STRUCTURAL CORRESPONDENCE AND UNIVERSAL APPROXIMATION IN DIAGONAL PLUS LOW-RANK NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

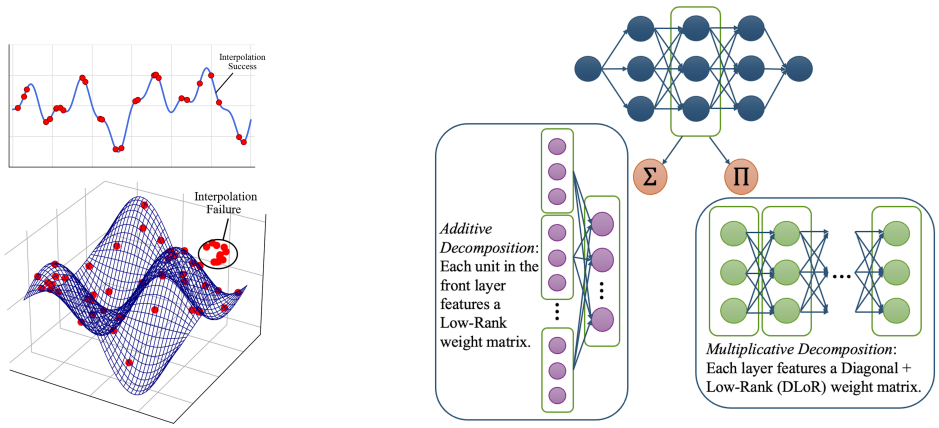
Structured low-rank and sparse-plus-low-rank parameterizations, motivated by methods such as LoRA and low-rank factorization, impose strong structural constraints on the linear transformations within neural networks. We study the expressive consequences of these constrained structures from the perspective of universal approximation. We first characterize that purely low-rank neural networks, despite being able to exactly interpolate arbitrary finite scalar datasets, can fail to approximate continuous functions that vary along directions ignored by the low-rank hidden mapping. We show that augmenting the low-rank structure with a one-parameter diagonal component suffices to overcome this geometric bottleneck, yielding a Diagonal-plus-Low-Rank (DLoR) structure. We show that a full-rank linear transformations can be reconstructed using DLoR components either by expanding network width through additive decomposition or by extending network depth through multiplicative decomposition. We term this correspondence between additive width expansion and multiplicative depth extension the *Structural Correspondence* framework. By combining these constructions with local linearization of the activation function, we prove that DLoR neural networks restore universal approximation for general activation functions. These results characterize how architectural width and depth can compensate for severe rank constraints without requiring unrestricted dense hidden transformations.

1 INTRODUCTION

The era of modern foundation models has been defined by the aggressive scaling of deep neural network architectures to billions, or even trillions of parameters (LeCun et al., 2015; Achiam et al., 2023). While highly over-parameterized neural networks are appealing due to their favorable optimization landscapes and global convergence properties (Allen-Zhu et al., 2019; Baldi, 2024), their immense scale imposes prohibitive memory and computational bottlenecks (Rajbhandari et al., 2020; Strubell et al., 2019). This reality has driven a critical need for parameter and memory-efficient training frameworks (Lialin et al., 2023), as well as robust network compression techniques to alleviate the burden of deployment.

Techniques such as Low-Rank Adaptation (LoRA) (Hu et al., 2022; Lialin et al., 2024), sparse plus low-rank matrix factorization (Chandrasekaran et al., 2009; Ding et al., 2023), and parameter-efficient adapter modules (Houlsby et al., 2019) have become the preferred standard for efficiency. These low-rank structures are highly favored because they drastically reduce computational overhead while leveraging the phenomenon that pre-trained neural networks typically operate within a low intrinsic dimensionality (Aghajanyan et al., 2021). By operating on constrained low-rank manifolds, these methods can yield empirical fine-tuning performance that rivals full-rank training (Zhang et al., 2026). However, purely low-rank structures restrict parameters to a severely constrained low-dimensional subspace. Specifically, low-rank components are restricted to capturing only the dominant singular value spectrum, entirely ignoring variations outside this subspace. This prompts a question: what exact topological or geometric bottlenecks constrain the expressive power of severely rank-deficient neural networks? Recent literature has attempted to justify the expressive power of parameter-efficient models (Zeng & Lee, 2024; Giannou et al., 2023; Li et al., 2025). However, the theoretical guarantees provided by these works remain limited. Current claims rely on the presence of a pretrained dense base matrix, the specific use of the ReLU or Swish activation

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.



(a) Interpolation by rank-1 NN (b) Additive and multiplicative decomposition

Figure 1: Overview of our work. (a) Rank-1 neural networks (NN) can perfectly interpolate any set of scalar outputs, but inherently fail for multi-dimensional function approximations. (b) We use rank-deficient structures by either expanding width (with additive decomposition) or extending depth (with multiplicative decomposition). While the former can rely on purely low-rank weight matrices, the latter necessitates DLoR weight matrices.

function, or non-verifiable singularity conditions regarding the weight matrices. In contrast, we ask a more fundamental question: can a strictly constrained neural network achieve the Universal Approximation Theorem (UAT) (Hornik et al., 1989; Kidger & Lyons, 2020), without these additional priors and assumptions?

We first illustrate a distinction between finite-sample interpolation and continuous function approximation using general activation functions: while a single purely rank-1 hidden layer can exactly interpolate scalar datasets, this finite-sample interpolation capacity does not extend to uniform approximation of general functions on multidimensional input domains. Strict rank-1 mappings suffer from a geometric phenomenon that we term *orthogonal blindness*. Since the network evaluates inputs strictly through a 1-dimensional projection, it becomes completely invariant to variations in the data orthogonal to this projection vector. This raises a structural question: under severe rank constraints, what architectural mechanisms can compensate for the resulting loss of expressivity?

Crucially, we establish a **Structural Correspondence** framework showing that rank deficiency can be compensated for through two distinct architectural mechanisms. We prove that restructuring a weight matrix via an *additive decomposition* naturally translates to expanding the network's width, mapping a high-rank linear operation onto a parallel summation of nonlinear layers. Conversely, a *multiplicative decomposition* corresponds to extending the network's depth. These two mechanisms have different structural requirements: sums of low-rank matrices can recover high-rank matrices, whereas products of rank-deficient matrices remain rank deficient.

This structural asymmetry motivates augmenting the low-rank factors in the multiplicative construction with a minimal full-rank pathway. We consider a matrix structure of the form $M = \alpha I + UV^T$, a special instance of a sparse plus low-rank matrix that we term a **Diagonal plus Low-Rank (DLoR)** component. It introduces a sparse diagonal component to the low-rank structure $\mathrm{rank}(UV^T) \leq r$, where r is any arbitrary rank constraint. Unlike a dense prior, this αI component requires minimally one additional parameter to store, yet it provides a full-rank pathway to capture the orthogonal spectrum, allowing information to bypass the low-dimensional subspace and breaking the geometric bottleneck of orthogonal blindness. For the multiplicative construction, we further show that the resulting $O(N/r)$ depth scaling is tight up to constant terms in the worst case. This structure is also consistent with recent empirical frameworks such as SLTrain (Han et al., 2024), which combine low-rank factors with sparse matrices. A summary of our theoretical framework, contrasting the interpolation failure of purely rank-1 neural networks with our DLoR architectures, is provided in Figure 1. Specifically, our main contributions are as follows:

- **The Structural Correspondence Framework:** We introduce both additive and multiplicative decomposition methods using DLoR components. We prove that rank deficiency can be compensated for by either expanding network width (additive) or extending network depth (multiplicative). For the multiplicative DLoR construction, we further establish a matching worst-case depth lower bound.

- **Universal Approximation via DLoR Structures**: We prove that introducing only a minimal $\mathcal{O}(1)$ sparse diagonal to form a DLoR structure perfectly overcomes orthogonal blindness. We show that Deep DLoR Neural Networks with bounded width restore uniform approximation capabilities for continuous functions with any nonaffine continuous activation functions.

2 Related Work

Universal Approximation and the Depth-Width Trade-off. The Universal Approximation Theorem (UAT) establishes that neural networks are dense in the space of continuous functions on a compact domain, $C(\mathcal{K}, \mathbb{R}^m)$, with respect to the uniform norm. Classic results proved that a single hidden layer utilizing any non-polynomial activation function can achieve uniform ϵ-approximation (Hornik et al., 1989; Cybenko, 1989; Pinkus, 1999). However, these shallow architectures suffer from the curse of dimensionality: to approximate functions in a bounded domain $\mathcal{K} \subset \mathbb{R}^n$, shallow architectures require a prohibitive width scaling of $\mathcal{O}(\epsilon^{-n})$ (Yarotsky, 2017). Subsequent theoretical breakthroughs established that depth exponentially increases expressivity and provides fundamentally superior parameter efficiency compared to wide, shallow neural networks (Montúfar et al., 2014; Cohen et al., 2016; Rolnick & Tegmark, 2018). (Telgarsky, 2016) and (Eldan & Shamir, 2016) proved that certain highly oscillatory functions can be expressed with a deep network of $\mathcal{O}(L)$ depth and $\mathcal{O}(1)$ width, but necessitate an intractable $\mathcal{O}(2^L)$ width if restricted to shallow architectures. When approximating the class of highly smooth functions within the Sobolev space $\mathcal{W}^{s,\infty}([0,1]^n)$ (functions with s bounded derivatives), deep ReLU neural networks strictly bound the required parameter complexity to $\mathcal{O}(\epsilon^{-n/s}\log(1/\epsilon))$ with a depth scaling of exactly $\mathcal{O}(\log(1/\epsilon))$ (Yarotsky, 2017; Petersen & Voigtlaender, 2018). In the regime of strictly bounded width, research demonstrated that deep ReLU neural networks of minimal width $n+1$ are sufficient to approximate any scalar-valued function in $C(\mathcal{K}, \mathbb{R})$ (Lu et al., 2017; Hanin & Sellke, 2017). (Kidger & Lyons, 2020) generalized this mapping framework, proving that for *every* non-affine continuous activation function, a strict maximum width of $n+m+2$ is sufficient to approximate any continuous target $f : \mathbb{R}^n \to \mathbb{R}^m$ under a large depth. However, all these foundational proofs inherently allow the network's internal linear transformations to be unrestricted, rather than imposing explicit low-rank structural constraints. Our work bridges the gap between classic UAT and modern parameter-constrained architectures by questioning whether these universal approximation guarantees on $C(\mathcal{K}, \mathbb{R}^m)$ hold when the hypothesis space is restricted to a low-rank and sparse manifold.

Expressivity of Parameter-Efficient Architectures. Beyond full-rank dense neural networks, a parallel line of approximation theory investigates structurally constrained architectures. (Bolcskei et al., 2019) established a foundational framework demonstrating that sparsely connected deep neural networks can achieve information-theoretical optimal approximation rates, matching the fundamental limits of nonlinear approximation for a broad class of functions. Subsequent works (Elbrächter et al., 2021; Daubechies et al., 2022) further proved that strict sparsity enables deep neural networks to overcome the curse of dimensionality without parameter explosion. In the empirical regime of modern foundation models, architectural efficiency is typically enforced via Parameter-Efficient Fine-Tuning (PEFT) methods, such as Low-Rank Adaptation (LoRA) (Hu et al., 2022). However, much of the approximation theory for parameter-constrained networks has focused on sparsely connected architectures (Bolcskei et al., 2019), whereas modern parameter-efficient methods often impose structured low-rank constraints on otherwise dense linear transformations. This leaves a gap in understanding the expressive consequences of structured rank constraints. Recent attempts to formalize PEFT expressivity assume the presence of a frozen, full-rank base matrix, specific non-singularity constraints, and the ReLU activation function (Zeng & Lee, 2024; Giannou et al., 2023). While (Li et al., 2025) establishes universal approximation bounds for 50% low-rank architectures, their guarantees rely on the Swish activation function and mandate relatively wide bottlenecks. In contrast, we study the function class induced directly by severely rank-constrained weight matrices without assuming a pretrained dense base and only mild assumption on activation functions. Our Structural Correspondence framework shows how dense transformations can be recovered through either additive width or multiplicative depth extension when augmented with a minimal sparse diagonal.

Sparse Plus Low-Rank Decomposition. The decomposition of complex matrices into low-rank and sparse components has deep mathematical roots in robust statistics and convex optimization, where the rank-sparsity incoherence principle proves that these two structures complement each

other (Chandrasekaran et al., 2009; Yuan & Yang, 2013; Zhang et al., 2018; Bertsimas et al., 2023). Recently, this principle has been adapted empirically for deep learning to compress large language models (Li et al., 2023; Ding et al., 2023), optimize attention mechanisms (Chen et al., 2021), and facilitate full-rank-equivalent pretraining by capturing the lower-magnitude spectrum of the data (Han et al., 2024). Simultaneously, empirical studies on PEFT have discovered that adding identity residual mappings to LoRA drastically improves optimization stability and fine-tuning performance (Shi et al., 2024; Xia et al., 2024). Our DLoR parameterization is a highly constrained instance of this sparse-plus-low-rank family, in which the sparse component is reduced to a single scalar multiple of the identity. Rather than analyzing its optimization behavior, we study the resulting function class and show that this one-parameter diagonal augmentation is sufficient for bounded-width multiplicative constructions to recover dense transformations and universal approximation.

3 Theoretical Limits of Pure Low-Rank Architectures

Before analyzing our proposed sparse-augmented architecture DLoR, we characterize the expressive boundaries of strictly low-rank neural networks. We distinguish finite-sample interpolation from two geometric limitations induced by low-rank constraints: output-space affine collapse for multidimensional targets, and input-space orthogonal blindness on multidimensional domains. Proofs can be found in Appendix B.

We first introduce the setup for the finite-sample interpolation. Given M distinct inputs $\{x_i\}_{i=1}^M \subset \mathbb{R}^d$ and targets $\{z_i\}_{i=1}^M \subset \mathbb{R}^k$, consider a low-rank parametrization of a standard feedforward neural network with one hidden layer as: $W_1 = U_1 V_1^T$, $W_2 = U_2 V_2^T$, where $U_1 \in \mathbb{R}^{N\times r_1}$, $V_1 \in \mathbb{R}^{d\times r_1}$, $U_2 \in \mathbb{R}^{k\times r_2}$, and $V_2 \in \mathbb{R}^{N\times r_2}$. The network maps each individual input as:

$$f(x) = U_2 V_2^T \sigma(U_1 V_1^T x + b_1) + b_2. \tag{1}$$

We first show that rank-one matrices already suffice for finite scalar interpolation, and then characterize the geometric limitations that persist under general low-rank neural networks.

3.1 Finite-Sample Interpolation Under Rank Constraints

We begin with the most restrictive nontrivial setting, where both weight matrices are rank one. When the targets z_i are scalars ($k = 1$), the rank-one outer mapping $U_2 V_2^T$ reduces to a row vector. We show that even this highly constrained architecture can exactly interpolate an arbitrary finite dataset.

Lemma 1 (Scalar Interpolation). *Given distinct inputs $\{x_i\}_{i=1}^M \subset \mathbb{R}^d$ and arbitrary scalar targets $\{z_i\}_{i=1}^M \subset \mathbb{R}$, let σ be any continuous, non-mean-periodic function (common activations like ReLU, Sigmoid, and SiLU are not mean-periodic). For network width $N = M$, there exist rank-one factors $U_1 \in \mathbb{R}^{N\times 1}$, $V_1 \in \mathbb{R}^{d\times 1}$, $U_2 \in \mathbb{R}^{1\times 1}$, $V_2 \in \mathbb{R}^{N\times 1}$, and $b_1 \in \mathbb{R}^N$ such that $U_2 V_2^T \sigma(U_1 V_1^T x_i + b_1) = z_i,\ i = 1, ..., M$.*

Remark 1. *Our exact scalar interpolation result for general activation functions matches (Constantinescu & Popescu, 2023). Instead of using full-rank hidden weight matrix W_1, we demonstrate that a strictly rank-1 weight matrix is sufficient to perfectly interpolate arbitrary scalar targets.*

However, when the output layer is also rank-constrained, this interpolation capability does not extend to mappings in $\mathbb{R}^k$ with $k \geq 2$. The low-rank output layer can restrict the output to an affine subspace of a smaller dimension than the geometry of the multidimensional targets as formalized below.

Lemma 2 (Affine Collapse). *If $\mathrm{rank}(W_2) \leq r_2$, then $f(x) \in b_2 + \mathrm{Im}(W_2)$ for every x, which is an affine subspace of dimension at most r_2. Consequently, any target set whose affine span has dimension greater than r_2 cannot be exactly interpolated.*

This obstruction is caused by the output rank constraint. If W_2 is unrestricted, a rank-one hidden layer can again exactly interpolate arbitrary finite multidimensional targets (proof in Appendix B.4).

3.2 Geometric Limitations of Low-Rank Hidden Mappings

Exact finite interpolation is a strictly weaker condition than universal approximation. Even with a full-rank outer layer, a low-rank hidden transformation remains invariant along a nontrivial input subspace, preventing approximation of general target functions that vary along these directions. We term this geometric limitation *Orthogonal Blindness*.

Definition 1 (Orthogonal Blindness). *Let $\mathcal{N}(V_1^T) = \{x \in \mathbb{R}^d \mid V_1^T x = 0\}$ denote the at least $(d - r_1)$-dimensional null space orthogonal to $\mathrm{Col}(V_1)$. For any input x, we apply the orthogonal decomposition $x = x_\parallel + x_\perp$, where $x_\perp \in \mathcal{N}(V_1^T)$. A purely Rank-$r$ mapping $f(x) = g(V_1^T x)$ evaluates to: $f(x_\parallel + x_\perp) = g\big(V_1^T(x_\parallel + x_\perp)\big) = g(V_1^T x_\parallel) = f(x_\parallel)$.*

Consequently, if the target function varies along directions in the orthogonal space of $\mathrm{Col}(V_1)$, the strictly constrained network fails to achieve function approximation. Figure 4 illustrates the rank-one case. These limitations motivate the architectural mechanisms introduced in Section 4.

4 The Structural Correspondence Framework

To overcome the affine collapse and orthogonal blindness intrinsic to purely low-rank mappings formalized in Section 3, we propose augmenting the network with a minimal sparse diagonal component. Proofs can be found in Appendix C.

Definition 2 (Diagonal plus Low-Rank (DLoR) Component). *A linear transformation $M \in \mathbb{R}^{N\times N}$ is a DLoR component of rank r if it is constrained to the form $M = \alpha I_N + UV^T$ where $\alpha \in \mathbb{R}$ is a scalar forming a sparse diagonal, and $U, V \in \mathbb{R}^{N\times r}$ form a low-rank matrix ($r \ll N$).*

This αI_N component requires only $\mathcal{O}(1)$ parameters to store, yet it serves as a vital residual pathway. When $\alpha \neq 0$, the DLoR component can retain full rank, whereas purely rank-1 architectures necessarily have a nontrivial orthogonal null space. A similar effect has been observed for ResNet (Lin & Jegelka, 2018). In this section, we establish a structural correspondence demonstrating how any dense, full-rank matrix $W \in \mathbb{R}^{N\times N}$ can be perfectly reconstructed using solely these Diagonal plus Low-Rank components by rigorously trading off either network width (via additive decomposition) or network depth (via multiplicative decomposition).

4.1 Additive Decomposition: Width Expansion

A fundamental approach to reconstructing a high-rank matrix is through linear addition. Consider an arbitrary (high-rank) weight matrix W and decompose it into L independent low-rank components via singular value decomposition (SVD) for some value L, i.e., $W = \sum_{l=1}^{L} M_l$, where each matrix M_l is a DLoR component with $\alpha = 0$ (although any $\alpha \in \mathbb{R}$ is also acceptable). Inside a neural network, these components cannot simply be summed before activation; they must be mapped through the non-linear layer. We demonstrate that projecting this linear addition through the activation function structurally translates to expanding the network's width into a parallel ensemble.

Assumption 1. *$\rho : \mathbb{R} \to \mathbb{R}$ is continuous and non-affine. There exists an open interval containing c on which ρ is continuously differentiable, with $\rho'(c) \neq 0$.*

Theorem 1 (Additive Width Expansion). *Let $W = \sum_{l=1}^{L} M_l \in \mathbb{R}^{N\times N}$, with $L \geq 2$. Under Assumption 1, choose nonzero scalars β_l such that $\sum_{l=1}^{L} \beta_l = 0$. Then, for every compact input domain $\mathcal{K} \subset \mathbb{R}^N$, the parallel wide structure computes the exact linear mapping Wx as $h \to 0$:* $\lim_{h\to 0} \sup_{x\in\mathcal{K}} \left\| \sum_{l=1}^{L} \frac{\beta_l}{h\rho'(c)} \rho\Big(\frac{h}{\beta_l} M_l x + c\mathbf{1}_N\Big) - Wx \right\|_2 = 0.$

A decomposition with one component can be extended by a zero component.

Width Architecture and Elimination of Affine Drift: The limit expression in Theorem 1 precisely describes a width expansion. The parallel summation can be compactly written as $B\rho(W_{wide}x + b_{wide})$, where the outer linear projection is $B = [\frac{\beta_1}{h\rho'(c)} I_N, \ldots, \frac{\beta_L}{h\rho'(c)} I_N]$, and the inner parameters W_{wide} and b_{wide} are formed by vertically stacking the blocks $\frac{hM_l}{\beta_l}$ and $c\mathbf{1}_N$, respectively. This increases the layer width by a factor of L. The zero-sum condition $\sum_{l=1}^{L} \beta_l = 0$ cancels the constant zeroth-order terms introduced by the local Taylor expansions. Consequently, when the construction is used to approximate a dense layer $y = \rho(Wx + b)$, the original target bias b can be retained without an additional correction induced by these constant offsets. Here $B \in \mathbb{R}^{N\times LN}$ is a structured summation operator with scalar identity blocks. The wide architecture consists of parallel DLoR branches and B is not itself a square DLoR component.

4.2 Multiplicative Decomposition: Depth Extension

While the additive decomposition effectively recovers expressivity, expanding network width is often computationally undesirable for memory-constrained inference. Alternatively, to bound the network width to its original dimension N, we establish that a high-rank weight matrix can be factorized multiplicatively. We prove that any full-rank transformation can be decomposed into sequential

DLoR layers. To achieve universal expressivity using highly constrained DLoR components, our theoretical framework relies on the property that continuous, non-linear activation functions can smoothly approximate linear mappings. We adapt a lemma from (Kidger & Lyons, 2020) to establish this identity mapping.

Lemma 3 ((Kidger & Lyons, 2020), Lemma 4.1)**.** *Under Assumption 1, define the scalar affine functions:* $\Phi_h(x) = hx + c, \Psi_h(y) = \frac{y-\rho(c)}{h\rho'(c)}$. *Then, as the scaling parameter* $h \to 0$*, the composition* $(\Psi_h \circ \rho \circ \Phi_h)(x)$ *converges uniformly to the exact identity function* x *on any compact domain.*

Theorem 2 (Multiplicative Depth Extension)**.** *For any invertible matrix* $W \in \mathbb{R}^{N\times N}$ *and any rank constraint* $1 \le r \le N$*, let* $L = \lceil N/r \rceil$*. Let* ρ*,* Φ_h *and* Ψ_h *be as in Lemma 3. There exists a non-zero scalar* $\alpha \in \mathbb{R}$ *and a sequence of low-rank matrices* $U_l, V_l \in \mathbb{R}^{N\times r_l}$ *(where* $r_l \le r$*) defining DLoR components* $M_l = \alpha I_N + U_l V_l^T$ *for* $l = 1, \dots, L$*, such that* $W = M_L \cdots M_1$*. Moreover, for any compact input domain* $\mathcal{K} \subset \mathbb{R}^N$ *and target bias* $b \in \mathbb{R}^N$*, the deep sequential structure approximates the exact dense layer and the following convergence holds uniformly for* $x \in \mathcal{K}$*:*

$$\lim_{h\to 0} \rho\left(M_L \circ \Psi_h \circ \rho \circ \Phi_h \circ M_{L-1} \circ \cdots \circ \Psi_h \circ \rho \circ \Phi_h \circ M_1(x) + b\right) = \rho(Wx + b).$$

Remark 2 (Role of the $h \to 0$ limit)**.** *The limit* $h \to 0$ *is a standard analytical device for the approximation result rather than an assumption on practical training. Theorems 1 and 2 construct a family of DLoR networks indexed by* h *whose realized functions converge uniformly on compact domains to the target dense layer. Thus, the results establish representational existence and density, but do not imply that gradient-based optimization will recover these constructions for arbitrarily small* h*. Finite-*h *behavior of the constructed networks is illustrated empirically in Section 6.1. Under an additional local* C^2 *condition, Appendix C.5 provides explicit finite-*h *error bounds for both constructions.*

Remark 3. *The* $O(N/r)$ *depth scaling in Theorem 2 is tight in the worst case: there exist linear maps for which any exact representation using the multiplicative depth extension with DLoR components requires* $L \ge \left\lceil \frac{N-1}{r} \right\rceil$*. See Appendix F.5.*

This theorem formally proves that depth can substitute for parameter rank. A single dense, high-rank linear transformation can be exactly decomposed into a deep sequence of L non-linear layers, strictly preserving a constant width N at every step. Moreover, the sequential composition of the DLoR components M_l with the affine mappings Φ_h and Ψ_h strictly preserves the DLoR architectural constraints. Because the mappings Φ_h and Ψ_h consist entirely of uniform scalar multiplications and constant vector additions, applying them to the linear operator simply scales the weight matrix and shifts the layer bias. We also provide an alternative construction utilizing an augmented $2N$-dimensional state space in Appendix E for clear intuition.

Proposition 1 (Exact ReLU decomposition)**.** *Let* $\mathcal{K} \subset \mathbb{R}^N$ *be compact,* $W \in \mathbb{R}^{N\times N}$*,* $b \in \mathbb{R}^N$*, and* $1 \le r \le N$*. The map* $x \mapsto \mathrm{ReLU}(Wx + b)$ *has an exact representation on* $\mathcal{K}$ *by a ReLU DLoR block of state width* N*, update rank at most* r*, and at most* $\lceil 3N/r \rceil$ *affine stages. If* W *is invertible,* $\lceil N/r \rceil$ *stages suffice.*

ReLU is exactly affine on its positive half-line. By adding finite bias shifts, every intermediate state can be kept in this region on the prescribed compact set, while the following bias correction removes the accumulated shift. Thus this construction preserves the state width and requires no limiting operation. The proof is given in Appendix C.8.

Remark 4 (Rectangular Transformations)**.** *The square formulation extends to rectangular matrices through orthogonal completion. Let* $W \in \mathbb{R}^{m\times n}$ *have maximal rank and* $W = U\Sigma V^T$ *be its full singular value decomposition (SVD). If* $m < n$*, appending the remaining right singular vectors as rows produces an invertible* $n \times n$ *matrix whose first* m *output coordinates equal* Wx*. If* $m > n$*, similarly, we append the remaining left singular vectors as columns. Hence, rectangular layers can be handled by the square construction in Theorem 2. Rank-deficient matrices can be treated by an arbitrarily small maximal-rank perturbation.*

Remark 5 (Hybrid Structural Combinations)**.** *While Theorems 1 and 2 establish the pure limits of width and depth respectively, the Structural Correspondence framework naturally permits hybrid topologies. Because matrix addition and matrix multiplication are mathematically composable, any general high-rank weight matrix* W *can be decomposed into a complex combination of both summations and products of Sparse Low-Rank components. For instance,* W *could be approximated by a parallel ensemble of several moderately deep sub-networks, taking the form* $W = \sum_i (\prod_j M_{i,j})$*.*

This allows machine learning practitioners to dynamically navigate the depth-width trade-off space, tailoring the architectural reconstruction to perfectly match the specific hardware constraints (e.g., parallel GPU thread limits versus sequential VRAM constraints) of their deployment environment.

The limit $h \to 0$ in Theorems 1 and 2 is an analytical device for general activations: it places each activation in a locally linear regime and yields explicit existence constructions. ReLU admits a stronger finite-scale statement. Proposition 1 therefore gives an exact same-width ReLU realization with no limiting operation and zero simulation error. For general activations, an additional local C^2 assumption yields explicit finite-h error bounds as in Proposition 3. These results concern representational existence and understanding the optimization behavior of directly trained DLoR networks is left for future work.

5 Universal Approximation for DLoR

Having established the structural correspondence of DLoR components, we now prove that embedding these decompositions into a deep or wide network architecture completely overcomes orthogonal blindness and fully restores the Universal Approximation Theorem (UAT). We use ρ in hidden layers and an affine final readout. In the deep construction, trainable square affine matrices are DLoR, with fixed input padding and output coordinate selection as in Remark 4. In the wide construction, parallel DLoR branches are combined by the structured summation operator described in Section 4.1.

It is critical to note the broad applicability of the activation condition in Theorems 1 and 2. It only requires that there exist *some* expansion point $c \in \mathbb{R}$ where $\rho'(c) \neq 0$. This inherently covers smooth non-linearities like Sigmoid, Tanh, Swish, and GELU. ReLU also satisfies the assumption by choosing $c > 0$, where ρ is locally affine with derivative 1. In addition, ReLU admits the finite exact construction in Proposition 1.

Since our DLoR blocks can uniformly simulate any dense affine transformation $y = \rho(Wx+b)$ over a compact domain, and continuous functions map compact sets to compact sets, the intermediate representations at any hidden layer remain bounded, allowing us to enforce uniform bounds on the asymptotic Taylor remainders of our approximations. Proofs can be found in Appendix D.

5.1 UAT via Multiplicative Depth and Additive Width

We first prove that universal approximation for DLoR can be achieved by extending depth and preserving the width bound $N = n + m + 2$. Conversely, if network depth is restricted, universal approximation can be achieved by expanding the network width via our additive structural correspondence, mapping the high-rank transformation into a parallel ensemble. Both deep and wide DLoR architectures theoretically achieve universal approximation, aligning with classical approximation theory on the exponential benefits of depth and linear benefits of depth in expressivity.

Theorem 3 (Deep DLoR UAT)**.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be compact. Under Assumption 1, deep DLoR networks of arbitrary depth, state width $N = n + m + 2$, and update rank at most r, for any fixed $1 \leq r \leq N$, are dense in $C(\mathcal{K}, \mathbb{R}^m)$.*

Theorem 4 (Wide DLoR UAT)**.** *Let $\mathcal{K} \subset \mathbb{R}^n$ be compact. Under Assumption 1, finite networks formed by replacing dense layers with parallel DLoR branches and the structured aggregation of Theorem 1 are dense in $C(\mathcal{K}, \mathbb{R}^m)$.*

Beyond the standard Universal Approximation Theorem for continuous functions, a profound consequence of our Structural Correspondence framework is that *any* existing universal approximation guarantee for a dense network can be systematically transferred to our constrained DLoR network.

Theorem 5 (General UAT Transfer)**.** *Under Assumption 1, let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a finite dense network with L affine stages, hidden activation ρ, and an affine final readout. Let W be at least the maximum of its input, output, and hidden dimensions. For every compact $\mathcal{K} \subset \mathbb{R}^n$, every $\eta > 0$, and every $1 \leq r \leq W$, there exists a DLoR network G of state width W and at most $L\lceil W/r \rceil$ affine stages such that $\|G - F\|_{\infty,\mathcal{K}} < \eta$. Its update ranks are at most r, and its boundary interfaces are fixed padding and coordinate selection. Consequently, if $\|F - f\|_{\infty,\mathcal{K}} \leq \epsilon$, then $\|G - f\|_{\infty,\mathcal{K}} < \epsilon + \eta$.*

5.2 Parameter Complexity for Sobolev Spaces

We count all allocated real scalar entries in U, V, the diagonal coefficient, and the bias, including entries set to zero by a construction. A width-N affine DLoR layer with factor rank r_j has $2Nr_j +$

$N+1$ such slots. For $q = \lceil N/r \rceil$ and $\sum_j r_j = N$, one simulated square layer therefore uses at most $\sum_{j=1}^{q}(2Nr_j + N + 1) = 2N^2 + q(N+1) = O(N^2)$.

For fixed width, this is a constant-factor overhead relative to a fully allocated dense matrix. We combine this accounting with the fixed-width approximation theorem of Siegel (2023, Theorem 1). It gives uniform error bounded by $C_{d,s}L^{-2s/d}$ on the unit Sobolev ball, with ReLU hidden width $25d + 31$. The width is independent of the requested accuracy, so DLoR conversion has constant overhead for fixed d. The following theorem uses the standard unrestricted-real-parameter model for ReLU networks. Proof and parameter accounting are in Appendix D.4.

Theorem 6 (DLoR Sobolev parameter bound)**.** *Fix integers $d, s \geq 1$ and let $\mathcal{B}_{d,s} = \{f \in W^{s,\infty}([0,1]^d) : \|f\|_{W^{s,\infty}} \leq 1\}$, using the continuous representatives of these functions. There exists $C_{d,s}$ such that, for every $f \in \mathcal{B}_{d,s}$ and $0 < \epsilon < 1/2$, a ReLU DLoR network G of width $N = 25d + 31$, with every trainable square affine matrix of the form $\alpha I_N + uv^T$, satisfies*

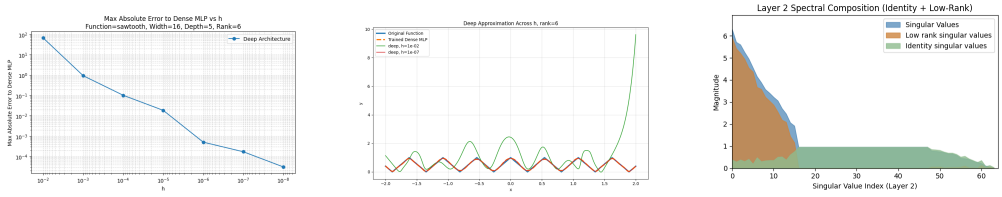
$$\|G - f\|_{\infty,[0,1]^d} < \epsilon, \qquad P(G), D(G) \leq C_{d,s}\epsilon^{-d/(2s)}.$$

counts affine stages, $P(G)$ counts all scalar slots including biases, and input padding and output coordinate projection are fixed as in Remark 4.

6 Numerical Experiments

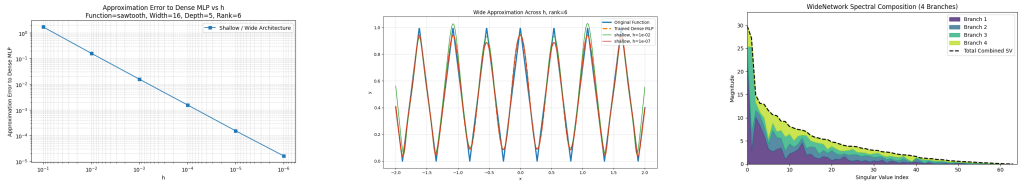
6.1 Validation of UAT

We first provide empirical evidence to show that the deep and wide structures proposed in Theorems 3 and 4 exhibit the universal approximation power as promised. Following the proof ideas of these theorems, we first train a fully connected multilayer perceptron (MLP) that approximates a given function. Then for each layer of the MLP, we construct the substructure of the deep sparse low-rank and the wide sparse low-rank network respectively and verify their approximation error to the dense MLP. We train the dense MLP in two settings: i) approximate a sawtooth function using synthetic data; ii) image classification on the MNIST dataset. The detailed experimental setup can be found in Appendix F.1.



(a) (b) (c)

Figure 2: Deep network approximation and decomposition: (2a) approximation error to dense MLP; (2b) approximation to sawtooth function as $h \to 0$; (2c) deep network spectral decomposition



(a) (b) (c)

Figure 3: Wide network approximation and decomposition: (3a) approximation error to dense MLP; (3b) approximation to sawtooth function as $h \to 0$; (3c) wide network spectral decomposition.

Sawtooth Results The sawtooth experiment is intended to validate the constructive approximation results in Theorems 1 and 2. Its non-smooth and oscillatory structure provides a nontrivial continuous target for demonstrating that the proposed DLoR architectures realize the approximation behavior predicted by the theory. We use the max-absolute-error to evaluate the approximation error. Figures 2a and 3a show that for both the deep network and the wide network, the approximation error to the dense MLP goes to 0 as $h \to 0$, validating the first part of our proofs for Theorems 3 and 4 that these structures can universally approximate dense MLP. Figures 2b and 3b show that once the trained MLP universally approximates the given test function, the constructed structures will subsequently approximate the test function universally, validating the final part of the proofs.

MNIST Results To further validate that the DLoR structures can approximate functions of higher dimensions and do not exhibit orthogonal blindness as purely low-rank structures, we construct deep and wide DLoR structures to approximate a MLP that is a classifier on the MNIST dataset. We apply the deep and wide DLoR model constructed from the baseline dense MLP to the same training and testing sets that the dense MLP is trained on and report the averaged loss and classification accuracy in table 1. The exact construction process and hyperparameters are in Appendix F.1. We observe that both constructed DLoR models have the same level of testing loss and testing accuracy as the baseline dense MLP, showing that the constructed models exhibit the same generalization performance on the test data as the dense MLP. The link between approximation and generalization can be explained by our Theorem 2 and 1: since the constructed models approximate the given MLP, and the given MLP generalizes, the constructed DLoR models will also generalize. To further validate that the constructed DLoR models are approximating the function represented by the baseline MLP, we also report max-logit-difference, max-relative-logit-difference, and relative-logit-difference as metrics for evaluating the approximation error in table 2. The exact definitions of these metrics can be found in Appendix F.3.

	Train Loss	Train Acc	Test Loss	Test Acc
Dense MLP	0.0172	0.994	0.0794	0.9805
Deep DLoR	0.0128	0.996	0.0799	0.9806
Wide DLoR	0.0125	0.996	0.0795	0.9805

Table 1: Loss and Accuracy

	$E_{\max}$	$E^{\text{rel}}_{\max}$	E_{rel}
Deep DLoR	1.26	0.0224	0.009
Wide DLoR	0.2918	0.0052	0.003

Table 2: MNIST Approximation Error

Experiment 2: Spectral Decomposition To better understand the difference between how the representational workload is divided in the substructures of the deep and wide network, we also train the DLoR deep and wide structures and visualize the singular-value decompositions of trained models inspired by Han et al. (2024). The training experiment here serves only as an illustrative demonstration of the potential mechanism underlying universal approximation of DLoR networks. A systematic study of training DLoR architectures from scratch is left to future work. We first directly train a deep DLoR structure and a wide DLoR structure from scratch to fit the synthetic data generated by the sawtooth function. After the training, we project the left and right singular vectors of the weight matrix onto each substructure. For the wide structure, the combined weight matrix is used for projection. The detailed setup can be found in Appendix F.4. For the deep structure, the Figure 2c shows that, after training, the learned transformations do not collapse to either purely low-rank approximations or purely identity mappings. Instead, the low-rank component contributes a small number of dominant singular directions, while the identity component preserves a broad background spectrum. The singular-value distribution of the DLoR layer resembles the sparse-plus-low-rank parametrization studied in Han et al. (2024), while in our DLoR structure, the sparse component is restricted to only a diagonal scaled-identity term. For the wide structure, Figure 3c shows a different pattern. Instead of components contributing separately, the combined singular-value spectrum is formed by the accumulation of many overlapping low-rank branch contributions. Each branch contributes a small amount across a broad range of singular directions. This suggests that the wide network distributes the representation across parallel branches in a less spectrally separated manner than the deep network.

7 Conclusion

In this paper, we investigated the universal approximation limits of rank-deficient neural networks. While a purely rank-1 network exhibits exact interpolation capability for scalar targets, it suffers from *Orthogonal Blindness* when extended to multi-dimensional continuous mappings. To resolve this without sacrificing parameter efficiency, we introduced a unified geometric theory based on sparse diagonal augmentation ($M = \alpha I + UV^T$). By establishing a *Structural Correspondence* framework of additive width expansion and multiplicative depth extension, we proved that Deep DLoR Neural Networks fully restore the Universal Approximation Theorem for general activation functions. These findings not only uncover the limits and power of low-rank architectures but also provide intuition for the next generation of parameter-efficient pretraining and fine-tuning frameworks.

AI use statement

In this work, generative AI tools assisted with checking proof arguments, manuscript polishing, and the implementation of codes. The authors reviewed the AI-assisted text, mathematical arguments,

and code. The authors take responsibility for the final content, including all AI-assisted text, claims, and artifacts.

Reproducibility Statement

This work is primarily theoretical. The assumptions are stated in Assumption 1, and complete proofs are provided in Appendices B–D, including the DLoR factorizations, finite-h bounds (Appendix C.5), exact finite-scale ReLU construction (Appendix C.8), and parameter counts (Appendix D.4). The experimental architectures, data processing, and evaluation metrics are described in Appendices F.1 and F.3.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 7319–7328, 2021.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Pierre Baldi. Deep learning over-parameterization: the shallow fallacy. In *Northern Lights Deep Learning Conference*, pp. 7–12. PMLR, 2024.

Dimitris Bertsimas, Ryan Cory-Wright, and Nicholas AG Johnson. Sparse plus low rank matrix decomposition: A discrete optimization approach. *Journal of Machine Learning Research*, 24 (267):1–51, 2023.

Helmut Bolcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1): 8–45, 2019.

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *International Conference on Learning Representations*, 2018.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Sparse and low-rank matrix decompositions. *IFAC Proceedings Volumes*, 42(10):1493–1498, 2009.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: unifying sparse and low-rank attention approximation, 2021.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 192–204. PMLR, 2015.

Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pp. 698–728. PMLR, 2016.

Vlad-Raul Constantinescu and Ionel Popescu. Approximation and interpolation of deep neural networks. *arXiv preprint arXiv:2304.10552*, 2023.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Ingrid Daubechies, Ronald DeVore, Simon Foucart, Boris Hanin, and Guergana Petrova. Nonlinear approximation and (deep) ReLU networks. *Constructive Approximation*, 55(1):127–172, 2022.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4133–4145, 2023.

Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. Deep neural network approximation theory. *IEEE Transactions on Information Theory*, 67(5):2581–2623, 2021.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pp. 907–940. PMLR, 2016.

Angeliki Giannou, Shashank Rajput, and Dimitris Papailiopoulos. The expressive power of tuning only the normalization layers. *arXiv preprint arXiv:2302.07937*, 2023.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik Jawanpuria, and Bamdev Mishra. Sltrain: a sparse plus low rank approach for parameter and memory efficient pretraining. *Advances in Neural Information Processing Systems*, 37:118267–118295, 2024.

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

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on learning theory*, pp. 2306–2327. PMLR, 2020.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. LoSparse: Structured compression of large language models based on low-rank and sparse approximation. In *International Conference on Machine Learning*, pp. 20336–20350. PMLR, 2023.

Zimeng Li, Li Hongjun, Jingyuan Wang, and Ke Tang. Approximation to smooth functions by low-rank swish networks. In *International Conference on Machine Learning*, pp. 35259–35291. PMLR, 2025.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. ReLoRA: High-rank training through low-rank updates. In *International Conference on Learning Representations*, 2024.

Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *International Conference on Machine Learning*, pp. 2603–2612. PMLR, 2017.

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

Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *Advances in Neural Information Processing Systems*, 33:21174–21187, 2020.

David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. In *International Conference on Learning Representations*, 2018.

Shuhua Shi, Shaohan Huang, Minghui Song, Zhoujun Li, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. ResLoRA: Identity residual mapping in low-rank adaption. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8870–8884. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.525. URL https://aclanthology.org/2024.findings-acl.525/.

Jonathan W. Siegel. Optimal approximation rates for deep relu neural networks on sobolev and besov spaces. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3645–3650, 2019.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, pp. 1517–1539. PMLR, 2016.

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of LoRA: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*, 2024.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural networks*, 94: 103–114, 2017.

Xiaoming Yuan and Junfeng Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *Pacific Journal of Optimization*, 9(1):167–180, 2013. URL http://www.ybook.co.jp/online2/oppjo/vol9/p167.html.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. In *International Conference on Learning Representations*, 2018.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *International Conference on Learning Representations*, 2024.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Longteng Zhang, Sen Wu, Shuai Hou, Zhengyu Qing, Zhuo Zheng, Danning Ke, Qihong Lin, Qiang Wang, Shaohuai Shi, and Xiaowen Chu. Salr: Sparsity-aware low-rank representation for efficient fine-tuning of large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):28337–28345, 2026. doi: 10.1609/aaai.v40i33.40062. URL https://ojs.aaai.org/index.php/AAAI/article/view/40062.

Xiao Zhang, Lingxiao Wang, and Quanquan Gu. A unified framework for nonconvex low-rank plus sparse matrix recovery. In *International Conference on Artificial Intelligence and Statistics*, pp. 1097–1107. PMLR, 2018.

A Background

Implicit Regularization vs. Explicit Rank Constraints. A parallel line of literature has studied the implicit regularization of gradient descent. Foundational work has demonstrated that deep, over-parameterized neural networks generalize remarkably well despite having the capacity to memorize random labels (Zhang et al., 2017; Brutzkus et al., 2018). Theoretical analysis of the optimization landscape reveals that extreme over-parameterization fundamentally smooths the loss surface, eliminating spurious local minima and ensuring global convergence (Choromanska et al., 2015; Nguyen & Hein, 2017; Yun et al., 2018). Within these over-parameterized spaces, gradient descent naturally biases deep linear neural networks towards low-rank or simple representations, effectively minimizing matrix rank rather than standard norms (Gunasekar et al., 2017; Arora et al., 2019; Razin & Cohen, 2020). However, there is a fundamental topological distinction between an over-parameterized network that implicitly learns a low-rank solution *after* exploring a high-dimensional landscape, and an architecture that is explicitly constrained to a strict low-rank manifold from initialization. Our paper formalizes why explicit constraints fail: while implicit regularization operates within a full-rank space capable of spanning all dimensions, explicit rank constraints physically sever the network's capacity to process orthogonal variations, leading to catastrophic geometric collapse.

B Proofs for Section 3

Notation. Throughout this paper, we establish the following mathematical conventions. Scalars are denoted by lowercase letters (e.g., α, c, h). Vectors are denoted by lowercase letters (e.g., x, u, v), and matrices by uppercase letters (e.g., W, U, V). The i-th element of a vector v is denoted by v_i, and the (i, j)-th entry of a matrix W is denoted by $W_{i,j}$. The identity matrix of dimension $N \times N$ is denoted by I_N, and the M-dimensional column vector of all ones is denoted by $\mathbf{1}_M$. The superscript T denotes the transpose of a matrix or vector. We use $\|\cdot\|$ to denote the standard Euclidean norm for vectors and the induced spectral norm for matrices. The Lebesgue measure on $\mathbb{R}^d$ is implicitly assumed when discussing probability distributions. Function composition is denoted by $\circ$, and the convolution of a function f with a measure μ is denoted by $f * \mu$. We use $Z_{a:b}$ to denote the submatrix of Z formed by columns a through b. For asymptotic limits, $f(h) = o(h)$ implies $\lim_{h\to 0} \|f(h)\|/h = 0$ uniformly, and $f(h) = \mathcal{O}(h)$ implies $\|f(h)\| \leq Ch$ for some constant $C > 0$. For a compact set $K \subset \mathbb{R}^d$, we denote by $C(K, \mathbb{R}^m)$ the space of all continuous functions $f : K \to \mathbb{R}^m$, equipped with the sup-norm $\|f\|_\infty := \sup_{x\in K} \|f(x)\|_2$. For products representing forward propagation, we use the ordered convention $\prod_{j=1}^{k} M_j := M_k \cdots M_1$.

B.1 Example: Heaviside step function

To build intuition, we first provide a lemma regarding random projections.

Lemma 4 (Injective Random Projection)**.** *Assume $v_1 \in \mathbb{R}^d$ is sampled from a probability distribution absolutely continuous with respect to the Lebesgue measure. Then, with probability 1, the scalar projections $y_i = v_1^T x_i$ are strictly distinct for all $i \in \{1, \ldots, M\}$.*

Because the projections y_i are distinct reals, we define a permutation index $s(\cdot)$ that strictly sorts them: $y_{s(1)} < y_{s(2)} < \cdots < y_{s(M)}$. We apply the identical permutation to the targets to obtain $z_{s(i)}$.

If we temporarily define $\sigma(x)$ as the Heaviside step function where $\sigma(x) = 1$ if $x \geq 0$ and 0 otherwise, we can construct an explicit thermometer encoding.

Theorem 7 (Thermometer Encoding)**.** *Let $N = M$ and $u_1 = \mathbf{1}_M$. For the Heaviside activation, there exist a threshold bias vector $b_1 \in \mathbb{R}^M$ and a readout vector $v_2 \in \mathbb{R}^M$ such that the purely rank-1 network achieves zero interpolation error.*

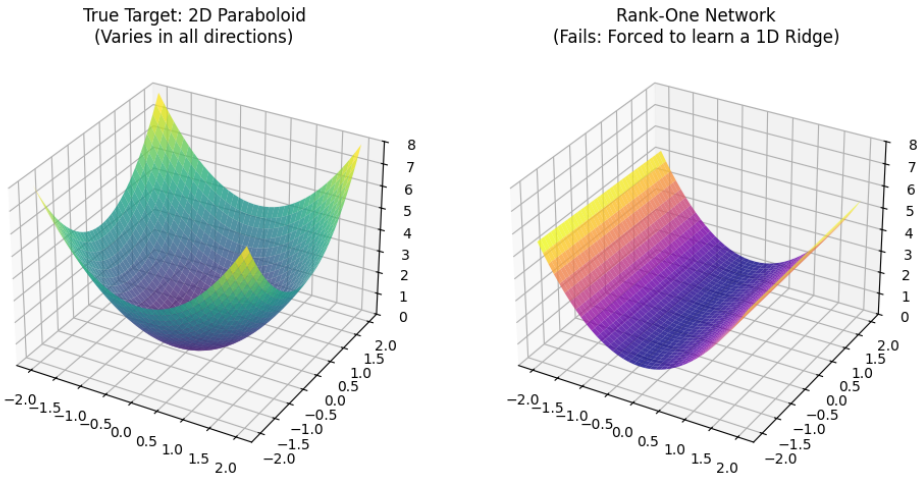


Figure 4: Illustration of approximation failure for rank-one neural networks: the target function $(x, y) \mapsto x^2 + y^2$ varies independently along two directions, while a rank-one network is restricted to a one-dimensional ridge structure. Consequently, it captures variation in only one direction and fails to approximate the full two-dimensional geometry of the function, a limitation extends to higher dimensional function approximation.

B.1.1 Proof of Lemma 4

Proof. Consider the set of all pairwise difference vectors between the input samples, defined as

$$\Delta_{ij} = x_i - x_j \tag{2}$$

for all $i \neq j$. If the scalar projections of two distinct points are equal, such that $y_i = y_j$, this strictly implies that their difference is orthogonal to the projection vector, yielding

$$v_1^T \Delta_{ij} = 0 \tag{3}$$

For any fixed, non-zero difference vector Δ_{ij}, this orthogonality condition defines a $(d-1)$-dimensional hyperplane in $\mathbb{R}^d$ passing through the origin. Because our dataset contains a finite number of samples M, there are exactly $\binom{M}{2}$ such hyperplanes. The finite union of these hyperplanes has a Lebesgue measure of strictly zero in $\mathbb{R}^d$. Consequently, because v_1 is sampled from a probability distribution that is absolutely continuous with respect to the Lebesgue measure, the probability of drawing a vector v_1 that falls perfectly inside this zero-measure union is exactly zero. Therefore, with probability 1, a randomly chosen v_1 satisfies $v_1^T \Delta_{ij} \neq 0$ for all pairs, guaranteeing strictly distinct projections. □

B.1.2 Proof of Theorem 7

Proof. We construct the threshold bias vector $b_1 = [-t_1, -t_2, \ldots, -t_M]^T$ by strategically placing the thresholds t_j between the sorted projection values using the midpoint rule. For the first threshold, we define a small margin $\delta > 0$ and set

$$t_1 = y_{s(1)} - \delta \tag{4}$$

For all subsequent thresholds where $j > 1$, we place them exactly halfway between adjacent sorted projections:

$$t_j = \frac{1}{2}\left(y_{s(j-1)} + y_{s(j)}\right) \tag{5}$$

The j-th component of the hidden layer for a given input $x_{s(i)}$ then evaluates to $h_{s(i),j} = \sigma(y_{s(i)} - t_j)$. Because the projection sequence is strictly sorted, if the index satisfies $j \leq i$, it follows that $y_{s(i)} \geq y_{s(j)}$. By the definition of our midpoints, we obtain that $y_{s(j)} > t_j$, which guarantees that the argument to the activation function is strictly positive, yielding $\sigma(y_{s(i)} - t_j) = 1$.

Conversely, if the index satisfies $j > i$, this implies $j - 1 \geq i$, meaning $y_{s(j-1)} \geq y_{s(i)}$. Our midpoint definition dictates that $t_j > y_{s(j-1)}$, resulting in a strictly negative argument and yielding $\sigma(y_{s(i)} - t_j) = 0$. Consequently, the hidden layer representation structurally collapses into a perfect thermometer encoding:

$$h_{s(i)} = (\underbrace{1, \ldots, 1}_{i}, \underbrace{0, \ldots, 0}_{M-i})^T \tag{6}$$

To finalize the exact interpolation, we construct the output weights $v_2 = [w_1, \ldots, w_M]^T$ using backward differences. We anchor the first weight to the first target by setting $w_1 = z_{s(1)}$, and compute the remaining weights as

$$w_j = z_{s(j)} - z_{s(j-1)} \tag{7}$$

for all $j \geq 2$. Summing these weights up to index i perfectly telescopes to the desired target $z_{s(i)}$, mathematically satisfying $v_2^T h_{s(i)} = z_{s(i)}$ and driving the interpolation error to zero. □

B.2 Proof of Lemma 1

A continuous function σ is mean-periodic if $\sigma * \mu = 0$ for some nonzero compactly supported signed Borel measure μ. Non-mean-periodicity ensures the linear independence of the distinct translates used below.

Proof. Fix an input projection vector v_1 that yields strictly distinct scalar projections $y_i = v_1^T x_i$. Let the hidden weights be $u_1 = \mathbf{1}_M$ and the biases be $b_1 = [b_{11}, \ldots, b_{1M}]^T$. We can explicitly define the $M \times M$ evaluation matrix H by its entries:

$$H_{i,j} = \sigma(y_j + b_{1i}) \tag{8}$$

By defining the data offsets as $\delta_j = y_j - y_1$ and the evaluation points as $t_i = y_1 + b_{1i}$, we can rewrite the matrix entries as $H_{i,j} = \sigma(t_i + \delta_j)$. This naturally motivates the definition of translated activation functions $g_j(t) = \sigma(t + \delta_j)$, allowing us to cleanly express the matrix entries as $H_{i,j} = g_j(t_i)$.

We first define the evaluation matrix $H(m)$ for some given scalar functions $g_1, \ldots, g_m$, and evaluation points $t_1, \ldots, t_m$ by its entries: $H(m)_{i,j} = g_j(t_i)$. Now we establish that if these functions $g_1, \ldots, g_m$ are linearly independent on the real line, we can recursively select the evaluation points t_i to ensure that the determinant of $H(m)$ is strictly non-zero by induction on m. Then by taking $m = M$ and setting $H = H(M)$, we prove the existence of the invertible evaluation matrix H. **Base case**: when $m = 1$, for a single dimension, linear independence ensures that g_1 is not the zero function, allowing us to select a point t_1 where $g_1(t_1) \neq 0$. For the induction hypothesis, assuming this property holds for $m - 1$ functions. At the inductive step, given linearly independent functions $g_1, \ldots, g_m$, by the induction hypothesis, there exists $t_1, \ldots, t_{m-1}$ such that $H(m-1)$ is strictly invertible. To extend $H(m-1)$ to $H(m)$, we evaluate the determinant of $H(m)$ by expansion along the final row as a function of an arbitrary evaluation point x:

$$f(x) = \sum_{i=1}^{m} C_i g_i(x) \tag{9}$$

The last coefficient of this expansion is exactly the determinant of $H(m-1)$, $C_m = \det(H(m-1))$, which is non-zero by our inductive hypothesis. Because the functions are linearly independent, this linear combination is strictly non-zero, ensuring there exists a valid evaluation point $t_m = x$ that renders the full matrix $H(m)$ invertible.

To formalize the required linear independence of the translated functions, we assume by contradiction that they are linearly dependent. This implies the existence of real constants c_j, not all zero, such that their linear combination vanishes for all real t:

$$\sum_{j=1}^{M} c_j \sigma(t + \delta_j) = 0 \tag{10}$$

This dependency perfectly describes the convolution of the activation function σ with a discrete measure μ. We explicitly construct this measure as a linear combination of Dirac measures centered

at the negative offsets, defined as

$$\mu = \sum_{j=1}^{M} c_j \delta_{-\delta_j} \tag{11}$$

Because the original projections y_j are strictly distinct, the corresponding offsets δ_j are also distinct. Combined with the non-zero constants c_j, this makes μ a strictly non-zero, compactly supported measure. This convolutional relationship forces σ to be a mean-periodic function, which directly contradicts our initial premise. Thus, the translated functions are linearly independent, the matrix H is strictly invertible, and we can compute the exact linear readout weights as $v_2^T = ZH^{-1}$. □

B.3 Proof of Lemma 2

Proof. For every hidden state $h(x)$, the output satisfies $W_2 h(x) + b_2 \in b_2 + \mathrm{Im}(W_2)$. This affine subspace has dimension $\mathrm{rank}(W_2) \le r_2$. Hence an interpolated target set must have affine span of dimension at most r_2. For $r_2 = 1$, all outputs lie on an affine line, so three noncollinear targets cannot be interpolated. □

B.4 Proof of Zero Interpolation Error for Full Outer Layer

In this subsection, we provide the formal theorem and proof supporting the claim that a network with a full-capacity outer layer can achieve exactly zero interpolation error on multidimensional targets, even when the hidden layer is restricted to a strictly rank-1 structure.

Theorem 8 (Zero Error Memorization for Multi-Dimensional Targets)**.** *Let $X \in \mathbb{R}^{d\times M}$ be a dataset matrix comprising M distinct input column vectors, and let $Z \in \mathbb{R}^{k\times M}$ be the corresponding arbitrary k-dimensional target matrix. Consider the two-layer neural network parameterized in matrix form as:*

$$f(X) = U_2 V_2^T \sigma(u_1 v_1^T X + b_1 \mathbf{1}_M^T) + b_2 \mathbf{1}_M^T \tag{12}$$

where $u_1 \in \mathbb{R}^M$ and $v_1 \in \mathbb{R}^d$ form the rank-1 hidden weight matrix $W_1 = u_1 v_1^T$, $b_1 \in \mathbb{R}^M$ is the hidden bias, $U_2 V_2^T \in \mathbb{R}^{k\times M}$ represents the factorized outer layer weights, and $b_2 \in \mathbb{R}^k$ is the outer bias. $\mathbf{1}_M \in \mathbb{R}^M$ is the all-ones vector.

If the outer layer $U_2 V_2^T$ is fully parameterized (i.e., capable of expressing any $k \times M$ matrix) and σ is continuous and non-mean-periodic, there exist parameters $\{u_1, v_1, b_1, U_2, V_2, b_2\}$ such that the network achieves exactly zero interpolation error:

$$U_2 V_2^T \sigma(u_1 v_1^T X + b_1 \mathbf{1}_M^T) + b_2 \mathbf{1}_M^T = Z \tag{13}$$

Proof. Choose v_1 so that $t_i = v_1^T x_i$ are pairwise distinct, and set $u_1 = \mathbf{1}_M$. The proof of Lemma 1 selects biases $b_{1,j}$, conditioned on these fixed projections, for which

$$H_{j,i} = \sigma(t_i + b_{1,j})$$

is nonsingular. The same hidden matrix H can be used for all k output coordinates.

The network output is

$$f(X) = U_2 V_2^\top H + b_2 \mathbf{1}_M^\top.$$

Let $W_2 := U_2 V_2^\top$. We require

$$W_2 H + b_2 \mathbf{1}_M^\top = Z.$$

Since H is invertible, for any b_2 we can set

$$W_2 = (Z - b_2 \mathbf{1}_M^\top) H^{-1}.$$

By assumption, the factorization $U_2 V_2^\top$ can represent any matrix in $\mathbb{R}^{k\times M}$ (equivalently, its factorization rank is at least $\min(k, M)$). Thus there exist U_2, V_2 such that $U_2 V_2^\top = W_2$.

With the above construction, we obtain $f(X) = Z$, completing the proof. □

C Proofs for Section 4

C.1 Proof of Lemma 3

Remark. Lemma 3 is a result in (Kidger & Lyons, 2020). We include their constructive proof for completeness and to improve readability, using arguments adapted to our setting.

Proof. By the Taylor expansion of the activation function ρ around the expansion point c, we have:

$$\rho(hx + c) = \rho(c) + h\rho'(c)x + \mathcal{R}(hx) \tag{14}$$

where the Taylor remainder strictly satisfies $\|\mathcal{R}(hx)\| = o(h)$ uniformly for all x within any compact set. Substituting the expansion into the outer affine function Ψ_h, we obtain:

$$\Psi_h(\rho(\Phi_h(x))) = \frac{\rho(c) + h\rho'(c)x + o(h) - \rho(c)}{h\rho'(c)} \tag{15}$$

The constant $\rho(c)$ terms in the numerator perfectly cancel. Dividing the remaining terms by the denominator $h\rho'(c)$ simplifies the expression to:

$$x + \frac{o(h)}{h\rho'(c)} = x + o(1) \tag{16}$$

Because $\rho'(c)$ is a non-zero constant, as the scaling parameter $h \to 0$, the $o(1)$ residual term uniformly vanishes. Consequently, the composition uniformly converges to exactly x. □

C.2 Proof of Theorem 1

Proof. By the Taylor expansion of the activation function ρ around the expansion point c, for any bounded input $x \in \mathcal{K}$, we have:

$$\rho\left(\frac{h}{\beta_l}M_l x + c\mathbf{1}_N\right) = \rho(c)\mathbf{1}_N + \frac{h}{\beta_l}\rho'(c)M_l x + \mathcal{E}_l(h) \tag{17}$$

Because the derivative ρ' is continuous and the domain $\mathcal{K}$ is compact, the remainder term is uniformly bounded, satisfying $\|\mathcal{E}_l(h)\| = o(h)$. We substitute this expansion into the left side of our additive ensemble equation:

$$\sum_{l=1}^{L} \frac{\beta_l}{h\rho'(c)}\left[\rho(c)\mathbf{1}_N + \frac{h}{\beta_l}\rho'(c)M_l x + \mathcal{E}_l(h)\right] \tag{18}$$

Distributing the summation yields three distinct terms:

$$\frac{\rho(c)}{h\rho'(c)}\mathbf{1}_N\left(\sum_{l=1}^{L}\beta_l\right) + \sum_{l=1}^{L}M_l x + \sum_{l=1}^{L}\frac{\beta_l}{h\rho'(c)}\mathcal{E}_l(h) \tag{19}$$

Crucially, because we enforce the structural constraint $\sum_{l=1}^{L}\beta_l = 0$, the zero-order bias term perfectly annihilates. Since we defined the exact additive decomposition as $W = \sum_{l=1}^{L}M_l$, the first-order term simplifies exactly to Wx. The entire expression reduces to:

$$Wx + \sum_{l=1}^{L}\frac{\beta_l}{h\rho'(c)}\mathcal{E}_l(h) \tag{20}$$

Because $\|\mathcal{E}_l(h)\| = o(h)$, the scaled remainder term strictly evaluates to $\frac{1}{h}o(h) = o(1)$. As the scaling parameter $h \to 0$, the residual error uniformly vanishes, and the ensemble converges exactly to the dense affine transformation Wx. □

C.3 Proof of Theorem 2

Proof. Let $Z \in \mathbb{R}^{N\times N}$ be an arbitrary invertible change-of-basis matrix. We define the block matrix $\mathcal{M}_k(Z)$ formed by taking the first $k \cdot r$ columns from WZ and the remaining columns from Z. Because W is an isomorphism, the set of matrices Z for which the column subspaces of $WZ_{1:kr}$ and $Z_{kr+1:N}$ act as complementary subspaces forms a Zariski dense open set in $\mathbb{R}^{N\times N}$. Since the finite intersection of dense open sets remains dense, it is guaranteed that there exists an invertible matrix Z such that the determinant $\det(\mathcal{M}_k(Z)) \neq 0$ for all $k \in \{1, \dots, L-1\}$. We fix one such matrix Z.

To justify nonemptiness for each k, fix a kr-dimensional subspace A. The equal-dimensional subspaces A and WA have a common complement C. Choosing the first kr columns of Z as a basis of A and the remaining columns as a basis of C makes both Z and $\mathcal{M}_k(Z)$ invertible. Thus the relevant determinant polynomials are nonzero, and their finite product is nonzero. One choice of Z therefore satisfies all these conditions simultaneously.

Choosing an arbitrary scalar $\alpha \neq 0$, we define the global error matrix:

$$E = W - \alpha^L I_N \tag{21}$$

We decompose this error matrix E into L rank-bounded components using our chosen basis Z. Let $C_j \in \mathbb{R}^{N\times r_j}$ represent the j-th block of columns of the product EZ, and let $D_j^T \in \mathbb{R}^{r_j\times N}$ represent the corresponding j-th block of rows of the inverse basis Z^{-1}. By block matrix multiplication, we can express the error as:

$$E = \sum_{j=1}^{L} C_j D_j^T \tag{22}$$

Since the inner dimension is exactly $r_j \leq r$, each term strictly has a rank of at most r. We now define a sequence of partial product matrices for $k = 0, \dots, L$:

$$P_k = \alpha^{k-L}\left(\alpha^L I_N + \sum_{j=1}^{k} C_j D_j^T\right) \tag{23}$$

Notice that the boundary conditions perfectly align, yielding $P_0 = I_N$ and $P_L = W$. Consider the product $P_k Z$. Because D_j^T extracts the j-th block of columns from any matrix multiplied by Z, the columns of $P_k Z$ perfectly match the columns of $\mathcal{M}_k(Z)$ up to scalar multiples of α. Factoring out these scalars yields the proportional determinant relationship $\det(P_k Z) \propto \det(\mathcal{M}_k(Z))$. Since we explicitly chose Z such that $\det(\mathcal{M}_k(Z)) \neq 0$, it strictly follows that $\det(P_k) \neq 0$, making every intermediate matrix P_k globally invertible.

We explicitly construct our sparse low-rank components by setting $U_k = \alpha^{k-L} C_k$ and $V_k = (P_{k-1}^{-1})^T D_k$. We verify this decomposition by defining the iterative product $Q_k = \prod_{j=1}^{k}(\alpha I_N + U_j V_j^T)$ and demonstrating that $Q_k = P_k$ by mathematical induction. For the base case $k = 0$, $Q_0 = P_0$. Assuming $Q_{k-1} = P_{k-1}$, we evaluate the next step:

$$Q_k = (\alpha I_N + U_k V_k^T) P_{k-1} = \alpha P_{k-1} + U_k\left(V_k^T P_{k-1}\right) \tag{24}$$

By substituting our definition of V_k, the term $V_k^T P_{k-1}$ simplifies exactly to D_k^T. This reduces the equation to:

$$Q_k = \alpha P_{k-1} + \alpha^{k-L} C_k D_k^T \tag{25}$$

Expanding the αP_{k-1} expression and adding the low-rank term incorporates the k-th block of the decomposition, yielding exactly P_k. By mathematical induction, the final product evaluates to $Q_L = P_L = W$.

Having established the algebraic decomposition $W = \prod_{l=1}^{L} M_l$, we now rigorously prove the asymptotic convergence of the sequential mapping. Let $\mathcal{A}_h(z) = (\Psi_h \circ \rho \circ \Phi_h)(z)$ represent the intermediate identity approximation block. By substituting the definitions of Φ_h and Ψ_h and expanding the activation function via Taylor's Theorem, for any bounded input z, this block applies the identity mapping plus a vanishing uniform error: $\mathcal{A}_h(z) = z + \frac{\mathcal{R}(hz)}{h\rho'(c)}$, where the remainder strictly satisfies $\|\mathcal{R}(hz)\| = o(h)$.

Let the ideal linear state sequence be recursively defined as $Z^{(0)} = x$ and $Z^{(l)} = M_l Z^{(l-1)}$, noting that $Z^{(L-1)} = (\prod_{l=1}^{L-1} M_l)x$. We track the actual forward-propagated state of the deep network, defined recursively as $\tilde{Z}^{(0)} = x$ and $\tilde{Z}^{(l)} = \mathcal{A}_h(M_l \tilde{Z}^{(l-1)})$. We proceed by induction to show $\|\tilde{Z}^{(l)} - Z^{(l)}\| \to 0$ uniformly as $h \to 0$.

Base Case ($l = 1$): The first block evaluates to $\tilde{Z}^{(1)} = \mathcal{A}_h(M_1 x) = M_1 x + \frac{\mathcal{R}(hM_1 x)}{h\rho'(c)}$. Because x resides in a compact set $\mathcal{K}$, the linear projection $M_1 x$ is strictly bounded. The scaled remainder evaluates to $\frac{o(h)}{h} = o(1)$. Thus, $\tilde{Z}^{(1)} = Z^{(1)} + o(1)$.

Inductive Step: Assume $\tilde{Z}^{(l-1)} = Z^{(l-1)} + \mathcal{E}_{l-1}(h)$ where $\mathcal{E}_{l-1}(h) = o(1)$. The next block evaluates to $\tilde{Z}^{(l)} = \mathcal{A}_h(M_l Z^{(l-1)} + M_l \mathcal{E}_{l-1}(h))$. Applying the Taylor expansion to this specific pre-activation vector yields:

$$\tilde{Z}^{(l)} = M_l Z^{(l-1)} + M_l \mathcal{E}_{l-1}(h) + \frac{\mathcal{R}\big(hM_l(Z^{(l-1)} + \mathcal{E}_{l-1}(h))\big)}{h\rho'(c)} \tag{26}$$

Because M_l is a fixed, bounded linear operator, the transformed prior error remains $M_l \mathcal{E}_{l-1}(h) = o(1)$. Furthermore, because the argument inside the remainder scales linearly with h over a bounded state, the Taylor remainder strictly evaluates to $o(h)$, making the fraction $o(1)$. Therefore, the total accumulated error is the sum of two $o(1)$ terms, strictly remaining $o(1)$ uniformly across the domain.

Final Readout ($l = L$): By induction, the network reaches the penultimate state $\tilde{Z}^{(L-1)} = Z^{(L-1)} + o(1)$. The final layer applies the target bias b and the activation function ρ without the inverse mapping Ψ_h, evaluating to:

$$H^{(L)} = \rho\left(M_L \tilde{Z}^{(L-1)} + b\right) = \rho\left(M_L Z^{(L-1)} + b + M_L o(1)\right) \tag{27}$$

Since $M_L Z^{(L-1)} = Wx$, the inner argument converges uniformly to $Wx+b$ as $h \to 0$. Because ρ is continuous on a compact neighborhood containing both the limiting affine image and its sufficiently small perturbations, it is uniformly continuous, mathematically permitting us to pass the limit inside the function:

$$\lim_{h\to 0} H^{(L)} = \rho\left(\lim_{h\to 0}[Wx + b + o(1)]\right) = \rho(Wx + b) \tag{28}$$

This completes the proof that the sequential composition of h-parameterized DLoR components uniformly recovers the exact dense layer mapping, preserving the original width N.

Main idea. The construction replaces a dense linear map by a sequence of DLoR maps acting on a fixed-width state. The matrix factorization reproduces the target linear transformation, while centering and rescaling the activation near a point c with $\rho'(c) \neq 0$ makes each nonlinear step converge uniformly to its intended linear update on the compact set of intermediate states. Consequently, the final pre-activation converges uniformly to $Wx+b$. Uniform continuity of ρ on a compact neighborhood then gives uniform convergence of the network output to $\rho(Wx + b)$. The state width remains N throughout, while the factorization determines the depth. □

C.4 An exact rectangular lift

Proposition 2 (Rectangular shear factorization)**.** *Let $W \in \mathbb{R}^{m\times n}$, let $r \geq 1$ be an integer, and put $D = n + m$. Define $Jx = (x, 0)$ and $P(u, v) = v$. Then*

$$T_W = \begin{bmatrix} I_n & 0 \\ W & I_m \end{bmatrix} \quad \textit{satisfies} \quad PT_W J = W.$$

The matrix T_W is invertible and is a product of $q = \lceil \mathrm{rank}(W)/r \rceil$ matrices of the form $I_D + E_l$, with $\mathrm{rank}(E_l) \leq r$. For $W = 0$, the product is the identity and $q = 0$.

Proof. Choose a rank decomposition $W = \sum_{l=1}^{q} W_l$ with $\mathrm{rank}(W_l) \leq r$ and put $E_l = \begin{bmatrix} 0 & 0 \\ W_l & 0 \end{bmatrix}$. Since $E_i E_j = 0$ for all i, j,

$$(I_D + E_q) \cdots (I_D + E_1) = I_D + \sum_{l=1}^{q} E_l = T_W.$$

Moreover, $T_W^{-1} = T_{-W}$. Each E_l has rank at most r and hence admits a factorization $U_l V_l^T$. The fixed embedding and selector give $PT_W Jx = Wx$. □

The bias of an affine map can be placed in the bottom coordinates. This result is exact at the matrix level; activation-aware realization uses the normalization in Lemma 3. Its state dimension is $n+m$, whereas the width-preserving network transfer uses the separate common-width construction in Remark 4.

C.5 Finite-scale approximation bounds

Proposition 3 (Finite-scale normalization)**.** *Suppose $\rho \in C^2([c-\tau, c+\tau])$, let $a = \rho'(c) \neq 0$, and assume $|\rho''| \leq M$ on this interval. Define*

$$\mathcal{A}_h(z) = \frac{\rho(c\mathbf{1} + hz) - \rho(c)\mathbf{1}}{ha}, \qquad \kappa = \frac{M}{2|a|}.$$

If $\|z\|_\infty \leq R$ and $0 < hR \leq \tau$, then

$$\|\mathcal{A}_h(z) - z\|_\infty \leq \kappa h R^2.$$

For the additive construction, write $R_l = \sup_{x\in\mathcal{K}} \|M_l x\|_\infty$. If $hR_l/|\beta_l| \leq \tau$ for all l, its uniform linear-output error is at most

$$\kappa h \sum_{l=1}^{L} \frac{R_l^2}{|\beta_l|}.$$

For sequential normalized factors, let $z_0(x) = \widetilde{z}_0(x) = x$, $z_l = M_l z_{l-1}$, and $\widetilde{z}_l = \mathcal{A}_h(M_l \widetilde{z}_{l-1})$. Set

$$S_l = \sup_{x\in\mathcal{K}} \|z_l(x)\|_\infty, \quad q_l = \|M_l\|_{\infty\to\infty}, \quad G_0 = 0, \quad G_l = 1 + q_l G_{l-1}.$$

Choose $R > \max_{1\leq l\leq L} S_l$. If $hR \leq \tau$ and $q_l \kappa h R^2 G_{l-1} \leq R - S_l$ for $2 \leq l \leq L$, then

$$\sup_{x\in\mathcal{K}} \|\widetilde{z}_L(x) - z_L(x)\|_\infty \leq \kappa h R^2 G_L.$$

Proof. The scalar second-order Taylor remainder is at most $Mh^2|z_i|^2/2$ in coordinate i; division by $|ha|$ gives the first bound. Applying this coordinatewise to each additive branch and summing gives the second bound. For the sequential construction, suppose $e_{l-1} \leq \kappa h R^2 G_{l-1}$. The stated condition ensures

$$\|M_l \widetilde{z}_{l-1}\|_\infty \leq S_l + q_l e_{l-1} \leq R.$$

Hence $e_l \leq q_l e_{l-1} + \kappa h R^2 \leq \kappa h R^2 G_l$, and induction proves the claim. □

For a final map $z \mapsto \rho(Mz+b)$ that is Lipschitz on the relevant compact neighborhood, the incoming error is multiplied by at most its Lipschitz constant. Under continuity alone, it is controlled by the corresponding modulus of continuity. These bounds give finite positive scales for any prescribed error. Their constants depend on the factor norms, and the normalization contains $1/(h|a|)$, so the real-arithmetic bounds do not imply uniform numerical conditioning as h decreases.

C.6 Exact realization with paired activation branches

Proposition 4. *Suppose $\rho(t) - \rho(-t) = at$ for all $t \in \mathbb{R}$, where $a \neq 0$. For every DLoR matrix M,*

$$\frac{\rho(Mz) - \rho(-Mz)}{a} = Mz \qquad (z \in \mathbb{R}^N).$$

Proof. Apply the scalar identity to each coordinate of Mz. Both M and $-M$ are DLoR matrices. □

The identity holds with $a = 1$ for ReLU, Softplus, SiLU, and exact GELU. Thus each matrix factor can be realized at finite scale by two tied activation branches followed by fixed subtraction. The construction uses twice the intermediate branch width and an explicit aggregation interface. It removes the small-h normalization from this realization, while the original single-branch UAT continues to follow from Assumption 1.

C.7 A SHARP FACTOR-COUNT BOUND FOR LINEAR PRODUCTS

Proposition 5. *If $W = M_L \cdots M_1$, where $M_l = \alpha_l I_N + R_l$ and $\operatorname{rank}(R_l) \le r$, then*

$$L \ge \left\lceil \frac{\min_{\gamma \in \mathbb{R}} \operatorname{rank}(W - \gamma I_N)}{r} \right\rceil.$$

For every even N, there is an invertible real matrix for which this lower bound equals $\lceil N/r \rceil$. For the fixed shear lift T_W in Proposition 2, it equals $\lceil \operatorname{rank}(W)/r \rceil$.

Proof. Expanding one factor at a time gives

$$\operatorname{rank}\left(M_L \cdots M_1 - \Big(\prod_{l=1}^{L} \alpha_l\Big) I_N \right) \le \sum_{l=1}^{L} \operatorname{rank}(R_l) \le Lr.$$

For even N, choose W block diagonal with $N/2$ copies of $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Then $W - \gamma I_N$ is invertible for every real γ. For the shear lift, $T_W - \gamma I_{n+m}$ is invertible when $\gamma \neq 1$, and has rank $\operatorname{rank}(W)$ when $\gamma = 1$. The stated bounds follow. □

This result concerns exact linear products. The lower bound for nonlinear fixed-width networks is treated separately in Appendix F.5.

C.8 EXACT RELU SIMULATION AND NETWORK TRANSFER

Lemma 5 (Factorization of arbitrary square matrices)**.** *For $W \in \mathbb{R}^{N \times N}$ and $1 \le r \le N$, there are $q \le \lceil 3N/r \rceil$ matrices $M_j = I_N + U_j V_j^T$ with $\operatorname{rank}(U_j V_j^T) \le r$ such that $W = M_q \cdots M_1$. Singular matrices are included.*

Proof. Write a full singular value decomposition $W = Q \Sigma R^T$. Every orthogonal matrix in dimension N is a product of at most N Householder reflections. To see this, a reflection maps its first column to the first coordinate vector, unless they already agree; the resulting orthogonal matrix fixes that vector, and induction applies to its restriction to the remaining $N - 1$ coordinates. Each reflection has the form $I_N - 2vv^T$ for a unit vector v. Moreover,

$$\Sigma = \prod_{i=1}^{N} \big(I_N + (\sigma_i - 1) e_i e_i^T \big),$$

including when some singular values σ_i are zero. Concatenating these factorizations gives $W = A_p \cdots A_1$, where $p \le 3N$ and every $A_i - I_N$ has rank at most one. An identity factor can be retained when needed so that $p \ge 1$.

For any k consecutive factors, the telescoping identity

$$A_k \cdots A_1 - I_N = \sum_{i=1}^{k} A_k \cdots A_{i+1} (A_i - I_N)$$

shows that their product differs from I_N by a matrix of rank at most k; an empty product denotes I_N. Grouping consecutive factors in blocks of size at most r therefore yields the stated $q \le \lceil p/r \rceil \le \lceil 3N/r \rceil$ DLoR factors. Any update of rank at most r can be written as $U_j V_j^T$ with $U_j, V_j \in \mathbb{R}^{N \times r}$ by padding its factors with zero columns. □

Proof of Proposition 1. Use Lemma 5 to write $W = M_q \cdots M_1$. When W is invertible, the matrix factorization in Theorem 2 instead gives $q \le \lceil N/r \rceil$; the following construction applies to either factorization. Put $A_0 = I_N$, $A_j = M_j \cdots M_1$, and $t_0 = 0$. If $\mathcal{K}$ is empty the claim is immediate. Otherwise, for each $1 \le j < q$ choose

$$t_j = \left(1 + \max_{x \in \mathcal{K}} \|A_j x\|_\infty \right) \mathbf{1}_N.$$

These vectors are finite by compactness. Define $z_0 = x$ and

$$z_j = \mathrm{ReLU}\big(M_j z_{j-1} + t_j - M_j t_{j-1}\big), \qquad 1 \le j < q. \tag{29}$$

Inductively, $z_{j-1} = A_{j-1}x + t_{j-1}$ makes the argument of the activation equal to $A_j x + t_j$, whose coordinates are positive for every $x \in \mathcal{K}$. Hence $z_j = A_j x + t_j$ exactly. The final stage consequently satisfies

$$\mathrm{ReLU}\big(M_q z_{q-1} + b - M_q t_{q-1}\big) = \mathrm{ReLU}(Wx + b). \tag{30}$$

Omitting this last activation yields $Wx + b$. For $q = 1$, there are no intermediate stages and the formula holds with $z_0 = x$ and $t_0 = 0$. Only biases are added to the factorization, so every weight remains DLoR and every state has width N. The finite shifts depend on the target factors and compact domain; no bound on coefficient magnitudes is asserted. □

Proposition 6 (Exact transfer of ReLU networks)**.** *Let $F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_H}$ be a feedforward network with $H \ge 1$ affine stages, ReLU after every stage except the final affine readout, and stage dimensions $n_0, \dots, n_H \le N$. For any compact $\mathcal{K} \subset \mathbb{R}^{n_0}$ and $1 \le r \le N$, there is a DLoR network G with state width N, fixed input padding and output projection, and update rank at most r, such that $G = F$ on $\mathcal{K}$. Its number D of affine stages and allocated parameter count P satisfy*

$$D \le H \left\lceil \frac{3N}{r} \right\rceil, \qquad P \le H \left\lceil \frac{3N}{r} \right\rceil (2Nr + N + 1). \tag{31}$$

In particular, for rank-one updates, $D \le 3NH$ and $P \le 3NH(3N + 1)$.

Proof. Let $J_n : \mathbb{R}^n \to \mathbb{R}^N$ append zeros, and let $P_n : \mathbb{R}^N \to \mathbb{R}^n$ select the first n coordinates. For a source stage with weight $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$ and bias b_j, use the padded square weight and bias

$$\widehat{W}_j = J_{n_j} W_j P_{n_{j-1}}, \qquad \widehat{b}_j = J_{n_j} b_j.$$

Starting from $J_{n_0}x$, each completed hidden stage produces the zero-padded source state because $\mathrm{ReLU}(0) = 0$. The input set to each stage is compact, being the image of $\mathcal{K}$ under a continuous finite network. Apply Proposition 1 on this set to simulate each hidden stage, and use its affine-output variant for the final stage. Induction over the source stages gives exact agreement of the completed blocks with the padded source network. Applying P_{n_H} then gives $G(x) = F(x)$ on $\mathcal{K}$. Intermediate shifted states may use all N coordinates; padding is restored at each completed source stage.

Each source affine stage contributes at most $\lceil 3N/r \rceil$ DLoR stages. Counting all entries in $U, V \in \mathbb{R}^{N \times r}$, the N biases, and one diagonal scalar gives $2Nr + N + 1$ allocated parameters per stage, even when the construction fixes some entries. Fixed input padding and output projection carry no trainable parameters. This proves the bounds. □

Exact additive realization. The parallel architecture of Section 4.1 also admits an exact ReLU construction with its existing branches. Let $W = \sum_{l=1}^{L} M_l$, $L \ge 2$, and choose nonzero scalars β_l with $\sum_l \beta_l = 0$. On a nonempty compact domain $\mathcal{K}$, choose a finite scalar

$$c > \max_{1 \le l \le L} \max_{x \in \mathcal{K}} \left\| \frac{M_l x}{\beta_l} \right\|_\infty .$$

Every branch is then in the positive linear region, and

$$\sum_{l=1}^{L} \beta_l \,\mathrm{ReLU}\left(c\mathbf{1}_N + \frac{M_l x}{\beta_l} \right) = c\mathbf{1}_N \sum_{l=1}^{L} \beta_l + \sum_{l=1}^{L} M_l x = Wx. \tag{32}$$

Adding b after aggregation, followed by ReLU when the target stage is hidden, exactly realizes the desired target stage. A one-term decomposition can be extended by a zero branch to permit nonzero zero-sum coefficients. The structured rectangular aggregation is $B = [\beta_1 I_N, \dots, \beta_L I_N]$, as in the width-expanded architecture; each branch retains its DLoR form under scalar rescaling.

D PROOFS FOR SECTION 5

We benchmark our capacity against the bounded-width UAT established for full-rank dense neural networks.

Lemma 6 ((Kidger & Lyons, 2020)). *Let $\rho : \mathbb{R} \to \mathbb{R}$ be any non-affine continuous function, continuously differentiable at at least one point with a non-zero derivative. Let $\mathcal{K} \subset \mathbb{R}^n$ be compact. The class of full-rank neural networks of arbitrary depth and strictly bounded width $N = n + m + 2$, utilizing activation ρ, is dense in $C(\mathcal{K}, \mathbb{R}^m)$ with respect to the uniform norm.*

D.1 PROOF OF THEOREM 3

Proof. By Lemma 6, choose a dense network F_0 with $\|F_0 - f\|_{\infty,\mathcal{K}} < \epsilon/3$. Embed its affine stages at the common width N as in Remark 4. Perturb its finitely many square weights to invertible matrices to obtain F with $\|F - F_0\|_{\infty,\mathcal{K}} < \epsilon/3$. It remains to simulate F with error below $\epsilon/3$.

If a factorization has $L = 1$, the target affine matrix is already DLoR and no intermediate identity approximation is required. For $L \geq 2$, use the following construction. For any dense layer evaluating $x \mapsto \rho(Wx + b)$, we factor $W = \prod_{l=1}^{L} M_l$ where each component is strictly $M_l = \alpha I_N + U_l V_l^T$. Let the ideal linear state sequence be $Z^{(0)} = x$, and $Z^{(l)} = M_l Z^{(l-1)}$. We construct a sub-network of L sparse low-rank layers parameterized by $h > 0$. For the initial layer, we set the weights to $W^{(1)} = hM_1$ and the bias to $B^{(1)} = c\mathbf{1}_N$. The output computes to:

$$H^{(1)} = \rho(c\mathbf{1}_N + hZ^{(1)}) = \rho(c)\mathbf{1}_N + h\rho'(c)Z^{(1)} + \mathcal{E}^{(1)}(h) \tag{33}$$

Because the intermediate state $Z^{(1)}$ is bounded on the compact set $\mathcal{K}$, the Taylor remainder uniformly satisfies $\|\mathcal{E}^{(1)}(h)\| = o(h)$.

For all intermediate layers where $1 < l < L$, we maintain the linear state scaling by assigning the weights $W^{(l)} = \frac{1}{\rho'(c)} M_l$ and the compensating bias $B^{(l)} = c\mathbf{1}_N - \frac{\rho(c)}{\rho'(c)} M_l \mathbf{1}_N$. Assuming inductively that the previous output holds the form $H^{(l-1)} = \rho(c)\mathbf{1}_N + h\rho'(c)Z^{(l-1)} + \mathcal{E}^{(l-1)}(h)$ with a uniformly bounded $o(h)$ error, the pre-activation evaluates to:

$$A^{(l)} = W^{(l)} H^{(l-1)} + B^{(l)} = c\mathbf{1}_N + hZ^{(l)} + \frac{1}{\rho'(c)} M_l \mathcal{E}^{(l-1)}(h) \tag{34}$$

Applying the activation ρ yields an output $H^{(l)} = \rho(c)\mathbf{1}_N + h\rho'(c)Z^{(l)} + \mathcal{E}^{(l)}(h)$. Indeed, the perturbation of $c\mathbf{1}_N$ in $A^{(l)}$ is $hZ^{(l)} + o(h) = O(h)$ uniformly. Differentiability at c therefore gives a Taylor remainder $o(h)$, while the propagated previous remainder also remains $o(h)$.

For the final layer L, we reverse the scaling of h and inject the target bias b. We assign $W^{(L)} = \frac{1}{h\rho'(c)} M_L$ and $B^{(L)} = b - \frac{\rho(c)}{h\rho'(c)} M_L \mathbf{1}_N$. The final pre-activation computes to:

$$A^{(L)} = W^{(L)} H^{(L-1)} + B^{(L)} = Z^{(L)} + b + \frac{1}{h\rho'(c)} M_L \mathcal{E}^{(L-1)}(h) \tag{35}$$

Since the complete sequence yields $Z^{(L)} = Wx$, and the previous error term is bounded by $o(h)$, the scaled error term evaluates strictly to $\frac{o(h)}{h} = o(1)$. Thus, as $h \to 0$, $A^{(L)}$ uniformly converges to $Wx + b$. Uniform continuity on a compact neighborhood gives uniform convergence to $\rho(Wx + b)$ for a hidden target layer. For the target network's final affine readout, omit this last activation and use $A^{(L)}$ directly.

To compose the blocks, place each ideal intermediate image inside a fixed compact neighborhood. If e_{k-1} is the incoming state error, $\delta_k(h)$ is the uniform block error on that neighborhood, and ω_k is a modulus of continuity of the kth target layer, then

$$e_k \leq \delta_k(h) + \omega_k(e_{k-1}), \qquad e_0 = 0.$$

There are finitely many target layers, $\delta_k(h) \to 0$, and $\omega_k(t) \to 0$ as $t \to 0$. Induction therefore keeps the approximate states in these neighborhoods and makes the final simulation error smaller than $\epsilon/3$ for sufficiently small positive h. The three error contributions sum to less than ϵ, proving the claim. □

D.2 Proof of Theorem 4

Proof. Choose a finite dense network F with $\|F - f\|_{\infty,\mathcal{K}} < \epsilon/2$ by Lemma 6, using the boundary embedding in Remark 4.

Given a dense target layer evaluating $y = \rho(Wx + b)$, we apply the additive width decomposition $W = \sum_{l=1}^{L} M_l$. As rigorously established in the proof of Theorem 1, the parallel ensemble computes a pre-activation response parameterized by h:

$$\tilde{A}(x, h) = \sum_{l=1}^{L} \frac{\beta_l}{h\rho'(c)} \rho\left(\frac{h}{\beta_l} M_l x + c\mathbf{1}_N\right) \tag{36}$$

We demonstrated that this structure evaluates to $\tilde{A}(x, h) = Wx + \mathcal{E}(h)$, where the remainder term $\mathcal{E}(h) \to 0$ uniformly over the compact domain $\mathcal{K}$ as $h \to 0$.

We then inject this parallel ensemble output into a subsequent exact readout layer, or directly into the next activation sequence, augmented with the original dense bias b. This evaluates to:

$$\tilde{y} = \rho(\tilde{A}(x, h) + b) = \rho(Wx + b + \mathcal{E}(h)) \tag{37}$$

Because the chosen activation function ρ is uniformly continuous on a compact neighborhood of the affine image of $\mathcal{K}$, for any arbitrary precision bound $\delta > 0$, there exists a sufficiently small scaling parameter $h > 0$ such that the magnitude of the remainder $\|\mathcal{E}(h)\|$ is strictly small enough to guarantee $\|\tilde{y} - y\|_\infty < \delta$. For the final affine readout, omit the outer activation. The finite-composition error argument in the proof of Theorem 3 then makes the total simulation error smaller than $\epsilon/2$. The triangle inequality gives the required approximation to f. □

D.3 Proof of Theorem 5

Proof. Embed the given network F at width W as in Remark 4. For a prescribed tolerance η, perturb its square weights to obtain a network f_{dense} with invertible weights and $\|f_{\text{dense}} - F\|_{\infty,\mathcal{K}} < \eta/2$. Write the perturbed network as a composition of affine stages and hidden activations:

$$f_{\text{dense}}(x) = (\mathcal{L}_L \circ \cdots \circ \mathcal{L}_1)(x) \tag{38}$$

Here $\mathcal{L}_k(z) = \rho(W_k z + b_k)$ for $k < L$, whereas $\mathcal{L}_L(z) = W_L z + b_L$ is affine. The fixed output selector is applied after the last stage.

Set $q = \lceil W/r \rceil$. By Theorem 2, each invertible W_k has a factorization $W_k = M_{k,q} \cdots M_{k,1}$ with update ranks at most r. The corresponding DLoR block $\widetilde{\mathcal{L}}_k(\cdot, h)$ has q affine stages; for the final target stage its outer activation is omitted. On each fixed compact neighborhood of the ideal intermediate states, the block error satisfies

$$\|\tilde{\mathcal{L}}_k(z, h) - \mathcal{L}_k(z)\|_\infty = o(1) \quad (h \to 0) \tag{39}$$

This ensures that as the scaling parameter h approaches zero, the intermediate DLoR block uniformly converges to the exact dense layer mapping, $\lim_{h\to 0} \tilde{\mathcal{L}}_k(z, h) = \mathcal{L}_k(z)$, uniformly on these compact neighborhoods.

We construct the global DLoR network by sequentially composing these parameterized blocks, resulting in an expanded architecture of at most Lq affine stages:

$$\tilde{f}_{\text{DLoR}}(x, h) = (\tilde{\mathcal{L}}_L \circ \cdots \circ \tilde{\mathcal{L}}_1)(x, h) \tag{40}$$

Apply the compact-neighborhood induction from the proof of Theorem 3. Its error recursion combines the uniform block errors with the moduli of continuity of the target layers. Since the number of stages is finite, the composed network converges uniformly to f_{dense}:

$$\lim_{h\to 0} \|\tilde{f}_{\text{DLoR}}(x, h) - f_{\text{dense}}(x)\|_\infty = 0 \tag{41}$$

Choose a finite $h^* > 0$ so that $\|\widetilde{f}_{\text{DLoR}}(\cdot, h^*) - f_{\text{dense}}\|_{\infty,\mathcal{K}} < \eta/2$. Writing $G = \widetilde{f}_{\text{DLoR}}(\cdot, h^*)$, we obtain

$$\|G - F\|_{\infty,\mathcal{K}} < \eta.$$

If $\|F - f\|_{\infty,\mathcal{K}} \le \epsilon$, then $\|G - f\|_{\infty,\mathcal{K}} < \epsilon + \eta$ by the triangle inequality. In particular, a dense approximation with error below half a prescribed tolerance transfers to a DLoR approximation within that tolerance. □

D.4 Proof of Theorem 6

Proof. Use Siegel (2023, Theorem 1) with $p = q = \infty$; its embedding condition is $0 < s/d$, which holds. At width $N = 25d + 31$, the infimal approximation error of a ReLU network with L hidden layers is bounded by $C_0 L^{-2s/d}$ on $\mathcal{B}_{d,s}$. These networks have $L + 1$ affine stages. Choose

$$L = \left\lceil (4 \max\{C_0, 1\}/\epsilon)^{d/(2s)} \right\rceil .$$

The infimum is then at most $\epsilon/4$, so there is a particular network F with $\|F - f\|_{L^\infty} < \epsilon/2$. For the continuous representative of f, this essential-supremum error equals the uniform error on the closed cube.

Apply Theorem 5 at rank one to all $L + 1$ affine stages of F, including its rectangular input and output stages via padding. With simulation tolerance $\epsilon/2$, the resulting G has error below ϵ. Each original affine stage uses at most N DLoR stages, each with $2N$ factor entries, N bias entries, and one diagonal scalar. Consequently,

$$D(G) \le N(L + 1),$$
$$P(G) \le (3N + 1)N(L + 1) = O_{d,s}(\epsilon^{-d/(2s)}).$$

All allocated slots are counted, including zeros and repeated scalar values; fixed padding and coordinate projection have no trainable entries. Because $N = 25d + 31$ is independent of ϵ, the DLoR conversion changes the count only by a constant depending on d. Selecting smaller perturbations or h changes coefficient values without adding slots. The real-parameter model permits the arbitrarily fine precision used by the construction. □

E Alternative Construction via Augmented State Space (Width $2N$)

While we previously established a purely multiplicative decomposition that strictly preserves the original network width N, we present here an alternative, highly intuitive additive construction. By temporarily relaxing the width constraint to $2N$, we can explicitly accumulate rank-1 updates in an augmented state space before applying the non-linear activation.

We aim to approximate a target full-rank neural network layer, defined by the mapping $x \mapsto \rho(Wx + b)$ where $W \in \mathbb{R}^{N \times N}$ and $b \in \mathbb{R}^N$. We replace this single dense layer with a block of layers in a Deep Sparse Low Rank architecture. The ideal accumulation maps have the form I_{2N} plus a rank-one update; the activation-normalized network weights are scalar multiples of these maps and remain DLoR.

We decompose the target matrix W into a sum of k independent rank-1 matrices, where the number of components $k = \text{rank}(W) \le N$:

$$W = \sum_{i=1}^{k} u_i v_i^T \tag{42}$$

where $u_i, v_i \in \mathbb{R}^N$ are column vectors. To accumulate these rank-1 projections without corrupting the original input features, we construct a network operating in an augmented, concatenated state space of width $2N$. The ideal, error-free state vector at any step l is denoted as $Z^{(l)} = \begin{bmatrix} X^{(l)} \\ Y^{(l)} \end{bmatrix} \in \mathbb{R}^{2N}$. The top block $X^{(l)} \in \mathbb{R}^N$ perfectly preserves the original input x, while the bottom block $Y^{(l)} \in \mathbb{R}^N$ acts as a memory buffer to sequentially accumulate the rank-1 transformations. We initialize this state with the zero-padded input, $Z^{(0)} = \begin{bmatrix} x \\ 0 \end{bmatrix}$.

Our strategy applies a sequence of ideal linear transformations, denoted by the matrices M_i. Each M_i represents the application of the i-th rank-1 update to the augmented state. We define $M_i = I_{2N} + \tilde{u}_i \tilde{v}_i^T$, where the augmented vectors are zero-padded to match the $2N$ dimensional space:

$$\tilde{u}_i = \begin{bmatrix} 0 \\ u_i \end{bmatrix} \in \mathbb{R}^{2N}, \quad \tilde{v}_i = \begin{bmatrix} v_i \\ 0 \end{bmatrix} \in \mathbb{R}^{2N} \tag{43}$$

Multiplying the state vector by the operator M_i perfectly preserves the identity mapping for the input block X while cleanly accumulating the rank-1 projection into the buffer block Y:

$$M_i \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} I_N & 0 \\ 0 & I_N \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} 0 \\ u_i \end{bmatrix} \begin{bmatrix} v_i \\ 0 \end{bmatrix}^T \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X \\ Y + u_i v_i^T X \end{bmatrix} \tag{44}$$

We construct a sub-network of exactly $k+1$ layers to implement these ideal M_i transformations while navigating the non-linear activation function ρ. Throughout this construction, we denote the l-th layer's weight matrix and bias vector as $W^{(l)}$ and $B^{(l)}$. The pre-activation linear state is defined as $A^{(l)} = W^{(l)} H^{(l-1)} + B^{(l)}$, and the post-activation output is $H^{(l)} = \rho(A^{(l)})$. We define an expansion point c where $\rho'(c) \neq 0$, and track the cumulative asymptotic error $E^{(l)}(h)$ between the actual network state $H^{(l)}$ and the ideal linear state $Z^{(l)}$. For brevity in our bias formulations, let $s_l = \tilde{v}_l^T \mathbf{1}_{2N}$ represent the sum of the elements in the augmented vector $\tilde{v}_l$.

To ensure rigorous uniform approximation, let $K \subset \mathbb{R}^N$ be the compact set of inputs x. The initial padded states reside in a compact set $\mathcal{K}^{(0)} \subset \mathbb{R}^{2N}$. Because each ideal operator M_i is a continuous linear transformation, the ideal forward-propagated states $Z^{(l)} = M_l \dots M_1 Z^{(0)}$ also reside in strictly bounded compact sets $\mathcal{K}^{(l)}$. By Taylor's Theorem, for any compact domain, the element-wise activation function can be expanded as:

$$\rho(c\mathbf{1}_{2N} + \Delta) = \rho(c)\mathbf{1}_{2N} + \rho'(c)\Delta + \mathcal{R}(\Delta) \tag{45}$$

where the remainder satisfies $\|\mathcal{R}(\Delta)\| = o(\|\Delta\|)$ uniformly for any bounded perturbation Δ. We proceed to construct the network layer by layer, proving inductively that the error is uniformly bounded by $o(h)$.

Layer 1 ($l = 1$): We encode the initial state into the localized linear activation regime around the expansion point c. We define the constructed weights and biases as $W^{(1)} = hI_{2N} + h\tilde{u}_1 \tilde{v}_1^T = hM_1$ and $B^{(1)} = c\mathbf{1}_{2N}$. The actual pre-activation vector is $A^{(1)} = c\mathbf{1}_{2N} + hM_1 Z^{(0)}$. Applying the activation function yields the output:

$$H^{(1)} = \rho(c\mathbf{1}_{2N} + hZ^{(1)}) = \rho(c)\mathbf{1}_{2N} + h\rho'(c)Z^{(1)} + E^{(1)}(h) \tag{46}$$

where the ideal state updates to $Z^{(1)} = M_1 Z^{(0)}$. The error term is exactly the Taylor remainder $E^{(1)}(h) = \mathcal{R}(hZ^{(1)})$. Since $Z^{(1)} \in \mathcal{K}^{(1)}$ is bounded, the perturbation magnitude is strictly $\mathcal{O}(h)$, bounding the remainder by $\|E^{(1)}(h)\| = o(h)$ uniformly.

Intermediate Layers ($1 < l \leq k$): Assume inductively that the previous output satisfies $H^{(l-1)} = \rho(c)\mathbf{1}_{2N} + h\rho'(c)Z^{(l-1)} + E^{(l-1)}(h)$, with a uniformly bounded error $\|E^{(l-1)}(h)\| = o(h)$. We define the parameters for layer l to unscale the previous output, apply the new projection M_l, and rescale back into the linear regime:

$$W^{(l)} = \frac{1}{\rho'(c)} I_{2N} + \frac{1}{\rho'(c)} \tilde{u}_l \tilde{v}_l^T = \frac{1}{\rho'(c)} M_l \tag{47}$$

$$B^{(l)} = \left(c - \frac{\rho(c)}{\rho'(c)} \right) \mathbf{1}_{2N} - \frac{\rho(c) s_l}{\rho'(c)} \tilde{u}_l \tag{48}$$

Notice that the action of the linear operator M_l on the vector of ones is exactly $M_l \mathbf{1}_{2N} = \mathbf{1}_{2N} + s_l \tilde{u}_l$. Therefore, we can rewrite the constructed bias compactly as $B^{(l)} = c\mathbf{1}_{2N} - \frac{\rho(c)}{\rho'(c)} M_l \mathbf{1}_{2N}$. Computing the pre-activation vector $A^{(l)}$ yields:

$$A^{(l)} = \frac{\rho(c)}{\rho'(c)} M_l \mathbf{1}_{2N} + hM_l Z^{(l-1)} + \frac{1}{\rho'(c)} M_l E^{(l-1)}(h) + c\mathbf{1}_{2N} - \frac{\rho(c)}{\rho'(c)} M_l \mathbf{1}_{2N} \tag{49}$$

The massive constant terms involving $\rho(c)$ perfectly cancel. Defining the new ideal state as $Z^{(l)} = M_l Z^{(l-1)}$, we obtain the simplified pre-activation:

$$A^{(l)} = c\mathbf{1}_{2N} + \underbrace{hZ^{(l)} + \frac{1}{\rho'(c)} M_l E^{(l-1)}(h)}_{\Delta^{(l)}(h)} \tag{50}$$

Passing this through the activation function gives $H^{(l)} = \rho(c)\mathbf{1}_{2N} + \rho'(c)\Delta^{(l)}(h) + \mathcal{R}(\Delta^{(l)}(h))$. Substituting the perturbation back into the linear term isolates the total error:

$$H^{(l)} = \rho(c)\mathbf{1}_{2N} + h\rho'(c)Z^{(l)} + \underbrace{M_l E^{(l-1)}(h) + \mathcal{R}(\Delta^{(l)}(h))}_{E^{(l)}(h)} \tag{51}$$

By the inductive hypothesis, the prior error is $o(h)$. Because M_l is a bounded linear operator, the transformed prior error remains $o(h)$. The perturbation satisfies $\|\Delta^{(l)}(h)\| \leq h\|Z^{(l)}\| + o(h) = \mathcal{O}(h)$, guaranteeing the Taylor remainder is $\mathcal{R}(\Delta^{(l)}(h)) = o(h)$. The sum of these terms is strictly $o(h)$, proving that the error remains uniformly bounded across all iterations. At step k, the ideal state successfully accumulates the full weight matrix: $Z^{(k)} = M_k \dots M_1 Z^{(0)} = \begin{bmatrix} x \\ Wx \end{bmatrix}$.

Final Layer ($l = k+1$): We construct the final weights and biases to apply the inverse scaling $1/h$ and inject the zero-padded target bias $\tilde{b} = \begin{bmatrix} 0 \\ b \end{bmatrix}$. We assign $W^{(k+1)} = \frac{1}{h\rho'(c)} I_{2N}$ and $B^{(k+1)} = \tilde{b} - \frac{\rho(c)}{h\rho'(c)}\mathbf{1}_{2N}$. The final pre-activation computes to:

$$A^{(k+1)} = Z^{(k)} + \tilde{b} + \frac{E^{(k)}(h)}{h\rho'(c)} \tag{52}$$

Since the tracked error is bounded by $o(h)$, the scaled error term strictly vanishes as $h \to 0$, giving $\frac{o(h)}{h} = o(1)$. Noting that $Z^{(k)} + \tilde{b} = \begin{bmatrix} x \\ Wx + b \end{bmatrix}$, the final network output vector evaluates to:

$$H^{(k+1)} = \rho\left(\begin{bmatrix} x \\ Wx + b \end{bmatrix} + o(1)\right) \tag{53}$$

Because the continuous function ρ acts on a compact domain (the affine transformation of the bounded input set K), it is uniformly continuous on this bounded image. We are mathematically permitted to pass the limit inside the function, guaranteeing that the constructed network's output converges uniformly to the exact target vector $\begin{bmatrix} \rho(x) \\ \rho(Wx + b) \end{bmatrix}$ as $h \to 0$.

Between two macroscopic layers, the ideal state $[\rho(x)^T, y^T]^T$ must be mapped to $[y^T, 0^T]^T$. Let $S = \begin{bmatrix} 0 & I_N \\ 0 & 0 \end{bmatrix}$ and $S_\delta = S + \delta I_{2N}$, where $\delta > 0$. Factor the invertible matrix S_δ into at most $2N$ rank-one DLoR factors T_j. Use the normalized blocks $\mathcal{A}_\tau = \Psi_\tau \circ \rho \circ \Phi_\tau$ after every factor, including the last:

$$R_{\tau,\delta} = \mathcal{A}_\tau \circ T_q \circ \cdots \circ \mathcal{A}_\tau \circ T_1.$$

For fixed δ, this converges uniformly to S_δ on compact sets as $\tau \to 0$. Its terminal Ψ_τ is affine, with scalar linear part, and is absorbed into the first affine stage of the next accumulation block. This preserves the DLoR constraint and introduces no additional activation of the reset state. Thus the new top block approximates y, and the bottom block approximates zero.

Theorem 9 (Universal Approximation via Augmented State Space)**.** *Under Assumption 1, let $K \subset \mathbb{R}^n$ be compact. Deep DLoR networks of arbitrary depth, state width $2(n+m+2)$, update rank at most one, fixed input padding and output coordinate selection, and an affine final readout are dense in $C(K, \mathbb{R}^m)$ with respect to the uniform norm.*

Proof. Choose a finite dense approximant of width $N = n + m + 2$ with error below $\epsilon/3$, using the fixed boundary interfaces of Remark 4. Each of its H affine stages uses an accumulation block of at most $N + 1$ stages; each of the $H - 1$ transitions uses at most $2N$ additional reset stages. Thus the construction has at most $H(N+1) + 2N(H-1)$ stages and state width $2N$. For the final target affine stage, omit the terminal activation and select the bottom output coordinates. First choose the finitely many reset perturbations small enough that their total output effect is below $\epsilon/3$. Then choose the activation scales small enough that the total simulation error is below $\epsilon/3$, using the same compact-neighborhood induction as in Theorem 3. The resulting error is below ϵ. □

F EXPERIMENTAL SETTINGS

F.1 ILLUSTRATION OF UAT

Synthetic Data We choose the sawtooth function $f(x) = |((x \cdot \lambda) \mod 2) - 1|$ to illustrate our results, where $x \in \mathbb{R}$, and λ is a given frequency constant set to $\lambda = 3.7$. The synthetic dataset consists of 400 points in the interval $[-2, 2]$ generated by the sawtooth function and is split evenly into the training and test dataset.

Baseline MLP Training. Then we implement a five-layer MLP, where the first four layers are hidden layers consist of weight matrices of dimensions 16×1 for the first layer, and 16×16 for the next three layers, biases vectors of the corresponding dimension composed with the softplus activation function at each layer. The final output layer is of dimension 1×16, and no activation function is applied as the goal is function approximation. We train the MLP using the training dataset by supervised learning with mean-squared-error loss (MSE) using the Adam optimizer with learning rate 0.01 and 2000 epochs.

Construct Deep/Wide Neural Networks. To meaningfully validate the universal approximation ability of the deep and wide neural networks, we need to guarantee the numerical stability in construction. For both the deep and wide network, we fix rank $r = 6$ to induce a moderate number of substructures to avoid overly accumulating numerical imprecision, and $c = 0.5$ where the softplus function and its first derivative are evaluated stably. For the deep network, we use $h = 10^{-2}$ to 10^{-8} and $\alpha = 0.8$ to avoid degenerative α^L. For the wide network, we use $h = 10^{-1}$ to 10^{-6}, $\beta_l = 1$ for $l = 1, \dots L-1$, and $\beta_L = -L + 1$ to enforce the affine elimination.

F.2 UAT WITH MNIST DATASET

Baseline dense MLP The dense MLP to be approximated consists of a flattening layer and three linear layers with dimensions: 784×256, 256×128, 128×10. The softplus activation function is applied after the first two linear layers. The dense MLP is trained with the cross entropy loss with the Adam optimizer for 10 epochs with learning rate 0.001 and batch size 128.

Construction of the Deep DLoR Model The construction of the deep DLoR follows from the proof of Theorem 4 using $h = 0.0001$, $\alpha = 1$, $c = 0.5$. For the three linear layers of the dense MLP, $r = 128, 64, 5$ are used respectively to factorize.

Construction of the Wide DLoR Model The construction follows from the proof of Theorem 3 using $h = 0.001$, $c = 0.5$, and $r = 5$ is used to factorize all three linear layers.

F.3 METRICS

Loss and Accuracy

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(f(x_i)_{y_i})}{\sum_{j=1}^{C} \exp(f(x_i)_j)}, \qquad \text{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\arg\max_j f(x_i)_j = y_i\right].$$

Max Logits Difference We use $1 \le i \le N$ to iterate over all the data, and $1 \le j \le 10$ denotes the 10 classes.

$$E_{\max} = \max_{1 \le i \le N} \max_{1 \le j \le 10} |f_{\text{DLoR}}(x_i)_j - f_{\text{dense}}(x_i)_j|.$$

Max Relative Logits Difference We use $1 \le i \le N$ to iterate over all the data, and $1 \le j \le 10$ denotes the 10 classes.

$$E^{\text{rel}}_{\max} = \frac{\max_{i,j} |f_{\text{DLoR}}(x_i)_j - f_{\text{dense}}(x_i)_j|}{\max_{i,j} |f_{\text{dense}}(x_i)_j|}.$$

Relative Logits Difference The logits difference is computed in Frobenius norm:

$$E_{\text{rel}} = \frac{\|F_{\text{struct}} - F_{\text{dense}}\|_F}{\|F_{\text{dense}}\|_F},$$

where

$$F_{\text{dense}} = \begin{bmatrix} f_{\text{dense}}(x_1)^\top \\ \vdots \\ f_{\text{dense}}(x_N)^\top \end{bmatrix}, \qquad F_{\text{struct}} = \begin{bmatrix} f_{\text{DLoR}}(x_1)^\top \\ \vdots \\ f_{\text{DLoR}}(x_N)^\top \end{bmatrix}.$$

F.4 Spectral Decomposition

We use a four-layer MLP where each hidden layer is of dimension 64×64 as the baseline model for constructing the deep and wide neural networks.

Notation and Setup. For the deep network, define the effective pre-activation weight matrix $W = \alpha I + UV^\top$, where $UV^\top$ is the learned low-rank component of rank r and αI is a learned scaled identity matrix. We perform a Singular Value Decomposition (SVD) on W such that $W = P\Sigma Q^\top$. To isolate the contributions of the individual structural components to the overall network capacity, we project both $UV^\top$ and αI onto the shared left and right singular vectors of W. We define and plot the following vectors: (i) total singular values $\text{diag}(\Sigma)$; (ii) low-rank contributions $\text{diag}(P^\top UV^\top Q)$; (iii) identity contributions $\text{diag}(P^\top(\alpha I)Q)$. For the fixed rank $r = 4$, the spectral decomposition of the second layer in the deep network is visualized in Figure 2c.

For the wide network, the total output of a wide layer is $\sum_{i=1}^{k} \sigma(xW_i + b_i)$. While the activation function σ prevents the exact algebraic collapse of these branches into a single matrix, we analyze the pre-activation linear span to assess structural redundancy. We define $W_{total} = \sum_{i=1}^{k} W_i$, where $W_i = \alpha_i V_i U_i^\top$. We compute the SVD of the combined matrix, $W_{total} = P\Sigma Q^\top$. To determine how much each individual branch contributes to the global pre-activation feature space, we project each W_i onto the global singular vectors. For each branch $i \in \{1, \ldots, k\}$, we compute the branch-wise contribution $|\text{diag}(P^\top W_i Q)|$. For the fixed rank $r = 4$, the spectral decomposition of the wide network layer is visualized in Figure 3c.

F.5 Lower Bound for Exact Representation of a Linear Layer

We next show that the depth required by the multiplicative construction is unavoidable in the worst case. Importantly, the lower bound continues to hold for nonlinear networks satisfying the fixed-width and Jacobian conditions stated below between consecutive diagonal-plus-low-rank transformations.

Let

$$\mathcal{D}_d := \{D \in \mathbb{R}^{d\times d} : D \text{ is diagonal}\},$$

and define the diagonal-rank complexity of a matrix $W \in \mathbb{R}^{d\times d}$ by

$$\delta(W) := \min_{D\in\mathcal{D}_d} \text{rank}(W - D).$$

Thus, $\delta(W)$ measures the minimum rank of the residual after subtracting an arbitrary diagonal matrix.

We first establish the following lemma of δ.

Lemma 7. *For any $X, Y \in \mathbb{R}^{d\times d}$,*

$$\delta(XY) \le \delta(X) + \delta(Y).$$

Proof. Let $D_X, D_Y \in \mathcal{D}_d$ be such that

$$\text{rank}(X - D_X) = \delta(X), \qquad \text{rank}(Y - D_Y) = \delta(Y).$$

Write

$$X = D_X + E_X, \qquad Y = D_Y + E_Y,$$

where

$$\operatorname{rank}(E_X) = \delta(X), \qquad \operatorname{rank}(E_Y) = \delta(Y).$$

Then

$$XY = D_X D_Y + D_X E_Y + E_X Y.$$

Since $D_X D_Y$ is diagonal,

$$\begin{aligned}\delta(XY) &\le \operatorname{rank}(XY - D_X D_Y) \\ &= \operatorname{rank}(D_X E_Y + E_X Y) \\ &\le \operatorname{rank}(D_X E_Y) + \operatorname{rank}(E_X Y) \\ &\le \operatorname{rank}(E_Y) + \operatorname{rank}(E_X) \\ &= \delta(X) + \delta(Y),\end{aligned}$$

where we used $\operatorname{rank}(AB) \le \min\{\operatorname{rank}(A), \operatorname{rank}(B)\}$. □

The following proposition is an immediate consequence of the lemma. Products of diagonal-plus-low-rank matrices have diagonal-rank complexity that grows at most linearly with depth.

Proposition 7. *Consider matrices*

$$B_\ell = D_\ell + R_\ell, \qquad D_\ell \in \mathcal{D}_d, \qquad \operatorname{rank}(R_\ell) \le r_\ell,$$

for $\ell = 1, \dots, L$, and let $M_1, \dots, M_{L-1} \in \mathcal{D}_d$ be arbitrary diagonal matrices. Then

$$P = B_L M_{L-1} B_{L-1} \cdots M_1 B_1$$

satisfies

$$\delta(P) \le \sum_{\ell=1}^{L} r_\ell.$$

In particular, if $r_\ell \le r$ for every ℓ, then

$$\delta(P) \le Lr.$$

Proof. Apply lemma 7 recursively gives the result. □

We now apply Proposition 7 to neural networks. Consider a depth-L network whose linear transformations are

$$B_\ell = D_\ell + R_\ell, \qquad \operatorname{rank}(R_\ell) \le r,$$

with input, output, and every hidden state in $\mathbb{R}^d$, and an affine final stage. Suppose that at an interior input point all coordinate activations are differentiable at their realized preactivations. The chain rule then gives a diagonal activation Jacobian at every hidden stage, so

$$J_f(x) = B_L M_{L-1}(x) B_{L-1} \cdots M_1(x) B_1,$$

where each $M_\ell(x)$ is diagonal.

Theorem 10 (Target-dependent depth lower bound)**.** *Suppose a depth-L diagonal-plus-rank-r network of constant state width d, including its input and output, exactly represents $f(x) = Wx$ on a nonempty open subset of $\mathbb{R}^d$. Assume it has an affine final stage and admits the diagonal Jacobian factorization above at an interior point. This includes networks whose activations are differentiable at all realized preactivations, as well as ReLU networks restricted to a full-dimensional affine region. Then*

$$L \ge \left\lceil \frac{\delta(W)}{r} \right\rceil,$$

where

$$\delta(W) = \min_{D \in \mathcal{D}_d} \operatorname{rank}(W - D).$$

Proof. Since $f(x) = Wx$, its Jacobian is constant:

$$J_f(x) = W.$$

On the other hand, by the chain rule,

$$J_f(x) = B_L M_{L-1}(x) B_{L-1} \cdots M_1(x) B_1,$$

where the matrices $M_\ell(x)$ are diagonal. Proposition 7 therefore yields

$$\delta(W) = \delta(J_f(x)) \le Lr.$$

Since L is an integer, rearranging gives

$$L \ge \left\lceil \frac{\delta(W)}{r} \right\rceil.$$

For ReLU networks, choose a full-dimensional polyhedral region within the open domain. On this region the activation slopes are fixed; a preactivation identically equal to zero contributes slope zero. The same diagonal factorization and rank argument therefore apply. □

The following explicit construction shows that $\delta(W)$ can be as large as $d - 1$.

Lemma 8 (An explicit hard target W)**.** *Let*

$$W_\star = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ c_1 & \lambda_2 & \ddots & \vdots \\ 0 & c_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \lambda_d \end{pmatrix},$$

where

$$c_i \neq 0, \qquad i = 1, \dots, d-1.$$

Assume also that $\lambda_i \neq 0$ for $i = 1, \dots, d$, so that $W_\star$ is invertible. Then

$$\delta(W_\star) = d - 1.$$

Proof. Let

$$D = \operatorname{diag}(d_1, \dots, d_d)$$

be arbitrary. Consider the submatrix of $W_\star - D$ formed by rows $2, \dots, d$ and columns $1, \dots, d-1$:

$$(W_\star - D)_{2:d,\,1:d-1} = \begin{pmatrix} c_1 & \lambda_2 - d_2 & 0 & \cdots & 0 \\ 0 & c_2 & \lambda_3 - d_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & c_{d-2} & \lambda_{d-1} - d_{d-1} \\ 0 & \cdots & \cdots & 0 & c_{d-1} \end{pmatrix}.$$

This matrix is upper triangular with diagonal entries $c_1, \dots, c_{d-1}$. Hence

$$\det((W_\star - D)_{2:d,\,1:d-1}) = \prod_{i=1}^{d-1} c_i \neq 0.$$

Therefore,

$$\operatorname{rank}(W_\star - D) \ge d - 1$$

for every diagonal D, and thus

$$\delta(W_\star) \ge d - 1.$$

Conversely, choosing

$$D_\star = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$$

gives

$$W_\star - D_\star = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ c_1 & 0 & \ddots & \vdots \\ 0 & c_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & c_{d-1} & 0 \end{pmatrix},$$

which has rank $d-1$. Hence

$$\delta(W_\star) \le d-1.$$

Combining the two inequalities yields

$$\delta(W_\star) = d-1.$$

□

Combining Theorem 10 and Lemma 8 gives the desired worst-case lower bound.

For every $d \ge 2$ and $r \ge 1$, there exists a linear map

$$f_\star(x) = W_\star x$$

such that any depth-L diagonal-plus-rank-r network that exactly satisfying the fixed-width and Jacobian hypotheses of Theorem 10 represents $f_\star$ must satisfy

$$L \ge \left\lceil \frac{d-1}{r} \right\rceil.$$

Consequently, the worst-case depth required to represent arbitrary dense linear transformations is

$$L = \Omega(d/r).$$

The parameterization

$$B_\ell = \alpha_\ell I + R_\ell$$

is a special case of the diagonal-plus-low-rank parameterization obtained by setting

$$D_\ell = \alpha_\ell I.$$

Hence all of the preceding results apply directly to this restricted class.