \leq 1$. Forming $WS \in \mathbb{R}^{d\times s}$ and multiplying by $S^\top Z \in \mathbb{R}^{s\times B}$ then costs $\Omega\left(dsB\right) = \Omega\left(d^2B\right)$, the direct product up to constants. The computation is done here rather than cited because Theorem C.2 and its sources give the matching upper bound, and a lower bound does not follow from it.

For the high-probability claim, condition on the first row w of S. The other $d-1$ rows of S are independent of w and of one another. Given w, the coordinates $(Sw)_i = \sum_k S_{ik}w_k$ with $i \neq 1$ are independent normal variables of mean 0 and variance $\|w\|^2/s$. Hence $\sum_{i\neq 1}(Sw)_i^2 = \left(\|w\|^2/s\right)\chi$, where, given $w \neq 0$, which holds with probability one, χ is distributed as the sum of the squares of $d-1$ independent standard normal variables, and the squared relative error on the batch is

$$\frac{\|Sw - e_1\|^2\varsigma^2}{\varsigma^2} = \left(\|w\|^2 - 1\right)^2 + \frac{\|w\|^2}{s}\chi.$$

If $\|w\|^2 < 1/2$, the first term exceeds $1/4$, which exceeds ε^2 because $\varepsilon < 1/2$. If $\|w\|^2 \geq 1/2$, the second term is at least $\chi/(2s)$. When moreover $\chi > (d-1)/2$, this exceeds $(d-1)/(4s)$, which in turn exceeds ε^2 because $s < (d-1)/\left(4\varepsilon^2\right)$. So relative error at most ε forces $\chi \leq (d-1)/2$. Each square of a standard normal variable has mean 1 and variance $\mathbb{E}\left[g^4\right] - 1 = 2$, since $\mathbb{E}\left[g^4\right] = 3\mathbb{E}\left[g^2\right] = 3$ for a standard normal variable g, by integrating $y^3 \cdot ye^{-y^2/2}$ by parts. So χ has mean $d-1$ and variance $2(d-1)$, and by Chebyshev's inequality

$$\mathbf{Pr}\left[\chi \leq \frac{d-1}{2}\right] \leq \mathbf{Pr}\left[|\chi - (d-1)| \geq \frac{d-1}{2}\right] \leq \frac{2(d-1)}{(d-1)^2/4} = \frac{8}{d-1}.$$

Since $4\varepsilon^2 < 1$, the dimension $(d-1)/\left(4\varepsilon^2\right)$ exceeds $d-1$, so a sketch meeting the guarantee with probability above $8/(d-1)$ has $s \geq d$, and it costs no less than the direct product, as before. With Theorems C.6 and C.9, sketching and truncation help when the spectrum decays, and sketching does not help when it is flat. □

The last part of Theorem C.10 gives the high-probability form of its lower bound, with $(d-1)/\left(4\varepsilon^2\right)$ in place of $(d+1)/\varepsilon^2$: for $d \geq 2$ and $\varepsilon < 1/2$, a guarantee holding with probability above $8/(d-1)$ needs that dimension.

An unconditional lower bound against *every* method is a different matter. We can tie approximation to the hardness of labeling (Theorem B.9), but not exceed it.

Proposition C.11 (Approximation is Boolean-matrix-multiplication-hard)**.** *There is a family of two-layer networks and batches, on which every output coordinate has magnitude at least* 1*, such that for every relative error $\varepsilon < 1$, computing each output coordinate to relative error ε determines the Boolean product of an $n \times d$ by a $d \times n$ Boolean matrix. Approximate evaluation to entrywise relative error is therefore at least as hard as Boolean matrix multiplication, hence triangle-detection-hard by Theorem B.8, and inherits the growing-dimension wall of Theorem B.7.*

Proof. Let $A \in \{0,1\}^{n \times d}$ and $C \in \{0,1\}^{d \times n}$ be a Boolean-product instance. Take the two-layer network with $W_1 = A$, $b_1 = 0$, $W_2 = 2I$ and $b_2 = -\mathbf{1}$, and the batch $X = C$, whose columns we write $c_1, \ldots, c_n$. Since A and C are non-negative, $h_1(c_j) = Ac_j \geq 0$, so $z_1 = \sigma(h_1) = Ac_j$ and

$$f^i(c_j) = 2(AC)_{ij} - 1 \in \{-1, 1, 3, 5, \ldots\},$$

because $(AC)_{ij}$ is a non-negative integer. In particular $\left|f^i(c_j)\right| \geq 1$, and $f^i(c_j) > 0$ exactly when $(AC)_{ij} \geq 1$, that is exactly when $(A \odot C)_{ij} = 1$. Any $\tilde{f}^i(c_j)$ satisfying

$$\left|\tilde{f}^i(c_j) - f^i(c_j)\right| \leq \varepsilon \left|f^i(c_j)\right| < \left|f^i(c_j)\right|$$

carries the same sign, so an entrywise-ε approximation reveals $A \odot C$. The instance is built in $O(nd)$ time, so Theorem B.8 applies.

The shift by $\mathbf{1}$ is what keeps the task an approximate one. Without it the interesting entries are exactly the zeros of AC, and at $f = 0$ the guarantee $\left|\tilde{f} - f\right| \leq \varepsilon |f|$ forces $\tilde{f} = 0$ exactly; the reduction would then demand exact evaluation on those entries and so say nothing beyond Theorem B.9. The shift moves every output to magnitude at least 1, where a relative-error guarantee is a genuine tolerance, and the reduction survives it. The margin is at the output only: the hidden pre-activations $h_1^i(c_j) = (AC)_{ij}$ still vanish wherever $A \odot C$ does, so the family is not kept away from the unit boundaries. □

The hardness that Theorem C.11 yields is conditional, and the following remark says what an unconditional bound could not be.

Remark C.12 (The barrier to an unconditional bound). Theorem C.11 is method-independent, but it reduces to the hardness of labeling rather than exceeding it, and that hardness is conditional. It is worth saying what an unconditional bound could not be. Within the uniform-width regime with $B \geq d$, no family requires $\Omega(BT)$ approximate evaluation: Theorem B.3 (i) already evaluates every such network *exactly* in $O\left(LBd^{\omega-1+o(1)}\right) = o(BT)$ as $d \to \infty$. What is open is whether the κ dependence is necessary for every method, or only for the sketching route. Theorem C.8 shows the analysis of Theorem C.4 cannot shed it and Theorem C.10 shows the estimator cannot, but neither speaks against an evaluator built on a different principle. By Theorem C.11, an $n^{2+o(1)}$ algorithm for entrywise relative error on its square reduction family would place Boolean matrix multiplication in $n^{2+o(1)}$. That question remains open; removing κ from a bound without achieving this running time would not settle it. The reduction gives no unconditional lower bound on approximate evaluation.

D Proofs from Section 2

Proof of Theorem 2.3. Affine form. Fix x in the closure $\overline{\mathcal{R}_p}$. For $\ell < L$ and every coordinate j, the pre-activation satisfies $h_{\ell j}(x) \geq 0$ when $p_{\ell j} = 1$ and $h_{\ell j}(x) \leq 0$ when $p_{\ell j} = 0$: by Theorem 2.2 the inequality $h_{\ell j} > 0$, respectively $h_{\ell j} \leq 0$, holds on $\mathcal{R}_p$, each $h_{\ell j}$ is continuous as a composition of affine maps with the rectifier, and the closed sets $\{h_{\ell j} \geq 0\}$ and $\{h_{\ell j} \leq 0\}$ therefore contain the closure of $\mathcal{R}_p$ once they contain $\mathcal{R}_p$. So in both cases $z_\ell^j = \sigma(h_{\ell j}(x)) = p_{\ell j} h_{\ell j}(x)$, that is

$z_\ell = \Delta_\ell h_\ell(x)$ with $\Delta_\ell = \operatorname{diag}(p_\ell)$. We show by induction on ℓ that $h_\ell(x) = M_\ell x + v_\ell$ for the stated recursion. For $\ell = 1$ we have $h_1(x) = W_1 x + b_1 = M_1 x + v_1$. Assuming $h_{\ell-1}(x) = M_{\ell-1}x + v_{\ell-1}$,

$$\begin{aligned} h_\ell(x) &= W_\ell z_{\ell-1} + b_\ell = W_\ell \Delta_{\ell-1} h_{\ell-1}(x) + b_\ell \\ &= (W_\ell \Delta_{\ell-1} M_{\ell-1})\, x + (W_\ell \Delta_{\ell-1} v_{\ell-1} + b_\ell) = M_\ell x + v_\ell, \end{aligned} \tag{29}$$

which is the recursion. At $\ell = L$ this gives $f(x) = h_L(x) = M_L x + v_L$, so $A_p = M_L$ and $c_p = v_L$; unfolding the recursion for M_L yields $A_p = W_L \Delta_{L-1} W_{L-1} \cdots \Delta_1 W_1$, which is Eq. (3).

Cost. Each M_ℓ is a $d_\ell \times d_0$ matrix. Given $M_{\ell-1}$, forming $\Delta_{\ell-1} M_{\ell-1}$ scales the $d_{\ell-1}$ rows of a $d_{\ell-1} \times d_0$ matrix, at cost $O(d_{\ell-1} d_0)$; the product $W_\ell (\Delta_{\ell-1} M_{\ell-1})$, of shapes $d_\ell \times d_{\ell-1}$ and $d_{\ell-1} \times d_0$, costs $O(d_\ell d_{\ell-1} d_0)$ and dominates; the vector step $v_\ell = W_\ell \Delta_{\ell-1} v_{\ell-1} + b_\ell$ costs $O(d_\ell d_{\ell-1})$. Summing over layers,

$$\sum_{\ell=1}^{L} O(d_\ell d_{\ell-1} d_0) = O\left(d_0 \sum_{\ell=1}^{L} d_\ell d_{\ell-1} \right) = O(d_0 T). \tag{30}$$

Applying $(A_p, c_p) = (M_L, v_L)$ to one input costs $O(d_L d_0)$ for the product $A_p x$ and $O(d_L)$ for the sum, hence $O(d_0 d_L)$, the per-input figure used in Theorem 3.1. □

Proof of Theorem 2.4. Write C_p for the right-hand side of Eq. (4). The inclusion $\mathcal{R}_p \subseteq C_p$ is Theorem 2.3: on $\mathcal{R}_p$ the pre-activations h_ℓ agree with the forms $\tilde{h}_\ell$, and by Theorem 2.2 they carry the signs p prescribes.

For $C_p \subseteq \mathcal{R}_p$, let $x \in C_p$; we show by induction on ℓ that $h_\ell(x) = \tilde{h}_\ell(x)$ and $\pi_\ell(x) = p_\ell$. For $\ell = 1$, $h_1(x) = W_1 x + b_1 = M_1 x + v_1 = \tilde{h}_1(x)$, and the sign conditions defining C_p give $\pi_1(x) = p_1$. Assume both hold up to $\ell - 1$. For each coordinate j, the value $p_{\ell-1,j} = 1$ forces $h_{\ell-1,j}(x) > 0$ and $p_{\ell-1,j} = 0$ forces $h_{\ell-1,j}(x) \le 0$, so in both cases $z^j_{\ell-1} = \sigma(h_{\ell-1,j}(x)) = p_{\ell-1,j} h_{\ell-1,j}(x)$, that is $z_{\ell-1} = \Delta_{\ell-1} \tilde{h}_{\ell-1}(x)$. Substituting into Eq. (1) and comparing with the recursion of Theorem 2.3, $h_\ell(x) = W_\ell \Delta_{\ell-1} (M_{\ell-1} x + v_{\ell-1}) + b_\ell = M_\ell x + v_\ell = \tilde{h}_\ell(x)$, and the sign conditions of C_p give $\pi_\ell(x) = p_\ell$. Collecting the equalities $\pi_\ell(x) = p_\ell$ for $\ell \le L - 1$ gives $\pi(x) = p$, so $x \in \mathcal{R}_p$.

The description Eq. (4) is an intersection of N open or closed half-spaces, one per hidden unit, hence convex.

For the closure, let $\overline{C}_p$ be the set obtained by relaxing every strict inequality of Eq. (4) to $\tilde{h}_{\ell j}(x) \ge 0$. It is closed and contains $C_p = \mathcal{R}_p$, so $\overline{\mathcal{R}_p} \subseteq \overline{C}_p$, and when $\mathcal{R}_p$ is nonempty the reverse inclusion holds too: for $x \in \mathcal{R}_p$ and $y \in \overline{C}_p$, each $\tilde{h}_{\ell j}$ is affine along the segment from x to y, so where $p_{\ell j} = 1$ it is positive at every point of the segment other than y, being positive at x and non-negative at y, and where $p_{\ell j} = 0$ it is non-positive on the whole segment; thus the segment without y lies in $\mathcal{R}_p$ and y is a limit of points of $\mathcal{R}_p$. So $\overline{\mathcal{R}_p} = \overline{C}_p$, a finite intersection of closed half-spaces (or of the whole space, where a form is constant), that is a polyhedron.

Every facet of a polyhedron given by linear inequalities is cut out by one of them (Schrijver, 1986, Theorem 8.1), here by the zero set of one $\tilde{h}_{\ell j}$. When $\mathcal{R}_p$ is empty its closure is empty and there is nothing to show. □

E Proof from Section 3

Proof of Theorem 3.2. Evaluation from location. Run an optimal point-location procedure to obtain all patterns at cost $\operatorname{Loc}(f, X)$, then apply Theorem 3.1 to the now-labeled batch at cost $O(BN + Rd_0T + d_0 d_L B)$; their sum is the first inequality.

Location at the baseline. A single forward pass at x costs $\Theta(T)$ by Eq. (2) and computes every pre-activation $h_\ell(x)$, from which the pattern is read off as $\pi_{\ell j}(x) = \mathbb{1}(h_{\ell j}(x) > 0)$ in N comparisons. Running one pass per input gives $\operatorname{Loc}(f, X) = O(BT + BN) = O(BT)$, using $N \le T$ as at Eq. (2). □

The same baseline comparison also explains why the rectifiers themselves are not the source of the batched cost.

Remark E.1 (The rectifier is free). The cost of all rectifiers in a batch is $O(BN)$, dominated by the affine cost $\Theta(BT)$ because $N \leq T$, as at Eq. (2). Every nontrivial saving therefore comes from the geometry of regions or from accelerating the affine maps, never from the rectifier itself.