EXPANDERS MEET EXPRESSIVITY: UNIVERSAL APPROXIMATION ON FIXED SPARSE MASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work on pruning and the lottery-ticket hypothesis suggests that the graph topology of a sparse subnetwork matters for its performance. Spectral expanders and Ramanujan graphs have emerged as promising masks: they combine sparsity with strong global connectivity, and empirical studies report high accuracy and, in some settings, robustness to noise, even at high sparsity. Whether such masks are justified by approximation theory has remained open. Prior results establish that sparse masks without a spectral gap can emulate ReLU networks with unbounded depth and signed weights. Can a fixed, constant-degree expander mask do so efficiently, with the connections responsible for expansion carrying the computation themselves? We answer affirmatively for ReLU networks. Using auxiliary Ramanujan graphs, we build for every admissible width a three-regular mask, independent of the target and repeated in every layer, whose connections both give it a width-independent spectral gap and route values through the network. Networks on these masks represent exactly every ReLU network, with width linear in the number of neurons and nonzero weights per layer and polylogarithmic depth overhead per layer. Expansion also makes this representation robust to pruning. In a block version of the mask built from a bipartite expander, exact representation survives the deletion of a fixed fraction of the connections between each pair of connected blocks in every layer, with the remaining weights chosen afterwards; a same-degree construction without expansion loses input dependence under an allowed pruning. We also separate spectral expansion from routing: a width-seven Ramanujan mask, stacked any number of times, cannot realise every permutation of its neurons by moving values along vertex-disjoint paths. Moreover, we give uniform approximation rates with explicit width, depth and accuracy, including constructions with bounded weights and activations. We also obtain encoding lengths of optimal order for Hölder classes with exponent at most one under a fixed decoder, extend the results to compositions of Lipschitz functions that each depend on few variables, and prove stability bounds under perturbation of all weights. These results show that expander topology is compatible with efficient exact representation, complementing the empirical success of expander and Ramanujan masks in pruning and lottery-ticket settings.

1 INTRODUCTION

In this article, we study how network sparsity affects expressivity, focusing on universal approximation. A universal approximation theorem asserts that every function in a specified class can be approximated to arbitrary accuracy by a suitable neural network. For continuous functions on a compact cube, the error is measured in the uniform norm. A foundational result in this area is Cybenko's universal approximation theorem, which showed that networks with a continuous sigmoidal activation can approximate any continuous function on the unit cube given sufficiently many units [8]. Later quantitative results bound the required network size in terms of the target and the accuracy [59, 6].In practice, however, implementations often impose additional structural constraints, such as sparsity or fixed connectivity patterns, which can significantly affect the network's expressive power. Sparse networks now arise routinely: by pruning trained models, by pruning at initialisation, and by training sparse networks from scratch [21, 5, 29, 58, 54, 37, 12]. The lottery-ticket hypothesis [13] posits that a dense network contains sparse subnetworks which, trained in

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.

isolation, match the accuracy of the full model. Its strong form [48] asserts that a sufficiently large random network contains subnetworks that approximate a given target without any training; this has since been proved under increasingly mild overparameterisation [35, 43, 40, 9, 7]. These results establish the *existence* of good target-dependent subnetworks inside larger dense networks. A complementary question is which fixed, target-independent connectivity patterns suffice for universal approximation.

A second line of work asks which *topological* properties distinguish good masks. Viewing each layer as a bipartite graph, Prabhu et al. [47] proposed expander-structured layers as efficient sparse architectures, and the relational-graph structure of networks has been linked to their performance [60]. Pal et al. [41] observed that winning tickets preserve the Ramanujan property layer by layer, and that accuracy and robustness to noise degrade as layers lose it; they also proposed Ramanujan-preserving pruning. Subsequent work refined the spectral criteria used to rank pruning-at-initialisation masks [20, 19], and to recurrent layers [23]. Expander and Ramanujan masks have also been used constructively, see for instance the works [52, 10, 4, 24, 57, 11, 2]. Well-chosen random masks are also competitive baselines [33, 14]. The previous results are empirical, and none of them addresses the approximation-theoretic question: can a fixed, target-independent, constant-degree expander mask, repeated in every layer, support universal approximation, and can its expanding connections themselves carry the computation? This question matters because the empirical studies above implicitly rely on expander masks not restricting what the network can represent. If they did, selecting masks by their spectral properties could exclude good solutions. **We give a positive answer**: expander topology is compatible with full expressivity, and the expanding connections can carry the computation.

1.1 Main results

Our main contributions are the following.

- **A fixed expander mask that represents ReLU networks exactly.** For every power of two $n \geq 32$, we construct a three-regular mask A_n whose spectral gap is bounded away from zero uniformly in n (Theorem 3.3); degree three is the smallest regular degree for which this is possible (Proposition B.4). Reused in every hidden layer, A_n represents exactly every ReLU network of width at most $n/2$ and depth T whose neurons have fan-in and fan-out at most two, with $1 + (T + 1)\, O(\log^2 n)$ hidden layers (Theorem 3.1), and general ReLU networks at polylogarithmic depth cost per layer (Corollary 3.2). Hence the constant mask sequence has the universal approximation property (Corollary 4.3).
- **Expansion and routing are distinct properties.** Our construction uses the same connections for a uniform spectral gap and routing. A width-seven Ramanujan mask, stacked any number of times, cannot route every permutation of its neurons along vertex-disjoint paths (Proposition 3.4).
- **Expansion provides redundancy against pruning.** In a block construction built from a bipartite expander, exact representation survives the deletion of a fixed fraction of the connections in every layer, provided the remaining weights are chosen after the deletions (Theorem 6.2); a same-degree construction without expansion admits a pruning that destroys input dependence (Proposition 6.3).

The remaining results are consequences or extensions of the first: explicit approximation rates (Theorem 4.2), constructions with bounded weights (Theorem 5.1), encoding lengths of optimal order for Hölder targets (Theorem 5.2), improved rates for compositional targets (Theorem E.2), and stability under weight perturbations (Theorem F.1).

1.2 Relation to earlier work

One line of work studies approximation complexity through the number of nonzero weights. Optimal rates are known under this measure [6, 59, 44], with connectivity chosen for the target. A complementary line studies constraints on width: minimum-width and deep-narrow results characterize what networks can represent under width constraints [34, 17, 25, 42, 26, 31, 55, 32].

A closer comparison is with work on universality under fixed sparsity patterns. Universal approximation has been established for convolutional networks [64, 45] and sparse attention architectures [61, 62], and under fixed indegree constraints for sufficiently deep sparse maxout networks [15]. These architectures have different degree and spectral properties: banded convolutional masks have

a spectral gap that vanishes with width, while global tokens in sparse attention have unbounded degree. Our masks can represent convolutional networks, as they can represent any ReLU network (Appendix G). The distinction here is that one fixed, constant-degree mask is reused at every layer, with a spectral gap bounded away from zero uniformly in the width.

The use of fixed wiring to implement many computations has a longer history in permutation networks and universal circuits [3, 56, 63]; see also [27] for a complexity-theoretic perspective on universal approximation. Beneš networks route every permutation in $O(\log n)$ layers of 2×2 switches. As masks, these layers are two-regular, below the minimum degree of three for a uniform spectral gap (Proposition B.4). Our construction adds one connection per neuron to a switch layer to obtain both routing and expansion. It can also be viewed as a directed variant of the replacement product [49], built on bipartite Ramanujan 2-lifts [36]. We route values using matchings, following the routing-number framework of Alon, Chung and Graham [1]. Nenadov [38] obtains $O(\log n)$ routing rounds under a stronger spectral condition; our construction gives an $O(\log^2 n)$ bound for its established spectral gap.

Our pruning result is related to several uses of expanders for robustness: error containment [51], fault-tolerant switching [46], reinforcement against random faults [30], and maintaining routability under deletions [16]. This line of work studies connectivity, fault tolerance, and routability; our theorem uses block-wise expansion to preserve exact representation of a fixed ReLU architecture under layer-wise deletions.

The approximation and composition results also build on established constructions. Our interpolation uses piecewise-linear constructions with ReLU networks [18, 22], while our compositional function class follows Schmidt-Hieber [50]. Our masks constrain the architecture itself. Sparse update masks [53] and dynamic sparse training [12] instead select trained or active weights as data are processed.

Organisation. Section 2 introduces the model, and Section 3 establishes exact representation. Sections 4–6 present the approximation results and robustness guarantees. Section 7 discusses limitations. Proofs are collected in the appendix.

2 Networks, masks, and the fixed-architecture question

Network model. Let $\mathrm{ReLU}(t) = \max\{t, 0\}$, applied coordinatewise. A network with L hidden layers of width n and scalar output computes

$$h^{(0)} = x, \qquad h^{(\ell)} = \mathrm{ReLU}(W_\ell h^{(\ell-1)} + b_\ell), \quad 1 \le \ell \le L, \qquad F(x) = a^\top h^{(L)} + c. \tag{2.1}$$

Depth counts hidden layers only. The first layer copies the input coordinates into prescribed neurons, and the output layer reads two prescribed neurons (Appendix A).

Masks and expansion. For $\ell \ge 2$, the connectivity of layer ℓ is given by a binary mask $A_\ell \in \{0,1\}^{n \times n}$. As for W_ℓ, rows index output neurons and columns index input neurons, and

$$(A_\ell)_{ji} = 0 \quad \Longrightarrow \quad (W_\ell)_{ji} = 0. \tag{2.2}$$

The mask specifies the *permitted connections*; the weight on a permitted connection may still be zero. The mask can therefore be an expander even when the nonzero weights used for a particular function form a much sparser graph. An m-regular mask has exactly m ones in each row and each column, so every neuron has fan-in and fan-out m. We normalise it as $P_\ell = \frac{1}{m} A_\ell$, which is doubly stochastic, and call its second largest singular value $\sigma_2(P_\ell)$ its *spectral parameter*. With $\mathbf{1}$ the all-ones vector, $J_n = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and $\|\cdot\|_2$ the spectral norm,

$$\sigma_2(P_\ell) = \|P_\ell - J_n\|_2 . \tag{2.3}$$

We use the same notation $\sigma_2(\frac{1}{m}B)$ for the biadjacency matrix B of any m-regular bipartite graph. A small value of $\sigma_2(P_\ell)$ means that the layer mixes information well across neurons. A family has a *uniform spectral gap*, and its members are *spectral expanders*, if $\sup_{n,\ell} \sigma_2(P_\ell) < 1$; this is a property of the masks, not of the weights. For a single mask, $\sigma_2(P_\ell) < 1$ is equivalent to connectivity of its bipartite graph. Expansion is a property of a family, and uniformity in the width

makes it nontrivial. A connected m-regular bipartite graph with biadjacency matrix A is *Ramanujan* if $\sigma_2(\frac{1}{m}A) \le \frac{2\sqrt{m-1}}{m}$, which is stronger than a uniform spectral gap.

We ask whether one fixed sequence of masks can support approximation of every continuous target, with only the depth and weights chosen for that target. The following definition formalizes this fixed-architecture question.

Definition 2.1 (Universal approximation with fixed masks)**.** For a fixed width n, let $\mathcal{A}_n = (A_2, A_3, \dots)$ be an infinite sequence of masks, with input and output layers as above. The sequence *has the universal approximation property* on $[0,1]^d$ if every continuous function can be approximated uniformly, to any accuracy, by a network whose connections lie in a finite prefix of the sequence:

$$\forall f \in C([0,1]^d)\ \forall \varepsilon > 0\ \exists L, \theta: \quad \|F_{\mathcal{A}_n, L, \theta} - f\|_\infty \le \varepsilon.$$

The masks do not depend on f or ε; only the depth L and the parameters θ do, and weights are not shared across layers.

Our exact-representation result is first proved for a restricted network form that makes the routing requirements explicit.

Definition 2.2 (Bounded fan-in network)**.** A bounded fan-in network of width w and depth T has nonnegative activations

$$z^{(0)} = (x, 0, \dots, 0), \qquad z^{(r)} = \mathrm{ReLU}(B_r z^{(r-1)} + \beta_r), \quad 1 \le r \le T, \tag{2.4}$$

where every row and every column of B_r has at most two permitted entries, fixed in advance; only their values depend on the target. The scalar output is an affine function of at most two neurons of the last layer. Copying a value, keeping it for later use, and creating a constant all use neurons.

Every ReLU network can be converted into this bounded fan-in normal form; Appendix G counts the cost exactly.

We count the resources of the fixed architecture, including connections that are permitted but assigned weight zero. Thus the *number of parameters of an architecture* P counts every permitted weight and bias, including fixed routing values. For three-regular hidden masks,

$$P \le 4nL + 3. \tag{2.5}$$

Logarithms in layer counts are base two, and $C, C_0, \dots$ denote absolute constants unless their dependence is shown.

3 Exact representation on three-regular expander masks

We state the representation theorem first, then give its general-network consequence and explain the mask construction and routing.

Theorem 3.1 (Exact representation on a fixed expanding mask)**.** *For every power of two $n \ge 32$, the three-regular mask A_n constructed below satisfies $\sigma_2(\frac{1}{3}A_n) \le \rho^* < 1$ (Theorem 3.3), and there is an integer $S_n = O(\log^2 n)$ such that every bounded fan-in network of width at most $n/2$, with input dimension $d \le n/2$ and depth T, is represented exactly on $[0,1]^d$ by the constant mask sequence $(A_n, A_n, \dots)$ with*

$$L = 1 + (T+1)S_n, \qquad P \le 4nL + 3. \tag{3.1}$$

If the weights and biases of the bounded fan-in network, including its output layer, have magnitude at most M, then those of the masked network have magnitude at most $\max\{1, M\}$. The masked network uses only copies of the parameters of the bounded fan-in network and the values 0 and 1; in particular, dyadic precision is preserved.

Since binary trees that copy values and accumulate sums convert every ReLU network into a bounded fan-in network, the theorem extends to arbitrary ReLU networks.

Corollary 3.2 (Representation of general ReLU networks)**.** *Let $\mathcal{N}$ be a ReLU network on $[0,1]^d$ with scalar output, D hidden layers of width at most $w \ge d$, and an affine output layer counted as layer $D+1$. For each layer r, let E_r be its number of nonzero weights and Δ_r the maximum of*

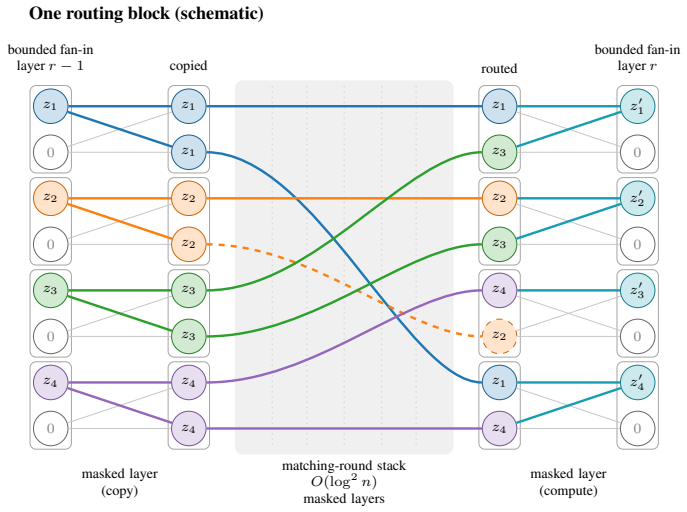


Figure 1: One routing block (schematic). The activations $z_1, \ldots, z_4$ at layer $r-1$ sit on the even-position neurons. A switch layer copies each value (coloured edges; grey edges are permitted connections with weight zero), matching rounds (dotted) move each copy to a neuron where it is read, and a final layer computes $z'_1, \ldots, z'_4$ and restores the zeros; the unused copy of z_2 (dashed) is ignored. In A_n, the copy layer uses the pair $(u, R(u))$ and the final layer the pair $(C^{-1}u, u)$. The mask is shown in Figure 2.

one and its largest fan-in or fan-out, and put $E = \max_r E_r$. *Then* $\mathcal{N}$ *is represented exactly by a network on* $(A_n, A_n, \ldots)$ *for some* $n \le C(w + E + 1)$, *with*

$$L \le C \log_2^2(w + E + 2)\Big[1 + \sum_{r=1}^{D+1} \log_2(\Delta_r + 1)\Big], \qquad P \le 4nL + 3,$$

and with parameters of magnitude at most $\max\{1, M\}$ *if those of* $\mathcal{N}$ *are at most* M *(Appendix G).*

We now describe the mask behind the theorem. Its construction combines uniform expansion with a switch structure that supports routing.

How the masks are built. Let G be a simple connected four-regular graph on $n/4$ vertices, with its edges oriented along an Euler circuit so that each vertex has two incoming and two outgoing edges. Replace every vertex by a cluster of four neurons, one per incident edge, ordered cyclically so that outgoing and incoming edges alternate. Let C be the cyclic permutation of each cluster and R the permutation exchanging the two neurons of each edge of G, and set $A = I + C + R$. Since C moves every neuron within its cluster and R moves it to another cluster, A is binary and three-regular. On each pair joined by R, the connections $I + R$ form a *switch* [3], a 2×2 block with all four connections permitted, so A is a switch layer with one extra connection per neuron from C. Numbering each cluster $0, 1, 2, 3$ along C, neurons of outgoing edges occupy the even positions; we call them *even-position* neurons, and the others *odd-position* neurons. Figure 2 in Appendix B illustrates the construction on a fragment of G.

Theorem 3.3 (Spectral gap of the mask A_n)**.** *Let the adjacency matrix* U_G *of* G *satisfy a Laplacian gap bound* $u^\top(4I - U_G)u \ge \gamma \|u\|_2^2$ *for* $u \perp \mathbf{1}$. *Then the mask* $A = I + C + R$ *satisfies*

$$\sigma_2(\frac{1}{3}A) \le \sqrt{1 - \kappa(\gamma)/9} < 1, \qquad \kappa(\gamma) := \frac{2\gamma}{\gamma + 16}. \tag{3.2}$$

For every power of two $n \ge 32$, *Lemma B.1 supplies auxiliary Ramanujan graphs with* $\gamma_R = 4 - 2\sqrt{3} > 1/2$, *and the resulting mask satisfies*

$$\sigma_2(\frac{1}{3}A_n) \le \rho^* := \sqrt{295/297} < 1. \tag{3.3}$$

The construction is a directed variant of the replacement product [49]: the cycles C enforce near-agreement within clusters, while R propagates it between clusters (Appendix B). Both connections are necessary: without R the clusters disconnect, and without C the pairs disconnect. The bound is independent of the width, which is the essential point: two-regular masks A have $\sigma_2(\frac{1}{2}A) \ge \cos(\pi/n) \to 1$ (Proposition B.4). With $\gamma = \gamma_R$ it gives $\sigma_2(\frac{1}{3}A_n) \le 0.9964$; the constants are not optimised. The resulting masks have the width-independent spectral gap established in Theorem 3.3.

Expansion gives the mask a uniform spectral gap, while exact simulation also requires routing values to prescribed neurons. We next explain how the construction provides that routing.

How the representation works, and why mixing is not enough. The masked network stores activations on even-position neurons and zero on odd-position neurons. Each routing block has three parts (Figure 1). A switch layer first copies the value at each even-position neuron u into $R(u)$. A sequence of matching rounds on Γ_n, the graph with adjacency matrix $C + C^\top + R$, then moves the copies to the neurons where they are needed; a matching round exchanges the values across the edges of a matching. By the eigenvalue bound of Alon, Chung and Graham [1], the *routing number* $\mathrm{rt}(\Gamma_n)$, the least number of rounds that realises every permutation, is $O(\kappa(\gamma_R)^{-2}\log^2 n)$ (Lemma B.5), and each round is realised by at most 29 layers with mask A_n (Lemma B.6). A final layer computes the next layer of the bounded fan-in network at each even-position neuron u from the values at u and $C^{-1}u$, and restores the zeros. Repeating this fixed-mask block gives Theorem 3.1 with $S_n = 2 + 29\ \mathrm{rt}(\Gamma_n)$. The routing depth is optimal up to one logarithmic factor (Remark B.7); full proofs are in Appendix B.

Fast spectral mixing connects every input to every output after logarithmically many layers. Proposition 3.4 shows that this property can coexist with a routing obstruction for simultaneous movement of all n values to prescribed positions (Lemma B.8).

Proposition 3.4 (A Ramanujan mask that cannot route)**.** *Index rows and columns by $\mathbb{Z}/7\mathbb{Z}$ and set*

$$A_{j,i} = 1 \quad \Longleftrightarrow \quad j - i \in \{0, 1, 3\} \pmod 7.$$

Then A is three-regular and $\sigma_2(\frac{1}{3}A) = \frac{\sqrt{2}}{3}$, so its bipartite graph, the Heawood graph, is Ramanujan. Nevertheless, no stack of copies of A routes an odd permutation of the seven neurons along vertex-disjoint paths.

The parity proof is given in Appendix B (Figure 4).

4 Uniform approximation with explicit width and depth

By Theorem 3.1, approximation on the fixed masks reduces to approximation by bounded fan-in networks, with no loss of accuracy. We therefore need an approximation scheme that such networks compute exactly. Piecewise-linear interpolation on a triangulated grid has this property, so the only errors are the interpolation error and the error in the coefficients.

Piecewise-linear interpolation on the Kuhn triangulation. Fix an integer $K \geq 1$, let $V_K = \{0, \dots, K\}^d$ be the grid, and let $N = |V_K| = (K+1)^d$ be its number of points. For each $v \in V_K$, set

$$\psi_v(x) = \mathrm{ReLU}\Big(1 - \max_i \mathrm{ReLU}(Kx_i - v_i) - \max_i \mathrm{ReLU}(v_i - Kx_i)\Big). \tag{4.1}$$

These are the hat functions of the Kuhn (Freudenthal) triangulation of the grid [28]. The usual product of one-dimensional hats would require multiplication; written with maxima, as in equation 4.1, the hat functions are computed exactly by a ReLU network (Figure 6). The accuracy of the interpolant is governed by the modulus of continuity

$$\omega_f(r) = \sup\big\{|f(x) - f(y)| :\ x, y \in [0,1]^d,\ \|x - y\|_\infty \leq r\big\}.$$

Lemma 4.1 (Partition of unity and coefficient stability)**.** *For every $x \in [0,1]^d$,*

$$\psi_v(x) \geq 0, \quad \sum_{v \in V_K} \psi_v(x) = 1, \quad \psi_v(x) > 0 \implies \|x - v/K\|_\infty < 1/K. \tag{4.2}$$

Consequently, if $|c_v - f(v/K)| \leq \delta$, then

$$\Big\| f - \sum_v c_v \psi_v \Big\|_\infty \leq \omega_f(1/K) + \delta. \tag{4.3}$$

More generally, replacing the coefficients c_v by c'_v changes the interpolant by at most $\max_v |c_v - c'_v|$ in uniform norm, and coefficients in $[a, b]$ give an interpolant with values in $[a, b]$.

Trading width for depth. We evaluate the grid hats in batches, so the batch size controls the tradeoff between width and depth.

Theorem 4.2 (Approximation on one fixed three-regular architecture)**.** *Let $f \in C([0,1]^d)$, $K \geq 1$ and $N = (K+1)^d$. Let $n \geq 16(d+1)$ be a power of two and*

$$1 \leq q \leq \min\{N, \lfloor n/(16(d+1)) \rfloor\}.$$

For all coefficients with $|c_v - f(v/K)| \leq \delta$, there is a network F on the constant mask sequence $(A_n, A_n, \ldots)$ with $F = \sum_v c_v \psi_v$ on $[0,1]^d$; hence

$$\|F - f\|_\infty \leq \omega_f(1/K) + \delta, \tag{4.4}$$

$$L \leq C\lceil N/q \rceil \left(1 + \lceil \log_2(d+1) \rceil + \lceil \log_2(q+1) \rceil\right) \log_2^2 n, \qquad P \leq 4nL + 3. \tag{4.5}$$

All parameters and all hidden activations are bounded in magnitude by

$$\max\{1, K, \max_v |c_v|\}. \tag{4.6}$$

The construction behind Theorem 4.2 processes the grid points in batches of q (Figure 7). In each batch, it copies the input coordinates, evaluates q hat functions in parallel with binary trees of maxima, and adds their contributions to two nonnegative partial sums, one for positive and one for negative coefficients; the output layer takes their difference. Counting every copy, its width is at most $8q(d+1)$. To fit on the masks of Theorem 3.1, this width must not exceed $n/2$, which gives the constraint on q above.

The width n enters only through the batch size q. At the smallest admissible width, with $q = 1$,

$$n < 32(d+1), \qquad L = O(N \log^3(d+1)), \qquad P = O((d+1)N \log^3(d+1)). \tag{4.7}$$

Larger widths parallelise the work, giving depth $O(\lceil N/q \rceil \log^3((d+1)(q+1)))$ at width $O(q(d+1))$.

The error bound now gives universality: as K grows, $\omega_f(1/K) \to 0$, while the admissible mask widths remain fixed.

Corollary 4.3 (Universality of a fixed mask sequence)**.** *For every $d \geq 1$ and every power of two $n \geq 16(d+1)$, the constant mask sequence $(A_n, A_n, \ldots)$ has the universal approximation property on $[0,1]^d$, in the uniform norm and in every L^p norm.*

For Hölder targets, the error bound gives explicit rates. Suppose $|f(x) - f(y)| \leq H \|x - y\|_\infty^\beta$, with $0 < \beta \leq 1$ and $\|f\|_\infty \leq M$. Choose $K = \max\{1, \lceil (2H/\varepsilon)^{1/\beta} \rceil\}$ and round the coefficients. This gives uniform error at most ε with $N = \Theta(\varepsilon^{-d/\beta})$ grid points, $P = O(N)$ parameters and $O(\log(1/\varepsilon))$ bits per parameter for fixed d, β, H, M (Corollary C.2). The next section gives a bounded-weight construction and reduces the encoding length to the metric-entropy order $\varepsilon^{-d/\beta}$.

5 Bounded weights, encoding length, and compositional targets

Bounded weights and activations. The weights of size K in equation 4.6 arise only in the deviations $Kx_i - v_i$ of the hat functions equation 4.1, and they can be traded for depth without losing exactness. The network replaces multiplication by $K = 2^k$ with k clipped doubling steps $y \mapsto \min\{1, 2y\}$, each a two-layer ReLU map with coefficients in $\{0, \pm 1, 2\}$. The hat functions depend only on these clipped values, so the replacement preserves them exactly (Appendix D).

Theorem 5.1 (Bounded weights and activations)**.** *Suppose $K = 2^k$ with an integer $k \geq 0$, and $|c_v| \leq M_0$. Under the width and batch conditions of Theorem 4.2, the same interpolant can be realised on the same mask sequence with*

$$L \leq C\lceil N/q \rceil \left(1 + k + \lceil \log_2(d+1) \rceil + \lceil \log_2(q+1) \rceil\right) \log_2^2 n. \tag{5.1}$$

All parameters, and all activations on the cube, are bounded in magnitude by $\max\{2, M_0\}$. If every c_v is dyadic with at most t fractional bits, the parameters need at most $\max\{k, t\}$ fractional bits.

Encoding length. Parameter correlations can make a network describable with fewer bits than its number of parameters times their precision. We measure its *encoding length* $\mathcal{B}$ as the length of a bit string from which a fixed decoder $\mathcal{D}_\varepsilon$ reconstructs all parameters on a prescribed prefix of the masks. The decoder, the masks, the dimension, the accuracy and the target class are fixed in advance; only the bit string depends on the target, and decoding happens before the network is run. We consider the class $\mathcal{H}_{\beta,H,M}$ of functions on the cube with Hölder exponent β, Hölder constant at most H and uniform norm at most M.

For such targets, neighbouring coefficients of the interpolant are close, so we encode successive differences. We order the grid points along a path in which consecutive points differ by one step in a single coordinate (Figure 8). After rounding at a scale comparable to ε, the first coefficient needs $O(\log(1 + M/\varepsilon))$ bits, and each subsequent difference takes one of five values, so three bits suffice. The routing weights and the structure of the bounded fan-in network do not depend on the target and are regenerated by the decoder.

Theorem 5.2 (Delta coding on the fixed masks)**.** *Let $d \geq 1$, $0 < \beta \leq 1$, $H, M \geq 0$ and $0 < \varepsilon \leq 1$, and let $K = \max\{1, \lceil (2H/\varepsilon)^{1/\beta} \rceil\}$ and $N = (K+1)^d$. There is a fixed decoder whose networks, on the masks of width $O(d)$ from equation 4.7, approximate every $f \in \mathcal{H}_{\beta,H,M}$ within ε, with encoding length*

$$\mathcal{B} \leq \lceil \log_2(3 + 8M/\varepsilon) \rceil + 3(N-1). \tag{5.2}$$

This is optimal up to a constant factor. A standard packing argument shows that every decoder approximating all of $\mathcal{H}_{\beta,H,M}$ within a sufficiently small ε must have encoding length $\mathcal{B} \geq 4^{-d-d/\beta}(H/\varepsilon)^{d/\beta}$ bits (Theorem D.2). Hence $\mathcal{B} = \Theta(\varepsilon^{-d/\beta})$ for fixed d, β and $H, M > 0$, which removes the logarithmic factor of separate storage. The exponent d/β is classical; the contribution of Theorem 5.2 is that this order is attained within the fixed three-regular architecture.

Compositional targets. The encoding bound still uses $N = (K+1)^d$ coefficients, reflecting a generic grid over all d input variables. We next exploit compositional structure to reduce the dependence on dimension. For compositions of Lipschitz functions arranged in h levels, each depending on at most s variables of the previous level (Definition E.1, following [50]), the fixed masks of width $O(p(s+1))$, where $p \geq d$ bounds the number of variables per level, achieve accuracy ε with depth of order $h(2h/\varepsilon)^s$ up to logarithmic factors for 1-Lipschitz components (Theorem E.2 and Corollary E.4 in Appendix E). The exponent of $1/\varepsilon$ is thus s rather than d.

6 Robustness to pruning and weight perturbations

Robustness to pruning. Let $Z \in \{0,1\}^{r \times r}$ be the biadjacency matrix of a connected d_Z-regular bipartite graph, and write $\nu = \sigma_2(\frac{1}{d_Z} Z)$ for its spectral parameter (Section 2). Replace each neuron i by a block of r neurons and use the hidden mask

$$\widetilde{A}_n = A_n \otimes Z. \tag{6.1}$$

Each connection permitted by the mask of A_n becomes a copy of Z between two blocks. The resulting mask is binary and $3d_Z$-regular. By Lemma H.2, its gap is the smaller of the gaps of A_n and Z; the pruning tolerance below depends only on ν. The input and output layers are replicated in the same way, and pruning acts only on the hidden masks (Definition H.1).

Definition 6.1 (Block-wise pruning ratio)**.** A pruning deletes permitted hidden connections, whose weights are then zero. Its block-wise pruning ratio is at most φ if every copy of Z loses at most $\varphi r d_Z$ connections in every hidden layer; biases are retained.

Theorem 6.2 (Exact representation after pruning)**.** *Suppose $\nu < 1/\sqrt{2}$ and*

$$\varphi \leq \frac{(1 - \sqrt{2}\nu)^2}{8}. \tag{6.2}$$

For every pruning of the masks equation 6.1 with block-wise pruning ratio at most φ and every bounded fan-in network of width at most $n/2$, input dimension $d \leq n/2$ and depth T, there are weights on the remaining connections that represent the bounded fan-in network exactly on $[0,1]^d$, with $L = 1 + (T+1)S_n$ hidden layers of width nr. In every layer, each block contains at least $(1 + \sqrt{2}\nu)r/2$ neurons with the correct value. The weights are copies of parameters of the bounded fan-in network or the values 0 and 1.

For a degree-64 bipartite Ramanujan expander Z, $\nu \leq 2\sqrt{63}/64 < 1/4$, so equation 6.2 permits $\varphi = 1/20$. The resulting hidden masks therefore have degree $3d_Z = 192$.

The proof shows by induction over layers that every block keeps a majority of correct neurons: by the expander mixing lemma for Z, few neurons can lose all their connections to the correct neurons of an input block (Appendix H). This identifies expansion as the structural property that enables resilience. If Z has the same degree but splits into small components, the network splits into independent slices, and a pruning can disconnect each slice in turn.

Proposition 6.3 (A bipartite graph Z without expansion)**.** *Let Z be a disjoint union of copies of the complete bipartite graph K_{d_Z,d_Z}, with d_Z dividing r and $d_Z \leq \varphi r$. Then Z is d_Z-regular with $\nu = 1$, and there is a pruning with block-wise pruning ratio at most φ after which every sufficiently deep network on the masks equation 6.1 computes a function independent of the input. In particular, it cannot approximate $x \mapsto x_1$ to error below $1/2$ once the depth is at least $\lceil 2/\varphi \rceil + 1$.*

The pruning results concern deleted connections. We now consider the effect of small perturbations to all weights, a different source of error.

Stability under weight perturbations. Theorem F.1 in Appendix F bounds the effect of perturbing all parameters of the architecture by at most η, including permitted connections assigned weight zero. The bound has a routing factor $S_n = O(\log^2 n)$ and a term $e^{3\eta L}$.

7 Discussion

Our results answer the questions raised in the introduction: for every admissible width, a three-regular mask, of the smallest regular degree compatible with a uniform spectral gap, supports universal approximation when reused in every layer, and the connections that create its gap also route values through the network. The construction uses routing structure alongside expansion, which also supplies redundancy against structured pruning. For the empirical work that motivated this paper, selecting masks for their spectral properties is therefore compatible with full expressivity.

Limitations. The representation theorem is proved for our constructed family, with auxiliary graphs supplied by an existence theorem. The resulting masks are guaranteed to be spectral expanders; we do not establish that they are Ramanujan. The width-seven example establishes a routing obstruction for vertex-disjoint paths and leaves other computations on that mask open. Our results concern this constructed family and do not address training. The construction is also far from optimal in its constants. Routing costs $O(\log^2 n)$ layers, compared with $O(\log n)$ for a Beneš network; the estimate contains a factor $\kappa(\gamma_R)^{-2} \approx 238$ before the absolute constants of [1] and Lemma B.6; the gap in equation 3.3, although width-independent, is small; and the width constants are not optimised. Representing a dense network requires width proportional to its number of weights, or additional depth (Appendix G). On the approximation side, interpolation depends exponentially on the dimension, and the width-depth product of the generic construction grows linearly with the number of grid points. At minimum width, our approximation bounds do not match unrestricted bounds; Appendix I gives lower bounds from bounded receptive fields. The results cover Hölder exponents $0 < \beta \leq 1$ and the stated compositions, and higher smoothness requires a different bounded fan-in network. The optimal encoding length refers to a fixed decoder rather than to separately stored parameters, and the exponent d/β is classical. The pruning theorem chooses the remaining weights after the deletions are known and assumes that deletions are spread over the copies of Z. The theorem therefore gives no guarantee for a trained network pruned without knowledge of the deletions. This work also leaves optimisation, generalisation and running time unaddressed. The stability theorem bounds perturbation effects, while the spectral gap alone gives no control of weight amplification.

Open questions. These limitations suggest natural next steps: routing in $O(\log n)$ layers at degree three; exact representation on less specifically designed masks, such as random regular or generic Ramanujan masks; pruning guarantees for fixed weights; and understanding whether, and how, trained networks use the expanding connections.

AI USE STATEMENT

We used an LLM for improving and condensing the initial draft (of over 40 pages, with appendix), some of the figures in the manuscript, checking LaTeX structure, and literature leads. The authors take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

This is a theoretical work with no datasets or experiments. All definitions and assumptions are stated in Section 2, and every result in Sections 3–6 is proved in the appendix: the mask construction and exact representation in Appendix B, the interpolation schedules in Appendix C, bounded weights and encoding length in Appendix D, compositional targets in Appendix E, stability in Appendix F, general networks in Appendix G, and pruning in Appendix H. All constants are either explicit or stated as absolute constants.

REFERENCES

[1] Noga Alon, F. R. K. Chung, and R. L. Graham. Routing permutations on graphs via matchings. *SIAM Journal on Discrete Mathematics*, 7(3):513–530, 1994.

[2] Omar Bazarbachi, Zijun Sun, and Yanning Shen. EGGS-PTP: An Expander-Graph Guided Structured Post-training Pruning Method for Large Language Models. arXiv:2508.09471, 2025.

[3] V. E. Beneš. Optimal rearrangeable multistage connecting networks. *Bell System Technical Journal*, 43(4):1641–1656, 1964.

[4] Arindam Biswas, Suryam Arnav Kalra, Pabitra Mitra, and Biswajit Basu. Sparse network initialization using deterministic Ramanujan graphs. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024.

[5] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

[6] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1):8–45, 2019.

[7] Rebekka Burkholz. Most activation functions can win the lottery without excessive depth. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[9] Arthur da Cunha, Emanuele Natale, and Laurent Viennot. Proving the lottery ticket hypothesis for convolutional neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

[10] Kiara Esguerra, Muneeb Nasir, Tong Boon Tang, Afidalina Tumian, and Eric Tatt Wei Ho. Sparsity-aware orthogonal initialization of deep neural networks. *IEEE Access*, 11:74165–74181, 2023. doi: 10.1109/ACCESS.2023.3295344.

[11] Amaljith Ev, Arindam Biswas, Suryam Arnav Kalra, Pabitra Mitra, and Biswajit Basu. XoRA: Expander Adapted LoRA Finetuning. In *Fine-Tuning in Modern Machine Learning: Principles and Scalability (FITML) Workshop at NeurIPS 2024*, 2024.

[12] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning (ICML)*, 2020.

[13] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[14] Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *International Conference on Machine Learning (ICML)*, 2023.

[15] Moritz Grillo and Tobias Hofmann. On the Expressivity of Sparse Maxout Networks. arXiv:2510.14068, 2025.

[16] Bernhard Haeupler and Antti Roeyskoe. Maintaining Routing Structures under Deletions via Self-Pruning. arXiv:2507.13044, 2025.

[17] Boris Hanin and Mark Sellke. Approximating continuous functions by ReLU nets of minimal width, 2017. arXiv:1710.11278.

[18] J. He and J. Xu. Deep neural networks and finite elements of any order on arbitrary dimensions. *arXiv preprint arXiv:2312.14276*, 2024.

[19] Duc N. M. Hoang, Souvik Kundu, Shiwei Liu, and Zhangyang Wang. Don't just prune by magnitude! Your mask topology is a secret weapon. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[20] Duc N. M. Hoang, Shiwei Liu, Radu Marculescu, and Zhangyang Wang. Revisiting pruning at initialization through the lens of Ramanujan graph. In *International Conference on Learning Representations (ICLR)*, 2023.

[21] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021.

[22] R. Hong and A. Kratsios. Bridging the gap between approximation and learning via optimal approximation by ReLU MLPs of maximal regularity. *arXiv preprint arXiv:2409.12335*, 2024.

[23] Suryam Arnav Kalra, Pabitra Mitra, Arindam Biswas, and Biswajit Basu. Graph expansion in pruned recurrent neural network layers preserves performance. In *The Second Tiny Papers Track at ICLR 2024*, 2024.

[24] Suryam Arnav Kalra, Arindam Biswas, Pabitra Mitra, and Biswajit Basu. Sparse neural architectures via deterministic Ramanujan graphs. *Transactions on Machine Learning Research*, 2025.

[25] Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on Learning Theory (COLT)*, 2020.

[26] N. Kim, C. Min, and S. Park. Minimum width for universal approximation using ReLU networks on compact domain. In *International Conference on Learning Representations*, 2024.

[27] A. Kratsios, S. Brugiapaglia, B. J. Kim, G. Cousins, and H. Sáez de Ocáriz Borde. Algorithmic foundations of deep learning: Complexity-theoretic rates and a characterization of universal approximation. *arXiv preprint arXiv:2606.26705*, 2026.

[28] H. W. Kuhn. Some combinatorial lemmas in topology. *IBM Journal of Research and Development*, 4(5):518–524, 1960. doi: 10.1147/rd.45.0518.

[29] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations (ICLR)*, 2019.

[30] Christoph Lenzen, Moti Medina, Mehrdad Saberi, and Stefan Schmid. Robust routing made easy: Reinforcing networks against non-benign faults. *IEEE/ACM Transactions on Networking*, 2023. arXiv:2307.05547.

[31] L. Li, Y. Duan, G. Ji, and Y. Cai. Minimum width of leaky-ReLU neural networks for uniform universal approximation. *arXiv preprint arXiv:2305.18460*, 2024.

[32] C. Liu and M. Chen. ReLU network with width $d + \mathcal{O}(1)$ can achieve optimal approximation rate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 30755–30788, 2024.

[33] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *International Conference on Learning Representations (ICLR)*, 2022.

[34] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[35] Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning (ICML)*, pp. 6682–6691, 2020.

[36] A. W. Marcus, D. A. Spielman, and N. Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. *Annals of Mathematics*, 182(1):307–325, 2015.

[37] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1):2383, 2018.

[38] Rajko Nenadov. Routing permutations on spectral expanders via matchings. *Combinatorica*, 43:737–742, 2023.

[39] Adi Nilli. On the second eigenvalue of a graph. *Discrete Mathematics*, 91:207–210, 1991.

[40] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[41] Bithika Pal, Arindam Biswas, Sudeshna Kolay, Pabitra Mitra, and Biswajit Basu. A study on the Ramanujan graph property of winning lottery tickets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17186–17201. PMLR, 2022.

[42] S. Park, C. Yun, J. Lee, and J. Shin. Minimum width for universal approximation. In *International Conference on Learning Representations*, 2021.

[43] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via SubsetSum: Logarithmic over-parameterization is sufficient. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[44] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

[45] Philipp Petersen and Felix Voigtlaender. Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proceedings of the American Mathematical Society*, 148(4):1567–1581, 2020.

[46] Nicholas Pippenger and Geng Lin. Fault-tolerant circuit-switching networks. *SIAM Journal on Discrete Mathematics*, 7(1):108–118, 1994.

[47] Ameya Prabhu, Girish Varma, and Anoop Namboodiri. Deep expander networks: Efficient deep networks from graph theory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–35, 2018.

[48] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[49] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002.

[50] J. Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

[51] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[52] James Stewart, Umberto Michieli, and Mete Ozay. Data-free model pruning at initialization via expanders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4519–4524, 2023. doi: 10.1109/CVPRW59228.2023.00475.

[53] Y.-L. Sung, V. Nair, and C. Raffel. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24193–24205, 2021.

[54] Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[55] G. Vardi, G. Yehudai, and O. Shamir. Width is less important than depth in ReLU neural networks. In *Proceedings of the 35th Conference on Learning Theory*, volume 178, pp. 1249–1281, 2022.

[56] L. G. Valiant. Universal circuits (preliminary report). In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing (STOC)*, pages 196–203, 1976.

[57] Dharma Teja Vooturi, Girish Varma, and Kishore Kothapalli. Ramanujan Bipartite Graph Products for Efficient Block Sparse Neural Networks. arXiv:2006.13486, 2020.

[58] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations (ICLR)*, 2020.

[59] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[60] Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *International Conference on Machine Learning (ICML)*, pp. 10881–10891, 2020.

[61] Chulhee Yun, Yin-Wen Chang, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. $O(n)$ connections are expressive enough: Universal approximability of sparse transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[62] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[63] S. Zhao, Y. Yu, Y. Zhang, and H. Liu. Valiant's universal circuits revisited: An overall improvement and a lower bound. In *Advances in Cryptology–ASIACRYPT*, volume 11921, pp. 401–425, 2019.

[64] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.

TABLE OF CONTENTS - APPENDIX

Background 15
Graph construction and exact representation 15
Interpolation and the batched bounded fan-in network 21
Bounded activations and encoding length 25
Proofs for compositional targets 27
Proof of the stability bound 29
Representation of general sparse and dense networks 30
Robustness to pruning 31
Lower bounds from bounded receptive fields and output conventions 33

A BACKGROUND

This appendix introduces the graph-theoretic and approximation-theoretic objects used in the main text. A binary mask is the biadjacency matrix of a bipartite graph between consecutive hidden layers: an entry equal to one permits the corresponding weight, while an entry equal to zero fixes it to zero. For example, a three-regular mask permits exactly three incoming and three outgoing connections at every hidden neuron.

The first layer places the input coordinates in prescribed neurons and sets the remaining activations to zero. The output layer reads two prescribed neurons. There are no linear hidden units: the values we copy are nonnegative, so ReLU acts as the identity on them.

A mask is m-regular when every row and every column contains exactly m ones. Its normalised biadjacency matrix is $P = \frac{1}{m}A$. Thus P is doubly stochastic: the constant vector is a singular vector with singular value one, and the remaining singular values measure deviations from uniform mixing. Writing $J_n = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, the relevant quantity is

$$\sigma_2(P) = \|P - J_n\|_2 .$$

The inequality $\sigma_2(P) \le \rho < 1$ means that a single layer contracts zero-mean discrepancies by a factor at most ρ. For a family of masks, the same constant ρ must work at every width; this is the width-independent spectral gap used in the main results.

These are different notions of connectivity. If $\|P_t \cdots P_1 - J_n\|_2$ is small, then every input may have a path to every output in the layered graph. Routing asks for more: the paths for all inputs must be chosen simultaneously, with no two paths sharing a neuron in an intermediate layer. The construction in the paper therefore uses the spectral gap to control mixing and a separate matching-round argument to control rearrangement.

For approximation, the grid points of $\{0, \dots, K\}^d$ index local hat functions. The Kuhn (Freudenthal) triangulation [28] divides each grid cube into simplices; its hat functions form a partition of unity. A ReLU network can evaluate these hats using maxima, then combine their coefficients to produce a piecewise-linear interpolant.

More concretely, inside each grid cube the triangulation selects a simplex according to the ordering of the fractional coordinates. At a point x, only the vertices of that simplex have nonzero hat values, those values are nonnegative, and they sum to one. Consequently, the interpolant $\sum_v c_v \psi_v(x)$ is a convex combination of nearby coefficients. If $c_v = f(v/K)$, its uniform error is controlled by the modulus of continuity of f at scale $1/K$; perturbing every coefficient by at most δ adds at most δ to the error. The max-and-ReLU representation of the hats is why this interpolation scheme fits the bounded fan-in compilation used later.

B GRAPH CONSTRUCTION AND EXACT REPRESENTATION

This appendix proves the graph and representation statements of Section 3. We use two external results: the existence of bipartite Ramanujan 2-lifts [36], quoted below with its precise consequence, and the eigenvalue bound for the routing number of Alon, Chung and Graham [1]. The singular-value estimate, the implementation of matching rounds and the representation are proved in full.

B.1 THE AUXILIARY RAMANUJAN GRAPHS

Lemma B.1 (Auxiliary graph at every required size)**.** *For every power of two $s \ge 8$, there is a simple connected four-regular bipartite Ramanujan graph G on s vertices, with adjacency matrix U_G, such that*

$$y^\top (4I - U_G) y \ge (4 - 2\sqrt{3}) \, \|y\|_2^2 \quad \textit{for every } y \perp \mathbf{1}. \tag{B.1}$$

Proof. The complete bipartite graph $K_{4,4}$ is simple and four-regular, with adjacency eigenvalues 4, -4 and 0. Marcus, Spielman and Srivastava [36, Theorems 5.3 and 5.5] show that every bipartite regular Ramanujan graph of degree D has a bipartite 2-lift whose new eigenvalues all have absolute value at most $2\sqrt{D-1}$. A 2-lift replaces each vertex by two copies and each edge by one of the

two perfect matchings between the copies of its endpoints. It preserves simplicity, regularity and bipartiteness, and it doubles the number of vertices.

Starting from $K_{4,4}$ and applying this repeatedly gives, for every $s = 8, 16, 32, \ldots$, a graph whose nontrivial adjacency eigenvalues have absolute value at most $2\sqrt{3}$. Each graph is connected: otherwise the eigenvalue 4 would have multiplicity at least two, which is impossible because the previous graph is connected and the new eigenvalues are at most $2\sqrt{3} < 4$. On $\mathbf{1}^{\perp}$ the largest adjacency eigenvalue is at most $2\sqrt{3}$; the eigenvalue -4 only increases the Laplacian and causes no difficulty. The spectral decomposition gives equation B.1, and $4 - 2\sqrt{3} > 1/2$. Since G is bipartite, $\frac{1}{4}U_G$ has the eigenvalue -1, so we use only this one-sided Laplacian bound and not a contraction bound on all of $\mathbf{1}^{\perp}$. □

Remark B.2 (Role of the Ramanujan input graphs). The auxiliary graphs need only have a width-independent Laplacian gap; the Ramanujan property is used only to obtain an explicit value of γ. The masks A_n inherit a uniform singular-value gap by Theorem 3.3, but they need not themselves be Ramanujan.

B.2 Proof of the spectral gap

We first view the mask as a graph built from the clusters of G. The following figure records the construction, and the energy estimate below transfers the gap of G to this graph and then to the mask. Figure 2 shows the construction on a fragment of G.

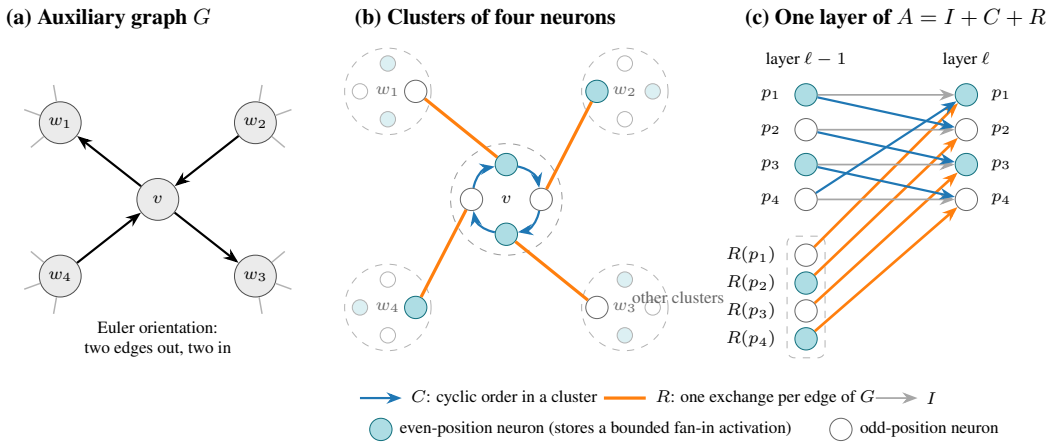


Figure 2: Construction of the mask of Section 3. (a) An Euler circuit orients the four-regular auxiliary graph G, so that every vertex has two outgoing and two incoming edges. (b) Every vertex becomes a cluster of four neurons, with even- and odd-position neurons alternating along the cyclic order C (blue). The exchange R (orange) joins an even-position neuron at the tail of each directed edge to an odd-position neuron at its head. The even-position neurons (teal) store the activations of the bounded fan-in network. (c) In one layer, a neuron p reads itself through I, its predecessor $C^{-1}p$ through C and its partner $R(p)$ through R, so every neuron has fan-in and fan-out three. Both kinds of connection are needed for the spectral gap: without R the clusters are disconnected, and without C the pairs joined by R are disconnected from one another.

Lemma B.3 (Energy bound for Γ). *Let G satisfy*

$$u^{\top}(4I - U_G)u \geq \gamma \|u\|_2^2 \qquad (u \perp \mathbf{1}) \tag{B.2}$$

for some $\gamma > 0$, and let Γ be the graph on the $n = 4s$ neurons with adjacency matrix $U_\Gamma = C + C^{\top} + R$. Then, with $\kappa(\gamma)$ as in Theorem 3.3,

$$x^{\top}(3I - U_\Gamma)x \geq \kappa(\gamma) \|x\|_2^2 \qquad (x \perp \mathbf{1}). \tag{B.3}$$

Proof. Let Q be the orthogonal projection that averages the four neurons of every cluster. Fix $x \perp \mathbf{1}$ and put

$$y = Qx, \qquad a = \|x - y\|_2, \qquad b = \|x - Rx\|_2, \qquad E = x^{\top}(3I - U_\Gamma)x.$$

Each edge of the cyclic orders appears once in $\|x - Cx\|_2^2$, and each edge of R twice in $\|x - Rx\|_2^2$, so $E = \|x - Cx\|_2^2 + \frac{1}{2}\|x - Rx\|_2^2$. The Laplacian of a four-cycle has eigenvalues $0, 2, 2, 4$ and vanishes exactly on constant vectors, so $\|x - Cx\|_2^2 \geq 2a^2$ and

$$E \geq 2a^2 + \tfrac{1}{2}b^2. \tag{B.4}$$

Write y as four equal copies of a vector u indexed by the vertices of G. Then $u \perp \mathbf{1}$ and $\|y\|_2^2 = 4\|u\|_2^2$. Every edge of G contributes to two coordinates of $y - Ry$, so, by equation B.2,

$$\|y - Ry\|_2^2 = 2 \sum_{\{i,j\} \in E(G)} (u_i - u_j)^2 = 2u^\top (4I - U_G)u \geq \frac{\gamma}{2}\|y\|_2^2 . \tag{B.5}$$

Also $\|y - Ry\|_2 \leq \|y - x\|_2 + \|x - Rx\|_2 + \|R(x - y)\|_2 = 2a + b$. Therefore

$$\|x\|_2^2 = a^2 + \|y\|_2^2 \leq a^2 + \frac{2}{\gamma}(2a + b)^2.$$

By equation B.4, $a^2 \leq E/2$, and the weighted Cauchy–Schwarz inequality gives $(2a + b)^2 \leq (2 + 2)(2a^2 + \frac{1}{2}b^2) \leq 4E$. Hence $\|x\|_2^2 \leq (\frac{1}{2} + \frac{8}{\gamma})E$, that is, $E \geq \kappa(\gamma)\|x\|_2^2$. □

Proof of Theorem 3.3. Fix $x \perp \mathbf{1}$ and let $E = x^\top(3I - U_\Gamma)x = \|x - Cx\|_2^2 + \frac{1}{2}\|x - Rx\|_2^2$, as in the proof of Lemma B.3. For permutation matrices P_1, P_2, P_3, expanding both sides gives

$$\left\| \tfrac{1}{3}\textstyle\sum_k P_k x \right\|_2^2 = \|x\|_2^2 - \tfrac{1}{9}\textstyle\sum_{k<l} \|P_k x - P_l x\|_2^2 .$$

With $(P_1, P_2, P_3) = (I, C, R)$ the sum is at least $\|x - Cx\|_2^2 + \|x - Rx\|_2^2 \geq E \geq \kappa(\gamma)\|x\|_2^2$ by Lemma B.3, so $\left\|\frac{1}{3}Ax\right\|_2^2 \leq (1 - \kappa(\gamma)/9)\|x\|_2^2$ for $x \perp \mathbf{1}$. Since $\frac{1}{3}A$ is doubly stochastic, $\sigma_2(\frac{1}{3}A)$ is its norm on $\mathbf{1}^\perp$, which proves equation 3.2. Finally, κ is increasing with $\kappa(1/2) = 2/33$, so Lemma B.1 gives $\sigma_2(\frac{1}{3}A_n)^2 < \frac{295}{297}$. □

We next verify that degree three is the minimum regular degree compatible with a uniform spectral gap.

Proposition B.4 (Minimal regular degree for uniform expansion)**.** *No family of one- or two-regular masks of unbounded width has a uniform spectral gap. For a two-regular mask of width $n \geq 2$, either $\sigma_2(\frac{1}{2}A) = 1$, or the bipartite graph of A is a single cycle and*

$$\sigma_2(\tfrac{1}{2}A) = \cos(\pi/n).$$

Proof of Proposition B.4. A one-regular mask is a permutation matrix, whose singular values all equal one. The bipartite graph of a two-regular mask is a disjoint union of cycles. If it has at least two components, their normalised constant vectors give at least two independent singular vectors with singular value one.

If there is a single component, separate permutations of rows and columns bring the matrix to the form $I + \Pi$, where Π is the cyclic permutation matrix of order n. Diagonalising Π by the discrete Fourier transform shows that the singular values of $\frac{1}{2}(I + \Pi)$ are

$$\left| \frac{1 + e^{2\pi i k/n}}{2} \right| = |\cos(\pi k/n)|, \qquad 0 \leq k < n.$$

The second largest is $\cos(\pi/n)$, which tends to one as $n \to \infty$, along any unbounded sequence of widths. Hence no uniform gap is possible at degree two. □

B.3 Routing to prescribed positions

The spectral gap gives a mixing guarantee; exact representation also needs an explicit schedule that sends all values to their prescribed positions. We bound the number of matching rounds required for this routing.

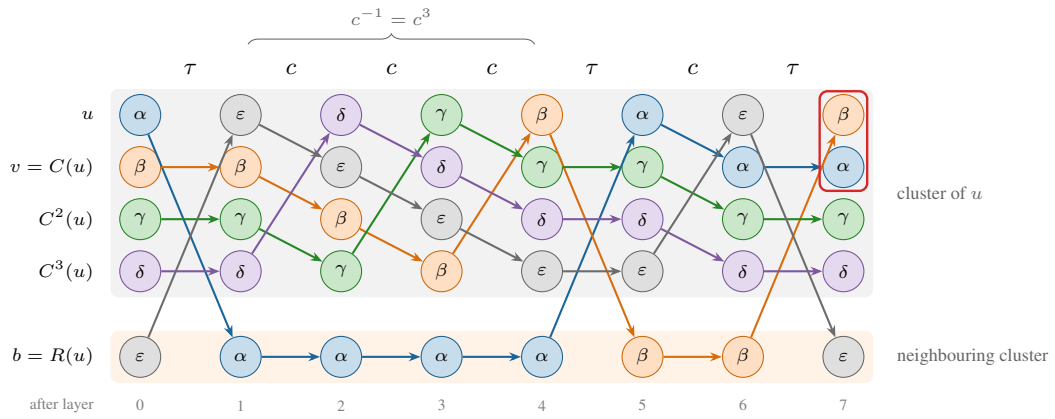


Figure 3: The seven-layer exchange of Lemma B.6 for neighbouring neurons u and $v = C(u)$. Each column shows the five values after one layer, and arrows follow each value. The layer τ exchanges u with the intermediate neuron $b = R(u)$ in a neighbouring cluster through R; the layer c rotates the cluster of u one step through C. Since C moves values in one direction only, c^{-1} costs three layers. The product $\tau\,(c\tau c^{-1})\,\tau$ exchanges α and β (red box) and returns every other value, including the intermediate neuron's ε, to its place.

A *matching round* on a graph exchanges the values at the two ends of every edge of a matching. The *routing number* $\mathrm{rt}(\Gamma)$ of a graph Γ is the least number of rounds that realises every permutation of its vertices [1]. We write Γ_n for the graph Γ of Theorem 3.3 on n neurons; in Theorem 3.1 it is built from the graph of Lemma B.1.

Lemma B.5 (Routing number of the graph Γ_n)**.** *For the graph Γ_n built from an auxiliary graph with Laplacian gap γ, every permutation of the vertices is realised by at most*

$$C\,\kappa(\gamma)^{-2}\log_2^2 n \tag{B.6}$$

matching rounds.

Proof. Add three loops at every vertex of Γ_n. The resulting six-regular multigraph has adjacency matrix $U_\Gamma + 3I$. By Lemma B.3, the nontrivial eigenvalues of U_Γ lie in $[-3, 3-\kappa(\gamma)]$, so those of $U_\Gamma + 3I$ lie in $[0, 6-\kappa(\gamma)]$ and have absolute value at most $\mu := 6 - \kappa(\gamma)$.

We use the following result of Alon, Chung and Graham [1, Theorem 7]: if every nontrivial eigenvalue of the adjacency matrix of a connected m-regular graph on n vertices has absolute value at most $\mu < m$, then every permutation of its vertices can be realised in $O\big(m^2 \log^2 n/(m-\mu)^2\big)$ matching rounds. Their proof uses only the normalised adjacency matrix, counted with multiplicity, and they apply it to graphs with m added loops at every vertex [1, proof of Corollary 4]; it therefore applies to the multigraph above. Substituting

$$m = 6, \qquad \mu = 6 - \kappa(\gamma), \qquad m - \mu = \kappa(\gamma)$$

gives $O(\kappa(\gamma)^{-2}\log^2 n)$ rounds. A step along a loop leaves a value in place, so deleting these steps realises the same permutation with matchings of Γ_n. □

Lemma B.6 (One matching round on the fixed mask)**.** *Every matching round on Γ_n is realised by at most 29 layers with mask A_n, zero biases, and weights in $\{0, 1\}$ acting on nonnegative values.*

Proof. Let M be the matching of the round. Its edges in R are handled by one layer in which every matched pair uses R and every other neuron uses I. The other edges of M join neighbouring neurons of a cluster, at most two disjoint ones per cluster.

Let u and $v = C(u)$ lie in one cluster and $b = R(u)$ in another. Let τ be the layer that exchanges u and b (connections R on this pair, I elsewhere), and c the layer that rotates the cluster of u one step

along C (connections C inside this cluster, I elsewhere). Since c fixes b, $c\tau c^{-1}$ exchanges v and b, so

$$\tau\,(c\tau c^{-1})\,\tau = (u\ v),$$

and every other neuron, b included, returns to its place. As $c^{-1} = c^3$, this takes seven layers (Figure 3).

The auxiliary graphs of Lemma B.1 are bipartite. Apply these exchanges simultaneously to all clusters on one side of the bipartition, and then to the other side. Clusters on one side are not joined by R, so the intermediate neurons b lie in clusters on the other side, which are not rotated in that phase; they are distinct because R is a permutation, and each returns to its place. Each cluster needs at most two exchanges, so the edges of M inside clusters take at most $2 \cdot 2 \cdot 7 = 28$ layers, and the round at most 29. All weights are 0 or 1 and all biases are zero; since the values are nonnegative, ReLU acts as the identity. □

Remark B.7 (Routing-depth scale). After t layers of a three-regular mask, each output depends on at most 3^t inputs, so routing every permutation of n values needs at least $\log_3 n$ layers. Lemmas B.5 and B.6 give $O(\log^2 n)$, which is optimal up to one logarithmic factor. Nenadov [38] obtains $O(\log n)$ matching rounds on sufficiently strong spectral expanders, a condition our masks do not satisfy.

If only some values must be sent to prescribed positions, we complete this partial assignment to a permutation by pairing the unused inputs and unused outputs arbitrarily. The values that are not needed are simply ignored.

B.4 Proof of the exact representation theorem

We now combine the routing schedule with the copy and computation layers to prove exact representation.

Proof of Theorem 3.1. Let $A_n = I + C + R$ be built as in Section 3 from the graph of Lemma B.1 on $s = n/4$ vertices, so that equation 3.3 holds by Theorem 3.3, and put $S_n = 2 + 29\ \mathrm{rt}(\Gamma_n)$, which is $O(\log^2 n)$ by Lemma B.5. Let $\mathcal{O}$ be the set of even-position neurons. It contains exactly one endpoint of every edge of R, and $C^{-1}\mathcal{O}$ is disjoint from $\mathcal{O}$ because even- and odd-position neurons alternate in each cluster. Pad the bounded fan-in network to width $n/2 = |\mathcal{O}|$ and assign its neurons to $\mathcal{O}$ in a fixed order. The input layer places the input coordinates on the corresponding neurons of $\mathcal{O}$ and sets every other neuron to zero. This uses only copies of coordinates, zero biases, and ReLU applied to nonnegative values.

Invariant: the activations h before layer r of the bounded fan-in network hold $z^{(r-1)}$ on $\mathcal{O}$ and zero elsewhere. One layer copies the value at every $u \in \mathcal{O}$ to $R(u)$, using the connection I at u and R at $R(u)$; there are now two copies of each activation of the bounded fan-in network. Since each neuron of the bounded fan-in network is read at most twice, we can assign a separate copy to every use in the next layer of the bounded fan-in network. Since each neuron of the bounded fan-in network reads at most two inputs, we assign the values it reads to u and $C^{-1}u$, where $u \in \mathcal{O}$ is its neuron; these targets are distinct. This is a partial assignment of at most n copies to at most n positions; we complete it to a permutation and realise it by $\mathrm{rt}(\Gamma_n)$ matching rounds, each implemented by Lemma B.6 and padded with identity layers to exactly 29 layers. All these layers have zero bias and weights 0 or 1, so ReLU passes the nonnegative values unchanged.

In the final layer of the block, the neuron $u \in \mathcal{O}$ indexed by j reads h_u through I and $h_{C^{-1}u}$ through C, and computes

$$\mathrm{ReLU}\big(\theta_j h_u + \theta'_j h_{C^{-1}u} + \beta_{r,j}\big),$$

where θ_j, θ'_j are the corresponding entries of B_r (or zero for an unused input). Since each row of B_r has at most two permitted entries, these are all its inputs. All other neurons receive zero weights and zero bias. The activations again hold $z^{(r)}$ on $\mathcal{O}$ and zero elsewhere. The block has $1 + 29\ \mathrm{rt}(\Gamma_n) + 1 = S_n$ layers, independently of the bounded fan-in network, and every layer uses the mask A_n. By induction, the invariant accounts for every value kept in memory and every copy made by the bounded fan-in network.

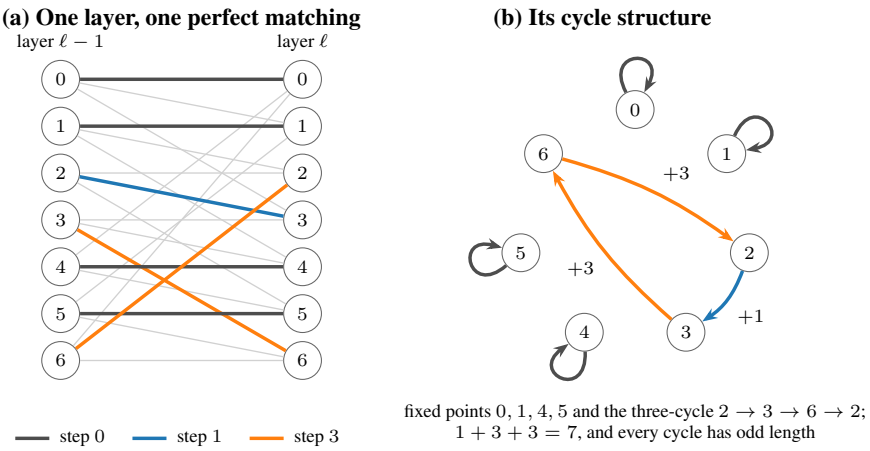


Figure 4: The width-seven Ramanujan mask of Proposition 3.4. (a) One layer: input neuron i connects to the output neurons i, $i + 1$ and $i + 3$ modulo seven (grey). Routing seven values along vertex-disjoint paths occupies every neuron, so each layer must use a perfect matching of this graph. One of its 24 perfect matchings is highlighted. (b) The highlighted matching consists of four fixed points and one three-cycle. The steps of a cycle lie in $\{1, 3\}$ and sum to a multiple of seven, which forces every cycle to have odd length. All 24 matchings, and hence all their compositions, are therefore even permutations, so an odd permutation, such as the transposition of neurons 0 and 1, is never routed.

After the T layers of the bounded fan-in network, one more block moves the values of the two neurons read by the output of the bounded fan-in network into two prescribed neurons. The copy and routing layers work as before, and the final layer keeps these two values while setting all other outputs to zero. The two-input output layer then produces the required affine output. If the output of the bounded fan-in network reads only one neuron, the second weight is zero. This gives exactly $L = 1 + (T + 1)S_n$ hidden layers. There are at most n input weights and n input biases, $3n$ weights and n biases in every later hidden layer, and three output parameters, which proves equation 2.5.

Apart from copies of the parameters of the bounded fan-in network, the only nonzero values used are ones. Negative weights appear only where the bounded fan-in network already has them. Hence neither the maximum magnitude nor the dyadic precision of the parameters increases. □

B.5 Mixing and the routing obstruction

We now separate pairwise reachability from simultaneous routing. The lemma shows what repeated spectral mixing guarantees; the proposition that follows exhibits a Ramanujan mask whose repeated layers still fail the routing test.

Lemma B.8 (Mixing). *For doubly stochastic $P_1, \dots, P_t$ with $\|P_\ell - J_n\|_2 \le \rho$,*

$$\|P_t \cdots P_1 - J_n\|_2 \le \rho^t, \tag{B.7}$$

and every entry of the product is positive when $\rho^t < 1/n$.

Proof of Lemma B.8. Set $E_\ell = P_\ell - J_n$. Since the P_ℓ are doubly stochastic, $J_n E_\ell = E_\ell J_n = 0$ and $J_n^2 = J_n$. Expanding the product gives

$$P_t \cdots P_1 = J_n + E_t \cdots E_1,$$

whose second term has operator norm at most $\prod_{\ell=1}^{t} \|E_\ell\|_2 \le \rho^t$. Each entry of the product differs from $1/n$ by at most this norm, so all entries are positive when $\rho^t < 1/n$. The unnormalised product counts directed paths, so this is a statement about reaching each output from each input individually. □

Proof of Proposition 3.4. The differences of distinct elements of $\{0, 1, 3\}$ run through all six nonzero residues modulo seven exactly once. Hence any two distinct rows of A share exactly one position with a one. Each row has three ones, which proves $AA^\top = 2I + \mathbf{1}\mathbf{1}^\top$ and the singular-value claim.

Let π be a permutation supported by A, i.e. $\pi(i) - i \in \{0, 1, 3\}$ for all i. Along a nontrivial cycle of π, every step is 1 or 3. For a cycle of length k with b steps equal to three, the total $k + 2b$ must be a multiple of seven. For $k = 2$ the total is even and lies between two and six; for $k = 4$ it is even and lies between four and twelve. Neither range contains an even multiple of seven. For $k = 6$ the total must be fourteen, so there are four steps equal to three and two equal to one. Cyclically, the four threes fall into the two gaps between the ones, so some one is followed by at least two consecutive threes. These three steps sum to seven and return to the starting vertex before the cycle is complete, a contradiction. Hence every nontrivial cycle of π has odd length, and π is an even permutation.

Seven vertex-disjoint paths through a stack of width seven occupy every vertex of every intermediate layer. Each layer therefore uses a permutation supported by A. Their composition is even and cannot equal a prescribed odd permutation. Finally, $\sqrt{2}/3 < 2\sqrt{2}/3$ proves the Ramanujan claim. The argument concerns routing along vertex-disjoint paths, not arbitrary weighted computations. □

Remark B.9 (Relation to Paley constructions). The nonzero squares modulo 7 are $\{1, 2, 4\} = \{0, 1, 3\} + 1$. Up to a cyclic shift of the rows, A is therefore the quadratic-residue (Paley) circulant of order 7, the incidence matrix of the Fano plane. The shift is a 7-cycle, hence an even permutation, so every permutation supported by the quadratic-residue form is also even, and the obstruction holds for that form as well.

For completeness, connecting sets without prescribing which input goes where does not suffice either, already for two neurons. The two layers

$$X = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad Y = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

connect every input to every output and give two disjoint paths for the identity. The two paths required by the transposition, however, both pass through the first middle neuron. This small network therefore has the superconcentrator property for two neurons but cannot realise every permutation.

B.6 A COMPUTATIONAL SANITY CHECK

To make the construction concrete, we instantiated the masks A_n for $n = 32, \ldots, 4096$. For $n = 32$ the auxiliary graph is $K_{4,4}$; each larger auxiliary graph is a bipartite 2-lift of the previous one whose new eigenvalues have absolute value at most $2\sqrt{3}$, so every instance satisfies the conclusion of Lemma B.1. Each mask was oriented along an Euler circuit and ordered as in Section 3. Every mask is binary and three-regular, and $\sigma_2(\frac{1}{3}A_n)$ lies below the bound equation 3.2 for its γ, and hence below ρ^* (Table 1). As n grows, the spectral gap levels off near 0.015, consistent with a width-independent gap, whereas the gap of the best two-regular mask vanishes like n^{-2} (Figure 5, Proposition B.4). We also enumerated all 7! permutations for the mask of Proposition 3.4: exactly 24 are supported by one layer, and all of them are even. These computations are illustrations only; all results are proved analytically.

C INTERPOLATION AND THE BATCHED BOUNDED FAN-IN NETWORK

This appendix proves Lemma 4.1, constructs the batched bounded fan-in network behind Theorem 4.2, and derives the universality and Hölder consequences. We begin with the geometry of the Kuhn triangulation and its hat functions.

C.1 PROOF OF THE PARTITION OF UNITY

Proof of Lemma 4.1. Nonnegativity is immediate. If $\psi_v(x) > 0$, both maxima in equation 4.1 are strictly less than one, so $|Kx_i - v_i| < 1$ for every coordinate. This proves the support claim.

For the sum, first let $Kx = a + r$ lie in the interior of a grid cell, with $a \in \{0, \ldots, K - 1\}^d$ and $0 < r_i < 1$. Every hat that can be nonzero at x belongs to a vertex $v = a + \mathbf{1}_T$ of this cell, with

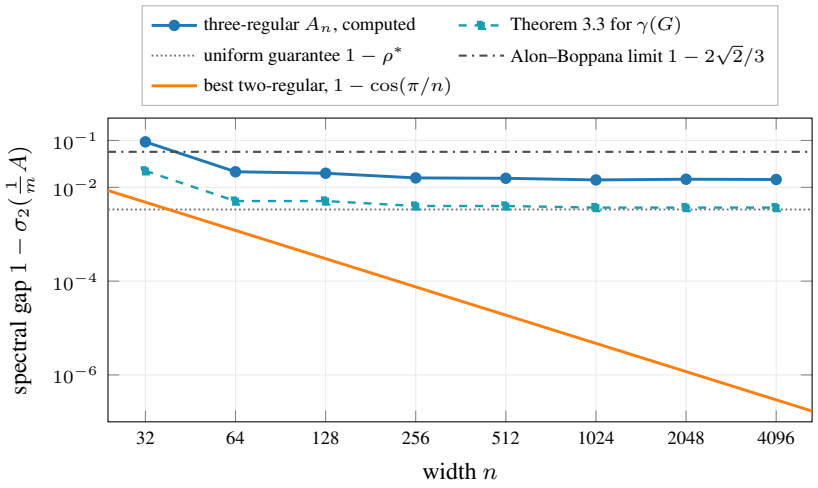


Figure 5: Spectral gaps of the masks A_n compared with two-regular masks (log–log scale). For each width n, one mask A_n was built from $K_{4,4}$ ($n = 32$) or from a bipartite Ramanujan 2-lift of the previous auxiliary graph, and $1-\sigma_2(\frac{1}{3}A_n)$ was computed directly (blue). It stays above the gap guaranteed by Theorem 3.3 for the Laplacian gap $\gamma(G)$ of each auxiliary graph (teal) and above the uniform guarantee $1-\rho^*$ (dotted), and it levels off near 0.015. By the Alon–Boppana bound [39], no three-regular mask family can keep a gap above $1-2\sqrt{2}/3 \approx 0.057$ as n grows (dash-dotted; small masks such as $n = 32$ may exceed it), so every three-regular family has spectral parameter close to one. The best two-regular mask, a single cycle, has gap $1-\cos(\pi/n) \approx \pi^2/(2n^2)$ (Proposition B.4; orange), which is smaller at every width and vanishes as n grows. The values are listed in Table 1.

width n	32	64	128	256	512	1024	2048	4096
$\gamma(G)$	4.000	0.764	0.764	0.592	0.592	0.546	0.546	0.546
$\sigma_2(\frac{1}{3}A_n)$	0.907	0.979	0.980	0.984	0.984	0.986	0.985	0.985
bound equation 3.2	0.978	0.995	0.995	0.996	0.996	0.996	0.996	0.996
gap of A_n	0.093	0.022	0.020	0.016	0.016	0.014	0.015	0.015
gap of cycle	$4.8{\cdot}10^{-3}$	$1.2{\cdot}10^{-3}$	$3.0{\cdot}10^{-4}$	$7.5{\cdot}10^{-5}$	$1.9{\cdot}10^{-5}$	$4.7{\cdot}10^{-6}$	$1.2{\cdot}10^{-6}$	$2.9{\cdot}10^{-7}$

Table 1: Laplacian gap $\gamma(G)$ of the auxiliary graph, spectral parameter of one mask A_n at each width, and the bound of Theorem 3.3 for that $\gamma(G)$; every value lies below the uniform bound $\rho^* \approx 0.9966$. The last two rows compare the spectral gap $1-\sigma_2(\frac{1}{3}A_n)$ with that of the best two-regular mask of the same width, a single cycle with gap $1-\cos(\pi/n)$ (Proposition B.4). The instances were generated by the script in the supplementary material.

$T \subseteq [d]$. Order the coordinates so that

$$r_{\pi(1)} \geq r_{\pi(2)} \geq \cdots \geq r_{\pi(d)}.$$

For nonempty proper T, the hat equals

$$\mathrm{ReLU}\left(\min_{i\in T} r_i - \max_{i\notin T} r_i\right).$$

It vanishes unless T is an initial segment of this ordering; when there are ties, any other set that remains possible still gives zero. At the successive initial segments, including the empty and the full set, the values are

$$1-r_{\pi(1)}, \quad r_{\pi(1)}-r_{\pi(2)}, \quad \ldots, \quad r_{\pi(d-1)}-r_{\pi(d)}, \quad r_{\pi(d)},$$

which sum to one. All hats are continuous and there are finitely many, so the identity extends to cell boundaries, the boundary of the cube, and ties.

Finally, for every vertex with positive weight, $|c_v - f(x)| \leq \delta + \omega_f(1/K)$. Taking the convex combination proves equation 4.3. The same convexity gives coefficient stability and preservation of the range. □

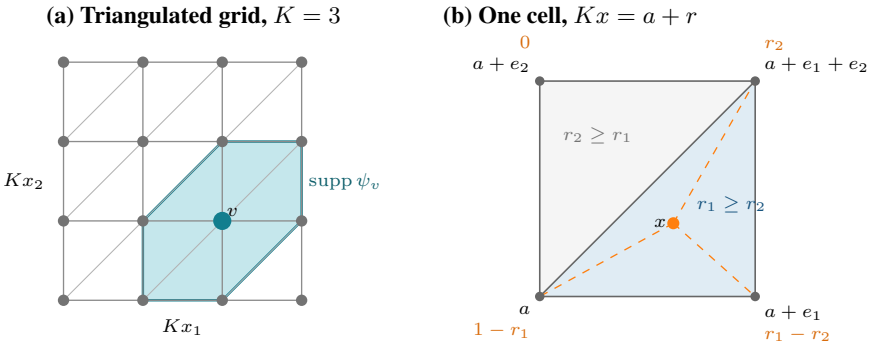


Figure 6: Hat functions of the Kuhn triangulation in two dimensions. (a) Each grid cell is split along its main diagonal into two simplices. The hat function ψ_v of equation 4.1 equals one at v, vanishes at every other grid point and is linear on each simplex; its support (shaded) is the hexagon formed by the six simplices that contain v. (b) Write $Kx = a + r$ with a a grid point and $r \in [0, 1]^2$. The ordering of the coordinates of r selects the simplex containing x. If $r_1 \geq r_2$, only the hats at a, $a + e_1$ and $a + e_1 + e_2$ are nonzero at x, with the values shown in orange: $1 - r_1$, $r_1 - r_2$ and r_2. These correspond to the successive initial segments in the proof of Lemma 4.1, and they sum to one.

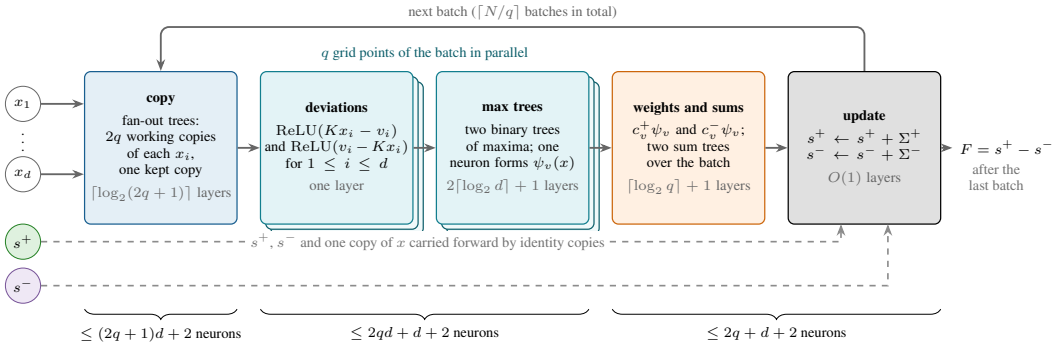


Figure 7: The batched bounded fan-in network of Lemma C.1, which evaluates the interpolant $\sum_v c_v \psi_v$ in $\lceil N/q \rceil$ batches of q grid points. In each batch, fan-out trees copy the input coordinates, the q hat functions are evaluated in parallel by binary trees of maxima, and sum trees add their weighted contributions to two nonnegative partial sums s^+ and s^-, so that every activation stays nonnegative. The partial sums and one copy of the input are carried to the next batch by identity copies (dashed). Each box gives the depth of its stage, and the braces give the largest number of neurons in each group of stages, all below $8q(d+1)$. After the last batch, the output layer computes $s^+ - s^-$.

C.2 A BOUNDED FAN-IN NETWORK OF BOUNDED WIDTH

We next implement the interpolant exactly. Evaluating q grid hats per batch controls the tradeoff between network width and depth.

Lemma C.1 (Batched exact interpolation). *Let $N = (K+1)^d$ and $1 \leq q \leq N$. For arbitrary real coefficients c_v, the interpolant $\sum_v c_v \psi_v$ is computed exactly by a bounded fan-in network of width at most $8q(d+1)$ and depth*

$$T \leq C_0 \lceil N/q \rceil \left(1 + \lceil \log_2(d+1) \rceil + \lceil \log_2(q+1) \rceil\right). \tag{C.1}$$

Its parameters and activations satisfy equation 4.6.

Proof. Keep the d input coordinates and two partial sums s^+, s^-, initially zero. Process the grid vertices in disjoint batches of at most q, filling the last batch with dummy vertices whose coefficients are zero.

At the start of a batch, make $2q$ working copies of every coordinate and keep one more copy in memory. A binary fan-out tree with $2q+1$ leaves and height at most $\lceil \log_2(2q+1) \rceil$ suffices; branches that finish early are carried forward by identity copies. Each neuron is read at most twice per layer, and at most $2q+1$ copies of a coordinate exist in any layer. The memory copy is kept until the end of the batch.

For each vertex in the batch, compute the $2d$ nonnegative deviations in equation 4.1. The two maxima are computed by balanced binary trees, since for nonnegative a, b the two-layer map

$$(a, b) \longmapsto (\mathrm{ReLU}(a-b), b) \longmapsto \mathrm{ReLU}(\mathrm{ReLU}(a-b)+b) = \max\{a, b\}$$

has fan-in and fan-out at most two. At each level of the tree an unpaired value is simply kept. Both maxima take at most $2\lceil \log_2 d \rceil$ layers (none when $d = 1$). One more neuron forms the hat.

Write $c_v = c_v^+ - c_v^-$ with $c_v^\pm \geq 0$. Each hat feeds the two products $c_v^+ \psi_v$ and $c_v^- \psi_v$. Separate sum trees for the positive and the negative products have depth at most $\lceil \log_2 q \rceil$ and contain only nonnegative values. Their roots are added to s^+ and s^-. The updated partial sums and the input coordinates are kept for the next batch, and the working neurons are reused. The output is the affine function $s^+ - s^-$.

The width is checked as follows. Copying the coordinates needs at most $(2q+1)d+2$ neurons, the deviations and maxima at most $2qd+d+2$, and the products and sums at most $2q+d+2$. The first level of each maximum uses two neurons per pair, no more than the pair already occupies. All these counts are below $8q(d+1)$. Neuron positions can be fixed in advance, with unused neurons set to zero, and every step respects fan-in and fan-out at most two. Counting layers, including a constant number of initialisation and partial sum steps per batch, gives equation C.1.

Only the weights and biases of the deviations have magnitude up to K; all other coefficients are 0, ± 1 or $c_v^\pm$. Deviations and intermediate maxima lie in $[0, K]$. At any x, the positive partial sum over any set of processed vertices satisfies

$$\sum_{v \text{ in the set}} c_v^+ \psi_v(x) \leq \max_v |c_v| \sum_v \psi_v(x) = \max_v |c_v|,$$

and similarly for the negative partial sum. Hence both partial sums, and not only the final output, satisfy equation 4.6. Copies and zero padding do not increase this bound. □

Proof of Theorem 4.2. The condition on q gives $8q(d+1) \leq n/2$. Pad the bounded fan-in network of Lemma C.1 to width $n/2$ and apply Theorem 3.1. The first and last blocks are absorbed into the constant in equation 4.5, since all logarithmic factors are at least one. Exact representation preserves the error bound of Lemma 4.1, the parameter bound and the activation bound. The bound equation 2.5 counts all parameters of the architecture, whether or not the network uses them. □

Proof of Corollary 4.3. Let $f \in C([0,1]^d)$ and $\varepsilon > 0$. By uniform continuity, choose K with $\omega_f(1/K) \leq \varepsilon$, and apply Theorem 4.2 with $c_v = f(v/K)$, $\delta = 0$ and $q = 1$, which is admissible because $n \geq 16(d+1)$. The resulting network uses a finite prefix of $(A_n, A_n, \ldots)$ and satisfies $\|F - f\|_\infty \leq \varepsilon$. The L^p statement follows from $\|g\|_{L^p([0,1]^d)} \leq \|g\|_\infty$. □

Corollary C.2 (Hölder targets with per-parameter storage)**.** *Suppose* $0 < \beta \leq 1$*,* $H \geq 0$*,* $\|f\|_\infty \leq M$*, and*

$$|f(x) - f(y)| \leq H \|x - y\|_\infty^\beta .$$

For $0 < \varepsilon \leq 1$*, choose*

$$K = \max\{1, \lceil (2H/\varepsilon)^{1/\beta} \rceil\}, \qquad N = (K+1)^d. \tag{C.2}$$

Theorem 4.2 gives uniform error at most ε*. All parameters can be chosen dyadic, bounded in magnitude by* $\max\{1, K, M+1\}$*, and stored separately with at most*

$$b \leq C + \lceil \log_2(2 + K + M) \rceil + \lceil \log_2(2/\varepsilon) \rceil \tag{C.3}$$

bits per parameter.

Proof of Corollary C.2. For the chosen K, the interpolation error is at most $HK^{-\beta} \le \varepsilon/2$, also when $H = 0$. Put $t = \lceil \log_2(1/\varepsilon) \rceil$ and round each $f(v/K)$ to the nearest multiple of 2^{-t}. The rounding error is at most $2^{-t-1} \le \varepsilon/2$, and each coefficient has magnitude at most $M + 1$. By Lemma 4.1, the total effect of the coefficient errors is bounded by the same tolerance, without a factor N.

The other parameters of the bounded fan-in network are integers of magnitude at most $\max\{1, K\}$, and the construction of Theorem 3.1 adds only zeros and ones. A sign bit, enough integer bits for the largest magnitude, and t fractional bits give equation C.3. □

D Bounded activations and encoding length

This appendix proves the bounded-weight construction, the delta-coding upper bound and the metric-entropy lower bound of Section 5.

D.1 Exact clipped doubling iterations

Proof of Theorem 5.1. For $z \in [0, 1]$, let $D(z) = \min\{2z, 1\}$. It is computed by the two-layer ReLU map

$$z \longmapsto \big(\mathrm{ReLU}(2z), \mathrm{ReLU}(2z - 1)\big) \longmapsto \mathrm{ReLU}\big(\mathrm{ReLU}(2z) - \mathrm{ReLU}(2z - 1)\big) = D(z). \quad \text{(D.1)}$$

Its coefficients are 0, 1, -1 and 2; the input is read twice and the output reads two neurons. Every intermediate activation is at most two. By induction, $D^{\circ k}(z) = \min\{2^k z, 1\}$ for every $k \ge 0$.

Replace the two deviations of coordinate i in equation 4.1 by

$$u_i^+ = D^{\circ k}(\mathrm{ReLU}(x_i - v_i/K)), \qquad u_i^- = D^{\circ k}(\mathrm{ReLU}(v_i/K - x_i)).$$

Their initial arguments lie in $[0, 1]$, and they equal the original deviations clipped at one. For non-negative a, b,

$$\mathrm{ReLU}(1 - a - b) = \mathrm{ReLU}(1 - \min\{a, 1\} - \min\{b, 1\}),$$

and taking a maximum commutes with clipping at one. Hence the hat computed from $u_i^{\pm}$ is *exactly* ψ_v.

The fan-out trees, maxima, products and partial sums are as in Lemma C.1. Applying the doublings to the $2qd$ deviations in parallel adds $2k$ layers to the bounded fan-in network. The first layer of each doubling needs at most $4qd + d + 2$ neurons, still below $8q(d + 1)$. The stored coordinates and the partial sums occupy separate neurons and are kept without extra reads. This gives the width and depth needed for equation 5.1.

The only new biases are $\pm v_i/K$, which have magnitude at most one and at most k fractional bits. Every activation in a doubling is at most two, the subsequent maxima are at most one, and the weighted partial sums are at most M_0 by the partition of unity. Hence every activation of the bounded fan-in network and of the masked network is bounded by $\max\{2, M_0\}$, and so is every parameter. The remaining coefficients have at most t fractional bits, and the representation does not increase the precision. □

D.2 Coding the coefficients

With the interpolation network in place, we use the Hölder regularity of the target to encode neighboring coefficients compactly.

Lemma D.1 (A nearest-neighbour ordering of the grid)**.** *The vertices of $\{0, \dots, K\}^d$ can be ordered as $v^{(1)}, \dots, v^{(N)}$ so that consecutive vertices differ by one in exactly one coordinate.*

Proof. For $d = 1$, use the increasing order. Given such an order in dimension $d - 1$, traverse the slices of the last coordinate from 0 to K, using the given order on even slices and its reverse on odd slices. The last vertex of one slice and the first vertex of the next then agree in the first $d - 1$ coordinates, and within each slice the induction hypothesis applies. Every vertex is visited exactly once. □

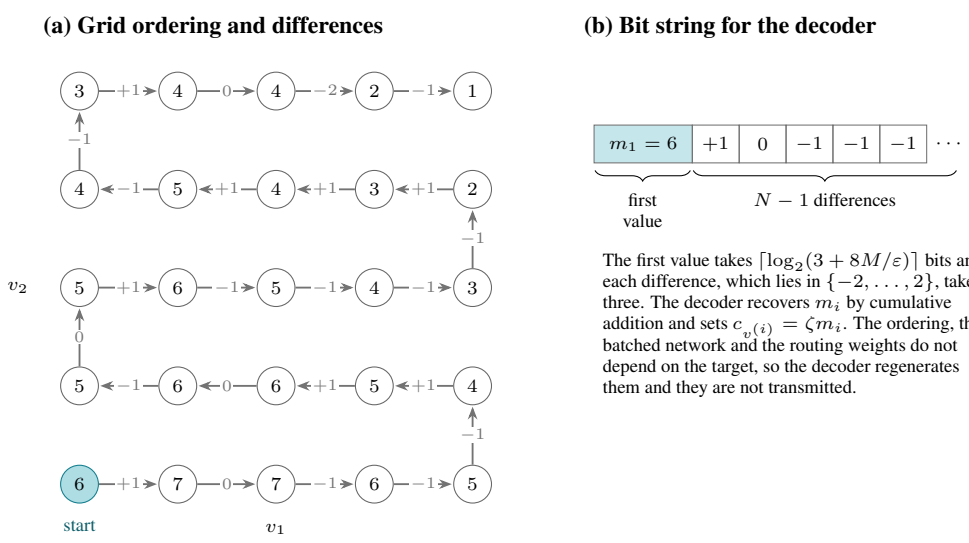


Figure 8: Delta coding of the interpolation coefficients (Theorem 5.2), illustrated for $d = 2$ and $K = 4$. (a) The ordering of Lemma D.1 traverses the rows of the grid alternately forwards and backwards, so consecutive grid points differ by one step in one coordinate. Each node shows the rounded coefficient $m_i = \mathrm{round}(f(v^{(i)}/K)/\zeta)$, and each arrow the difference $m_i - m_{i-1}$. By the Hölder condition and the choice of K and ζ, every difference lies in $\{-2, \dots, 2\}$. (b) Only the first value and the differences are transmitted, which gives the encoding length equation 5.2.

Proof of Theorem 5.2. Let $t = \lceil \log_2(2/\varepsilon) \rceil$ and $\zeta = 2^{-t}$. Then

$$\varepsilon/4 < \zeta \le \varepsilon/2. \tag{D.2}$$

Order the grid as in Lemma D.1 and set

$$m_i = \mathrm{round}(f(v^{(i)}/K)/\zeta), \qquad c_{v^{(i)}} = \zeta m_i,$$

with a fixed rule for ties. Rounding contributes at most $\zeta/2 \le \varepsilon/4$ and interpolation at most $\varepsilon/2$, so the total error is at most $3\varepsilon/4$.

For consecutive vertices, the Hölder condition and the choice of K give

$$\zeta |m_i - m_{i-1}| \le |f(v^{(i)}/K) - f(v^{(i-1)}/K)| + \zeta \le \varepsilon/2 + \zeta.$$

By equation D.2, $|m_i - m_{i-1}| < 3$, and since the difference is an integer it lies in $\{-2, -1, 0, 1, 2\}$. We encode it with three bits. The first integer satisfies $|m_1| \le \lceil M/\zeta \rceil$, so it can be encoded with at most

$$\lceil \log_2(2\lceil M/\zeta \rceil + 1) \rceil \le \lceil \log_2(3 + 8M/\varepsilon) \rceil$$

bits. This proves equation 5.2.

The decoder recovers the integers by cumulative addition and the coefficients by multiplying by the known ζ. It also reconstructs the grid ordering, the batched bounded fan-in network and the routing, none of which depends on the coefficient values; for every routing block it uses the first schedule of $\mathrm{rt}(\Gamma_n)$ matching rounds in a fixed enumeration, which exists by Lemma B.5. The connections for both positive and negative coefficients are permitted in advance, even when their values are zero. All these parameters are fixed functions of d, K and the architecture, so no target-specific bits are needed for them. Bit strings containing unused three-bit symbols are mapped to a fixed default network, so the decoder is defined on every input.

All decoded coefficients have magnitude at most $M + \zeta/2 \le M + 1$, so the earlier parameter and activation bounds apply. Rounding K up to a power of two only decreases the differences between neighbouring values of f and the interpolation error; the same five-value alphabet suffices, and Theorem 5.1 applies. The decoder performs the additions before the network is run, so they add no layers and no arithmetic error to the network. □

D.3 A PACKING LOWER BOUND

We now show that the encoding order above is necessary for every decoder, using a packing argument.

The *bits per parameter* b is a number of bits that suffices to store each parameter separately; since parameters can be correlated, $\mathcal{B}$ may be much smaller than bP.

Theorem D.2 (Metric-entropy lower bound)**.** *Let $d \geq 1$, $0 < \beta \leq 1$, and $H, M > 0$. Suppose a decoder with $\mathcal{B}$-bit inputs, with any output representation, approximates every member of $\mathcal{H}_{\beta,H,M}$ to uniform error at most ε. If*

$$(4\varepsilon/H)^{1/\beta} \leq 1/8, \qquad 4\varepsilon \leq M,$$

then

$$\mathcal{B} \geq 4^{-d-d/\beta}(H/\varepsilon)^{d/\beta}. \tag{D.3}$$

In particular, a fixed architecture with P separately stored parameters, each taking at most 2^b values, must satisfy the same bound with $\mathcal{B} = bP$.

Proof of Theorem D.2. Put $r = (4\varepsilon/H)^{1/\beta}$ and $k = \lfloor 1/(2r) \rfloor$. Choose the k^d centres whose coordinates are of the form $(2i-1)r$, $1 \leq i \leq k$. The open ℓ^∞ balls of radius r around them are disjoint and lie in the cube. For each centre a, define the bump

$$u_a(x) = H(r - \|x - a\|_\infty)_+^\beta.$$

The distance function and the positive part are 1-Lipschitz, and $|s^\beta - t^\beta| \leq |s - t|^\beta$ for $s, t \geq 0$, so u_a has Hölder constant at most H. For any set of centres, take the maximum of the corresponding bumps and zero. A maximum of functions with Hölder constant H again has Hölder constant H (compare the maximising term at one point with its value at the other, then exchange the points). Each such function has sup norm at most $Hr^\beta = 4\varepsilon \leq M$.

There are 2^{k^d} such functions. Two different ones differ by 4ε at a centre that belongs to exactly one of the two sets, since all other bumps vanish there. By the triangle inequality, no single function approximates two of them within ε. A decoder with $\mathcal{B}$ input bits has at most $2^{\mathcal{B}}$ outputs, so $\mathcal{B} \geq k^d$. Since $r \leq 1/8$, we have $k \geq 1/(4r)$, which gives equation D.3. The statement for separately stored parameters follows because P parameters with at most 2^b values each give at most 2^{bP} networks. □

E PROOFS FOR COMPOSITIONAL TARGETS

E.1 THE COMPOSITIONAL CLASS AND THE MAIN ESTIMATE.

This appendix states and proves the compositional result summarised in Section 5.

Definition E.1 (Compositional function class)**.** Let h, p, s be positive integers and $\lambda_1, \ldots, \lambda_h \geq 0$. A *compositional target* has h levels, with $d_j \leq p$ variables at level $0 \leq j \leq h$, where $d_0 = d$ (so $p \geq d$) and $d_h = 1$. Starting from $z^{(0)} = x$, each variable at level $j \geq 1$ is $z_k^{(j)} = g_{j,k}(z_{I_{j,k}}^{(j-1)})$, where $[m] = \{1, \ldots, m\}$ and $I_{j,k} \subseteq [d_{j-1}]$ is a nonempty set of at most s variables of the previous level and $g_{j,k} : [0,1]^{|I_{j,k}|} \to [0,1]$ is λ_j-Lipschitz in the ℓ^∞ norm. The target is $f = z_1^{(h)}$, and s is its *effective input dimension*.

Theorem E.2 (Compositional approximation on fixed masks)**.** *For a target as in Definition E.1, choose integers $K_j \geq 1$ and coefficient tolerances $\tau_j \geq 0$. Put*

$$e_j = \lambda_j/K_j + \tau_j, \qquad a_j = \prod_{k=j+1}^{h} \lambda_k,$$

with the empty product equal to one. There is a realisation on the fixed masks with

$$\|F - f\|_\infty \le \sum_{j=1}^{h} a_j e_j, \qquad n \le Cp(s+1), \qquad P \le 4nL + 3, \tag{E.1}$$

$$L \le C \log_2^2(p(s+1)+1) \sum_{j=1}^{h} \left[1 + \log_2(p+1) + (K_j+1)^s \log_2(s+1)\right]. \tag{E.2}$$

Every approximated component takes values in $[0,1]$*. A variable may be used by any number of components; the copies this requires are included in equation E.2.*

For 1-Lipschitz components and accuracy ε, the choice $K_j = \lceil 2h/\varepsilon \rceil$ gives depth of order $h(2h/\varepsilon)^s$ up to logarithmic factors; for contractive components, K_j can be chosen independently of h (Corollary E.4). The exponent of $1/\varepsilon$ is therefore s rather than d.

We first show how errors propagate through a Lipschitz composition, and then schedule the components on a common width.

E.2 Range preservation and error propagation

Lemma E.3 (Error propagation through a Lipschitz composition)**.** *Let* $g_j : [0,1]^{d_{j-1}} \to [0,1]^{d_j}$ *be* λ_j*-Lipschitz in* ℓ^∞*, and let* $\widetilde{g}_j$ *have the same domain and range with* $\|g_j - \widetilde{g}_j\|_\infty \le e_j$*. Then*

$$\|g_h \circ \cdots \circ g_1 - \widetilde{g}_h \circ \cdots \circ \widetilde{g}_1\|_\infty \le \sum_{j=1}^{h} e_j \prod_{k=j+1}^{h} \lambda_k.$$

Proof. The exact and approximate intermediate values both stay in the relevant cubes. If E_j is their largest difference after level j, then

$$E_j \le \lambda_j E_{j-1} + e_j, \qquad E_0 = 0 :$$

compare the exact map at the two inputs, then the exact and approximate maps at the approximate input. Expanding the recursion proves the claim. No Lipschitz bound on the approximate maps is needed. □

E.3 Scheduling the components

Proof of Theorem E.2. At level j, approximate each $g_{j,k}$ by the hat construction in its own dimension $s_{j,k} = |I_{j,k}|$, with grid size K_j. Choose its coefficients in $[0,1]$ within τ_j of the true values; for positive tolerance, round and then clip to $[0,1]$, which cannot increase the error because the true values lie in $[0,1]$. Lemma 4.1 gives an approximation $\widetilde{g}_{j,k}$ with values in $[0,1]$ and uniform error at most $\lambda_j/K_j + \tau_j = e_j$. The vector map g_j is λ_j-Lipschitz because restricting coordinates does not increase ℓ^∞ distances. Lemma E.3 proves equation E.1.

We now count the network size, instead of assuming that each component's inputs are freely available. At level j there are at most ps pairs (component, variable it reads). Using binary fan-out trees, make one copy of each variable for every component that reads it; variables that are not read are discarded. A given variable is read by at most p components, since level j has at most $d_j \le p$ components, so the trees have height at most $\lceil \log_2(p+1) \rceil$, and at most $p + ps$ neurons are needed, including the variables still being kept. Branches that finish early are carried forward until all inputs are ready.

For each of the at most p components, run the $q = 1$ network of Lemma C.1 on its own set of neurons, so its inputs do not compete with those of other components. Its width is at most $8(s_{j,k}+1)$ and its depth at most

$$C(K_j+1)^{s_{j,k}} \log_2(s_{j,k}+1) \le C(K_j+1)^s \log_2(s+1).$$

Run these networks in parallel, keeping the output of any network that finishes early. The coefficients are nonnegative, so the negative partial sum is not needed and each output is already a nonnegative activation in $[0,1]$. Copy these outputs into the next level and reset the unused neurons.

All levels fit into a common width $Cp(s+1)$, and the total depth of the bounded fan-in network is at most

$$C\sum_{j=1}^{h}\left[1+\log_2(p+1)+(K_j+1)^s\log_2(s+1)\right].$$

Padding the width so that twice it is an admissible power of two changes it by a constant factor only. Theorem 3.1 then proves the width bound in equation E.1 and equation E.2, with all copies, kept outputs and initial layers included. The output layer reads the final activation with weight one. □

Corollary E.4 (1-Lipschitz components)**.** *If every $\lambda_j \le 1$ and $0 < \varepsilon \le 1$, take*

$$K_j = \lceil 2h/\varepsilon \rceil, \qquad \tau_j \le \varepsilon/(2h).$$

Then $\|F-f\|_\infty \le \varepsilon$, $n = O(p(s+1))$, and

$$L \le Ch\left[1+\log_2(p+1)+(\lceil 2h/\varepsilon\rceil+1)^s\log_2(s+1)\right]\log_2^2(p(s+1)+1). \tag{E.3}$$

If instead $\lambda_j \le \lambda < 1$ for all j, one may take $K_j = \max\{1, \lceil 2\lambda/((1-\lambda)\varepsilon)\rceil\}$ and $\tau_j \le (1-\lambda)\varepsilon/2$, independently of h.

Proof of Corollary E.4. When all components are 1-Lipschitz, $a_j \le 1$ and $e_j \le \varepsilon/h$, so the sum in equation E.1 is at most ε. Substituting the common K_j into equation E.2 gives equation E.3.

In the contractive case, the stated K_j and τ_j give $e_j \le (1-\lambda)\varepsilon$; this also holds for $\lambda = 0$, since a constant component is interpolated without error. Moreover $a_j \le \lambda^{h-j}$, and the geometric sum is at most $1/(1-\lambda)$, so the total error is at most ε. □

F Proof of the stability bound

Here $\|B\|_\infty = \max_j \sum_i |B_{ji}|$ is the maximum absolute row sum, the operator norm induced by the maximum norm. We separate amplification within the computation from the additional error accumulated while values are copied and routed.

For a bounded fan-in network equation 2.4, write

$$\alpha_r = \max\{1, \|B_r\|_\infty\}, \qquad \Lambda_T = \sum_{r=0}^{T}\prod_{k=r+1}^{T}\alpha_k. \tag{F.1}$$

Let $U \ge 1$ bound every activation of the exact bounded fan-in network on the cube, and let A_{out} be the ℓ^1 norm of its two output weights. Exact copying and routing preserve the bound U in the masked network.

Theorem F.1 (Stability, with routing counted separately)**.** *Build the masked network as in Theorem 3.1, with $S = S_n$ and $L = 1 + (T+1)S$. Perturb every hidden weight and bias permitted by the mask, every input-layer parameter, and the two output weights and the output bias, each by at most $\eta \ge 0$, keeping entries outside the mask zero. If $\widetilde{F}$ is the perturbed network, then*

$$\left\|\widetilde{F}-F\right\|_\infty \le (A_{\mathrm{out}}+2\eta)\eta(3U+1)e^{3\eta L}(1+S\Lambda_T)+\eta(2U+1). \tag{F.2}$$

If $\|B_r\|_\infty \le 1$ for all r and $3\eta L \le 1$, this is at most

$$e(A_{\mathrm{out}}+2\eta)\eta(3U+1)L+\eta(2U+1). \tag{F.3}$$

The factor Λ_T measures the amplification by the bounded fan-in network itself; routing contributes only the factor $S = O(\log^2 n)$ and the term $e^{3\eta L}$, which is close to one for small perturbations. Many approximation networks, such as those summing many terms, have $\alpha_r > 1$, so equation F.3 does not apply to them directly.

Proof of Theorem F.1. Fix an input x. Let h_ℓ and $\widetilde{h}_\ell$ be the exact and perturbed activations of the masked network, and put $E_\ell = \left\|\widetilde{h}_\ell - h_\ell\right\|_\infty$. The exact activations are bounded by U: copy and

routing layers only duplicate or permute activations of the bounded fan-in network, the final layer of each block computes the next layer of the bounded fan-in network, and the last block only selects neurons.

Write $W'_\ell = W_\ell + \Delta W_\ell$ and $b'_\ell = b_\ell + \Delta b_\ell$. Each hidden row has at most three permitted entries, and each row of the input layer at most one. Hence $\|\Delta W_\ell\|_\infty \le 3\eta$ and $\|\Delta b_\ell\|_\infty \le \eta$. Since ReLU is 1-Lipschitz, the identity

$$W'_\ell \widetilde{h}_{\ell-1} - W_\ell h_{\ell-1} = W'_\ell(\widetilde{h}_{\ell-1} - h_{\ell-1}) + \Delta W_\ell h_{\ell-1}$$

gives

$$E_\ell \le (\mu_\ell + 3\eta)E_{\ell-1} + \eta(3U+1), \qquad E_0 = 0, \tag{F.4}$$

where $\mu_\ell = 1$ for the input layer and for every copy, routing and output-placement layer, and $\mu_\ell = \alpha_r$ for the layer that computes layer r of the bounded fan-in network. In all cases $\|W_\ell\|_\infty \le \mu_\ell$ and $\mu_\ell \ge 1$.

Since $\mu_\ell + 3\eta \le \mu_\ell(1+3\eta)$, unrolling equation F.4 gives

$$E_L \le \eta(3U+1)e^{3\eta L} \sum_{\ell=1}^{L} \prod_{k=\ell+1}^{L} \mu_k. \tag{F.5}$$

This sum can be computed exactly. Put $G_r = \prod_{k=r+1}^{T} \alpha_k$ for $0 \le r \le T$, so that $G_T = 1$ and $\Lambda_T = \sum_{r=0}^{T} G_r$. The input layer contributes G_0. In the block that computes layer r of the bounded fan-in network, each of the $S-1$ layers before the computing layer contributes G_{r-1}, and the computing layer contributes G_r. The last block contributes S terms equal to one. Hence

$$\begin{aligned}\sum_{\ell=1}^{L} \prod_{k=\ell+1}^{L} \mu_k &= G_0 + (S-1)\sum_{r=1}^{T} G_{r-1} + \sum_{r=1}^{T} G_r + S \\ &= 1 + S\Lambda_T,\end{aligned}$$

which also holds for $T = 0$.

Only the two prescribed output weights may be nonzero. Writing a for this pair, the perturbation satisfies $\|\Delta a\|_1 \le 2\eta$ and $|\Delta c| \le \eta$. Therefore

$$|\widetilde{F}(x) - F(x)| \le (A_{\text{out}} + 2\eta)E_L + \eta(2U+1).$$

Together with equation F.5, this gives equation F.2, uniformly in x. The proof never assumes that the perturbed activations are bounded. If all $\|B_r\|_\infty \le 1$, then $\Lambda_T = T+1$ and $1 + S\Lambda_T = L$; if moreover $3\eta L \le 1$, replacing $e^{3\eta L}$ by e gives equation F.3. □

G Representation of general sparse and dense networks

Theorem 3.1 applies to bounded fan-in networks, i.e. ReLU networks with fan-in and fan-out at most two. This appendix shows how a general ReLU network is converted into a bounded fan-in network, and counts the cost exactly.

G.1 A bound in terms of the number of connections

Proof of Corollary 3.2. Pad the activations before and after each layer to width w. Consider a layer $y = \mathrm{ReLU}(Az + b)$ with nonnegative z. If z_i is used c_i times by A, build a binary fan-out tree with c_i leaves (no tree if $c_i = 0$). Its height is at most $\lceil \log_2(\Delta_r + 1) \rceil$; leaves that are ready early are carried forward. At any time there are at most E_r leaves or intermediate copies, besides at most w activations kept during the copying.

At each leaf, compute $|A_{ji}|z_i$ and send it to the positive or the negative sum of row j, according to the sign of A_{ji}; no term is used by both sums. Balanced binary sum trees compute these sums p_j and n_j in at most $\lceil \log_2(\Delta_r + 1) \rceil$ layers (an empty sum is zero). One neuron with two inputs then computes $\mathrm{ReLU}(p_j - n_j + b_j)$. Rows without entries give constant neurons. At most $C(w + E_r + 1)$

neurons are needed at any time: the terms and partial sums occupy at most E_r neurons at each level of the trees, and at most $2w$ zero or completed values are kept. Every neuron reads and is read at most twice.

For the output layer, carry its two nonnegative sums to the two neurons read by the output layer, and put the output bias there, without applying ReLU to their difference. Concatenating the converted layers gives the bounded fan-in network. The only coefficients are the magnitudes of the original weights, their negatives, ones, and the original biases. Padding to an admissible power-of-two width changes the width by a constant factor, and The intermediate bounded fan-in network has width $O(w+E+1)$ and depth $O(1+\sum_r \log(\Delta_r+1))$, and Theorem 3.1 gives the bounds of Corollary 3.2. □

G.2 Convolutional networks

A convolutional ReLU layer with c input and output channels, m spatial positions and a kernel with k spatial positions has width $w = cm$, at most $E_r \le c^2km$ nonzero weights, and fan-in and fan-out at most ck. Corollary 3.2 therefore represents such networks exactly, with width $O(c^2km)$ and depth overhead $O(\log^2(c^2km)\log(ck+1))$ per layer.

G.3 A smaller-width alternative for dense layers

Corollary G.1 (Representation of dense networks by batching)**.** *Under the same input and output conventions, let all hidden widths be at most w. For every integer $1 \le q \le w$, a dense ReLU network $\mathcal{N}$ of depth D is represented exactly on the fixed three-regular masks with*

$$n \le Cq(w+1),$$
$$L \le C(D+1)\lceil w/q\rceil \log_2^3(w+1),$$
$$P \le C(D+1)(w+1)^2 \log_2^3(w+1).$$

The maximum parameter magnitude becomes at most the maximum of one and the original magnitude.

Proof. Keep the w nonnegative activations of the previous layer and reserve w neurons for the new activations. Process at most q rows of the weight matrix at a time. Copy each old activation into $2q$ working neurons, keeping one copy in memory. For row j, compute the nonnegative terms $A^+_{ji}z_i$ and $A^-_{ji}z_i$ in separate neurons. Separate binary sum trees compute p_j and n_j, and $\mathrm{ReLU}(p_j - n_j + b_j)$ gives the new activation. Completed activations are kept while the remaining rows are processed, and the old activations are discarded only when all new ones are computed.

Copying, multiplying by the weights, summing and storing need $O(q(w+1))$ neurons and $O(\log_2(w+1))$ layers per batch, with fan-in and fan-out at most two. Each hidden layer needs $\lceil w/q\rceil$ batches. The output layer is one more pair of positive and negative sums, with the subtraction and the bias in the output layer itself. The bounded fan-in network therefore has depth $O((D+1)\lceil w/q\rceil \log_2(w+1))$. Theorem 3.1 multiplies this by $O(\log_2^2(w+1))$, since $q \le w$. Finally, $q\lceil w/q\rceil \le 2w$ gives the bound on P. No new coefficient of magnitude larger than one is introduced. □

For dense networks at fixed width, the total overhead is therefore larger than the routing cost alone: the dense sums and the fan-out must also be computed sequentially.

H Robustness to pruning

This appendix gives the replicated architecture, the spectral gap of the block mask, and the proofs of Theorem 6.2 and Proposition 6.3.

Definition H.1 (Architecture with r-fold redundancy)**.** Replace each neuron of the input layer by its block of r replicas. The unpruned input layer permits a connection from each input coordinate to every neuron in its designated block. The unpruned readout permits connections from every neuron

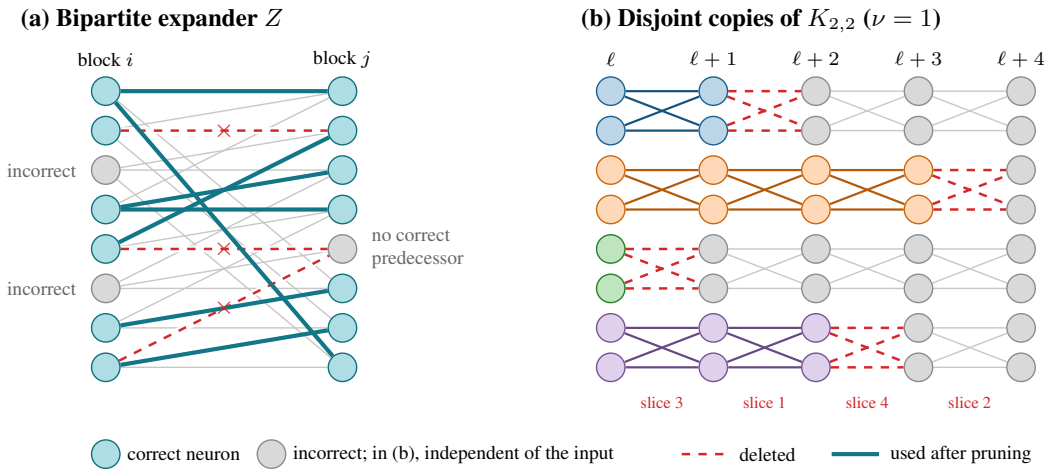


Figure 9: Expander graphs under pruning with bounded block-wise pruning ratio (Section 6). (a) One copy of a three-regular bipartite expander Z with $r = 8$ between two blocks. Two input neurons are incorrect and three connections are deleted. Every output neuron but one keeps a connection from a correct input and uses one such connection (thick); Lemma H.3 bounds the fraction that do not, at every layer. (b) A bipartite graph Z made of disjoint copies of $K_{2,2}$ splits the network into independent slices. Deleting all connections of one slice per layer has block-wise pruning ratio $\varphi = 1/4$; after four layers no neuron depends on the input, as in Proposition 6.3.

in each designated output block; after the pruning is known, only one weight per block need be nonzero. Pruning acts only on the hidden masks equation 6.1.

Lemma H.2 (Spectral gap of the Kronecker-product mask)**.** *Let Z be a d_Z-regular bipartite graph with $\nu = \sigma_2(\frac{1}{d_Z}Z)$. Then the mask equation 6.1 is binary and $3d_Z$-regular, and*

$$\sigma_2\left(\tfrac{1}{3d_Z}\widetilde{A}_n\right) = \max\{\sigma_2(\tfrac{1}{3}A_n), \nu\}. \tag{H.1}$$

Proof. The singular values of $(\frac{1}{3}A_n) \otimes (\frac{1}{d_Z}Z)$ are products of singular values of the two factors. Since both factors are doubly stochastic, the largest product is one and every other product is at most the larger of the two second singular values; both bounds are attained. □

To propagate correct values through deleted edges, we bound how many outputs can lose every connection from a mostly correct input set. Figure 9 illustrates this estimate and the contrasting failure without expansion.

Lemma H.3 (Expander mixing under edge deletion)**.** *In one copy of Z, let S be a set of inputs with relative size σ, delete at most $\varphi r d_Z$ connections, and let T be the set of outputs with no remaining connection from an input outside S, with relative size t. Then*

$$(1-\sigma)t \le \nu\sqrt{\sigma t} + \varphi. \tag{H.2}$$

Proof. Each of the $d_Z|T|$ connections into T either starts in S or has been deleted, so $d_Z|T| \le e(S,T) + \varphi r d_Z$. Write $e(S,T) = \mathbf{1}_T^\top Z \mathbf{1}_S$ and split $\mathbf{1}_S$ and $\mathbf{1}_T$ into their means and their components orthogonal to $\mathbf{1}$. Since $Z\mathbf{1} = Z^\top\mathbf{1} = d_Z\mathbf{1}$, the cross terms vanish, and since Z maps $\mathbf{1}^\perp$ into $\mathbf{1}^\perp$ with norm νd_Z,

$$e(S,T) \le \frac{d_Z|S||T|}{r} + \nu d_Z\sqrt{|S||T|}.$$

Dividing by $r d_Z$ proves equation H.2. □

Proof of Theorem 6.2. Put $\alpha = (1-\sqrt{2}\nu)/2 > 0$. Then $1 - \alpha - \sqrt{2}\nu = \alpha$, and equation 6.2 reads $\varphi \le \alpha^2/2$. Call a neuron *correct* if it holds the value of its block in the unpruned representation of

Theorem 3.1. We show by induction over layers that at most a fraction α of every block is incorrect. The input layer is not pruned, so all its neurons are correct.

In the unpruned representation every hidden neuron has at most two inputs with nonzero weight. Fix a block of layer ℓ and one such input block, in which a fraction $\sigma \le \alpha$ of neurons is incorrect. Let t be the fraction of neurons of the first block with no remaining connection from a correct neuron of the input block. Suppose $t > \alpha/2$. Then $\sigma \le \alpha < 2t$, so $\sqrt{\sigma t} < \sqrt{2}\, t$, and Lemma H.3 gives

$$(1-\alpha)t \le (1-\sigma)t \le \nu\sqrt{\sigma t} + \varphi \le \sqrt{2}\nu t + \varphi,$$

that is, $\alpha t \le \varphi \le \alpha^2/2$, so $t \le \alpha/2$, a contradiction. Hence each of the at most two inputs leaves at most a fraction $\alpha/2$ of the block without a correct predecessor, and outside a fraction α every neuron has, for each input, a remaining connection from a correct neuron.

Such a neuron puts the coefficient of the bounded fan-in network on one such connection per input, zero on all other connections in the mask, and the bias of the bounded fan-in network; it then computes the correct value. A neuron whose value is a constant or a stored zero needs no input and is correct. All remaining neurons receive zero weights and zero bias, and no later neuron reads them with nonzero weight. This closes the induction, and every block keeps at least $(1-\alpha)r = (1+\sqrt{2}\nu)r/2$ correct neurons. Since $\alpha < 1$, the output layer finds a correct neuron in each of the two output blocks. By Definition H.1, its fixed readout permits all r replicas in each block, so it assigns the two output-layer weights of the bounded fan-in network to one correct neuron per block and zero to the other permitted readout connections. All weights are parameters of the bounded fan-in network, 0 or 1. □

Proof of Proposition 6.3. Every connection of $A_n \otimes Z$ joins neurons whose indices in $[r]$ lie in the same component of Z, so the network splits into r/d_Z independent slices. Put $c = \lfloor \varphi r/d_Z \rfloor \ge 1$. In each hidden layer, delete all connections of c slices not deleted before. Each copy of Z loses cd_Z^2 of its rd_Z connections, a fraction $cd_Z/r \le \varphi$. Since $c \ge \varphi r/(2d_Z)$, every slice has lost all connections of some hidden layer within the first $\lceil 2/\varphi \rceil$ hidden masks. After that layer the slice does not depend on the input, and since no connection joins different slices, neither does the output. Thus the output is independent of the input. A function independent of the input differs from $x \mapsto x_1$ by at least $1/2$ at $x_1 = 0$ or at $x_1 = 1$. □

I Lower bounds from bounded receptive fields and output conventions

The following bounds concern how many input coordinates a network can combine at all. They are unrelated to the metric-entropy lower bound of Theorem D.2.

Proposition I.1 (Receptive fields and a common target)**.** *Suppose every hidden neuron has fan-in at most m, there are L hidden layers, and the output layer has fan-in at most r. Put $s = \min\{d, rm^L\}$. For $f_\Sigma(x) = \sum_{i=1}^d x_i$, every such network satisfies*

$$\|F - f_\Sigma\|_\infty \ge (d-s)/2, \qquad \|F - f_\Sigma\|_{L^2([0,1]^d)} \ge \sqrt{(d-s)/12}.$$

This holds even if the masks are chosen separately for each network.

Proof. By induction over layers, a neuron in layer ℓ depends on at most m^ℓ input coordinates, so the output depends on a set S of at most s coordinates. Fix x_S and change all other coordinates from zero to one. The target changes by $d - |S|$, while the output of the network does not change, so one of the two errors is at least $(d-|S|)/2$.

For the L^2 bound, under the uniform distribution the best mean-squared prediction of f_Σ from x_S is its conditional mean. The sum of the missing independent uniform coordinates has variance $(d-|S|)/12$, which bounds the squared error of any function of x_S from below. The same target works for every set S. □

Since every hidden neuron of our masks has fan-in three and the output reads two neurons, Proposition I.1 shows that approximating f_Σ to error below $1/2$ requires $2 \cdot 3^L \ge d$, that is, $L \ge \log_3(d/2)$: depth growing with the input dimension is unavoidable on any constant-degree mask.

If the output layer also has fan-in m, the condition for the output to see every coordinate is $m^{L+1} \geq d$, not $m^L \geq d$, with our convention for depth. A dense output layer, however, can see every coordinate through many different last-layer neurons. The following ANOVA argument covers that case.

For $f \in L^2([0,1]^d)$, let E_i average over coordinate i, and define

$$f_S = \prod_{i \in S} (I - E_i) \prod_{i \notin S} E_i f, \qquad S \subseteq [d].$$

These commuting orthogonal projections give mutually orthogonal components whose sum is f (the ANOVA decomposition).

Proposition I.2 (Lower bound via the ANOVA decomposition with a dense output layer)**.** *If each neuron of the last hidden layer depends on at most s coordinates and the output is any affine combination of these neurons, then*

$$\|f - F\|_2^2 \geq \sum_{|S| > s} \|f_S\|_2^2 .$$

If $s < d$, the target $f_\Pi(x) = \prod_{i=1}^d (2x_i - 1)$ also satisfies $\|f_\Pi - F\|_\infty \geq 1$, for any hidden width.

Proof. A function of the coordinates in T has zero component f_S for every $S \not\subseteq T$, since $I - E_i$ with $i \in S \setminus T$ annihilates it. Hence $F_S = 0$ whenever $|S| > s$, and orthogonality gives the L^2 bound.

For the uniform bound, consider the functional

$$\Lambda(g) = 2^{-d} \sum_{v \in \{0,1\}^d} \left(\prod_{i=1}^d (2v_i - 1) \right) g(v),$$

which has norm one with respect to the sup norm. It vanishes on every function that does not depend on some coordinate, as the two values of that coordinate cancel. Hence $\Lambda(F) = 0$ while $\Lambda(f_\Pi) = 1$, which gives the claim. For hidden fan-in m we may take $s = m^L$. □