REVENGE OF MONOSEMANTICITY: NEURON SPECIALIZATION AS A NEW FORM OF FEATURE LEARNING IN MLPS

Anonymous authors
Paper under double-blind review

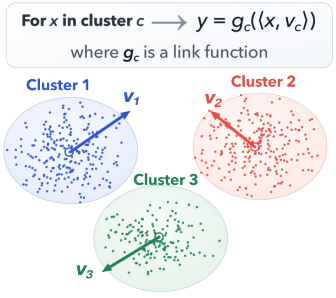
ABSTRACT

Understanding how neural networks learn and organize features is central to understanding their behavior. Much existing theory of feature learning has focused on the emergence of a global low-dimensional representation. We show that this picture is incomplete. In regression problems with clustered data, we demonstrate that multilayer perceptrons (MLPs) naturally develop monosemantic specialized neurons: individual neurons become strongly aligned with a specific predictive feature relevant to a particular region of the input space. Rather than learning a single global low-dimensional representation, MLPs learn a collection of local low-dimensional representations that can collectively span a high-dimensional space. This specialization provably gives MLPs a data-efficiency advantage over feature-learning methods based on a global low-dimensional representation.

1 INTRODUCTION

A major advance in understanding neural networks was the recognition that neural networks can perform feature learning by exploiting low-dimensional structure in the target, where prediction depends on only a small number of directions in the input space. In such settings, neural networks can adapt their representations to identify these predictive directions (Bach, 2017; Soltanolkotabi, 2017; Yehudai & Shamir, 2019; Wei et al., 2019; Ghorbani et al., 2019; 2020b; Malach et al., 2021; Abbe et al., 2022; 2023; Ba et al., 2022; Damian et al., 2022; Dandi et al., 2024; Bruna & Hsu, 2025). In this setting, the response takes the form $f(x) = g(U^\top x)$, with $U \in \mathbb{R}^{d\times r}$ and $r \ll d$, where $g : \mathbb{R}^r \to \mathbb{R}$ is the *link function*. Although $x \in \mathbb{R}^d$, the response depends only on the r-dimensional predictive subspace spanned by the columns of U. During training, a neural network can learn features aligned with that predictive subspace, effectively reducing the dimensionality of the problem from the ambient dimension d to the intrinsic dimension r. This perspective views feature learning primarily as a mechanism for discovering low-dimensional predictive structure.

In our work, we identify a qualitatively different mechanism of feature learning in multilayer perceptrons (MLPs) that is not tied to recovering a single global low-dimensional predictive subspace. In this setting, the data are drawn from multiple clusters, and each cluster can have its own predictive directions and link function. Thus, prediction may be low-dimensional within each cluster even when there is no low-dimensional global structure. The diagram on the right illustrates the *single-index* case, where each cluster has one predictive direction. As we will see, MLPs can simultaneously discover the cluster structure and learn the predictive functions relevant to each cluster. This uncovers a form of feature learning that goes beyond recovering a single global low-dimensional predictive subspace.



Main contributions.

We show that, in trained MLPs, a substantial fraction of individual neurons become *monosemantic*, specializing by aligning predominantly with a single cluster-specific predictive direction. This specialization allows the MLP to learn both the relevant local low-dimensional features and an implicit clustering that determines where each feature is useful. The resulting behavior resembles a

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.

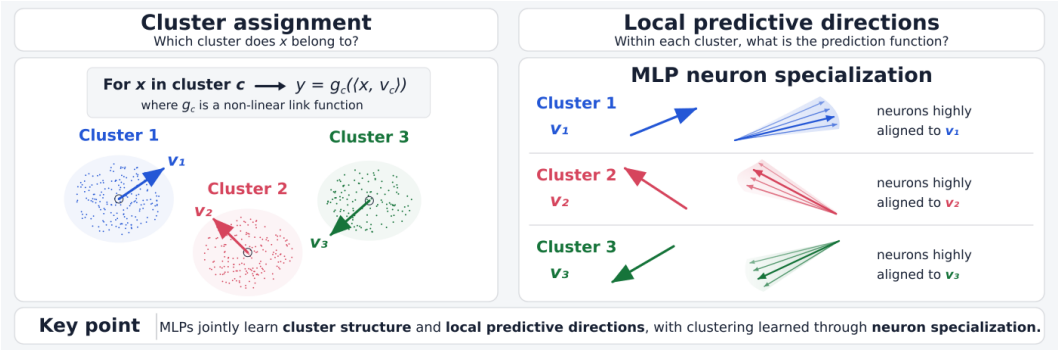
mixture of experts, but emerges within a standard MLP without an explicit routing module or expert decomposition.

We show empirically that this specialization leads to increasingly favorable sample-complexity scaling as the number of clusters grows, allowing MLPs to maintain strong performance even when no single global low-dimensional structure exists. While specialization occurs for standard ReLU and GELU (Hendrycks & Gimpel, 2016), modern architectures with multiplicative gating, such as ReGLU and SwiGLU (Shazeer, 2020), can substantially improve sample complexity in this setting.

We also establish a provable advantage of MLPs over methods based on a global low-dimensional representation. In particular, in our theoretical setting, as the number of clusters K increases, the MLP has the ability to achieve optimality (consistency) with only polynomially many training samples, $n = \mathrm{poly}(K)$. In contrast, methods based on a single global low-dimensional representation have a constant error when the number of samples is any polynomial in K.

1.1 Discussion

The form of feature learning identified in this work differs from classical feature learning, where the main goal is to recover a single global low-dimensional predictive subspace. In such settings, methods designed for learning low-dimensional features can perform as well as or better than MLPs. A prominent example is the Recursive Feature Machine (RFM), a supervised kernel-based method that uses input-response pairs to learn a global low-dimensional predictive subspace (Radhakrishnan et al., 2024; 2025). RFM is therefore a particularly informative comparison: like an MLP, it learns features from the data rather than operating with a fixed representation.



To study this phenomenon, we consider a mixture of single- or multi-index models. Suppose the data are drawn from K clusters. For an input $x \in \mathbb{R}^d$ from cluster c, the target is $f(x) = g_c(U_c^\top x)$, with $U_c \in \mathbb{R}^{d\times r}$. Thus, within each cluster, prediction depends on an r-dimensional predictive subspace and a cluster-specific link function. When $K = 1$, this reduces to the standard single- or multi-index setting with a single r-dimensional predictive subspace. When $K > 1$, however, the cluster-specific predictive subspaces may collectively span a much higher-dimensional space.

1.2 Prior Work

Feature learning of global low-dimensional structure. A large body of work studies feature learning for multi-index targets. These works have shown that neural-network models can adapt to an unknown low-dimensional subspace of the inputs, succeeding where fixed-kernel methods are sample-inefficient (Bach, 2017; Soltanolkotabi, 2017; Yehudai & Shamir, 2019; Ghorbani et al., 2020b). Recently, the literature has characterized how gradient-based training succeeds in recovering this subspace for isotropic data (Abbe et al., 2022; 2023; Damian et al., 2022; Ba et al., 2022; Dandi et al., 2024; Damian et al., 2025; Bruna & Hsu, 2025), and in the related setting of classifying Gaussian-mixture data with a constant number of clusters (Refinetti et al., 2021; Ben Arous et al., 2024). However, in our work, we prove that neuron specialization is a different form of feature learning, allowing MLPs to efficiently tackle settings where a single global low-dimensional representation is insufficient.

Neuron specialization. The emergence of specialized neurons has been observed in several settings. For instance, training neural networks to do modular arithmetic leads to specialized neurons, each representing different Fourier components (Nanda et al., 2023; Gromov, 2023; Morwani et al., 2024; He et al., 2026a;b). When learning XOR-type targets, it has also been shown that neurons specialize to four clusters rather than being distributed evenly across a predictive low-dimensional subspace (Frei et al., 2023; Glasgow, 2024). In teacher–student settings, student neurons have also been shown to align frequently with teacher neurons (Tian, 2020; Oostwal et al., 2021; Zhu et al., 2025), although this depends on initialization (Jarvis et al., 2025). These works mainly study specialization to globally relevant features or to cluster directions that themselves determine the target. In our work, neurons specialize to locally predictive features. Thus, an MLP must not only learn the predictive directions, but also preserve their association with the clusters in which they are relevant. Moreover, these works do not isolate a sample-complexity advantage of specialization over an adaptive feature-learning method based on a single global representation, such as the Recursive Feature Machine (Radhakrishnan et al., 2024). We establish such an advantage: when cluster-specific predictive directions collectively span a high-dimensional space, neuron specialization allows an MLP to preserve these local feature-cluster associations and achieve better sample complexity.

Appendix C briefly discusses monosemanticity in language models, specialization and latent-cluster recovery in mixture-of-experts models, and their connections to our work.

1.3 Paper Structure

Section 2 introduces the setting, Section 3 presents the experiments, and Section 4 provides the theoretical analysis.

2 Preliminaries

Throughout this work, we consider K well-separated Gaussian clusters, with cluster c drawn uniformly from $[K] := \{1, \dots, K\}$ and $x \mid c \sim \mathcal{N}(\mu_c, \Sigma_c)$, where $\mu_c \in \mathbb{R}^d$ and $\Sigma_c \in \mathbb{R}^{d\times d}$ are the cluster mean and covariance. For an input x from cluster c, most experiments and our theoretical analysis use the single-index model $y = g_c(\langle x, v_c\rangle) + \varepsilon$. Here, $v_c \in \mathbb{R}^d$ is a unit-norm predictive direction and $g_c : \mathbb{R} \to \mathbb{R}$ is a link function; both may vary across clusters. The noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is independent of x and c. In one experiment, we also consider a multi-index setting in which each cluster c has three orthonormal predictive directions $v_{c,1}, v_{c,2}, v_{c,3}$. The same cluster-specific link function is applied along each direction and the resulting components are aggregated as $y = \frac{1}{\sqrt{3}} \sum_{r=1}^{3} g_c(\langle x, v_{c,r}\rangle) + \varepsilon$. Throughout the paper, we consider the mean squared error (MSE) loss $\mathcal{L}(f) = \mathbb{E}[(f(x) - y)^2]$ for predictors $f : \mathbb{R}^d \to \mathbb{R}$.

3 Experiments

We empirically show that first-layer neurons in MLPs become monosemantic, specializing to cluster-specific predictive directions. We further show that MLPs achieve better sample complexity than RFM, which learns a global low-rank representation.

3.1 MLPs Develop Specialized First-Layer Neurons

We first demonstrate neuron monosemanticity in the clustered single-index model of Section 2.

Main findings. Figure 1 shows representative monosemantic first-layer neurons, each aligning predominantly with one cluster-specific predictive direction. By design, every predictive direction in this experiment coincides with a coordinate axis. Table 1 shows that a substantial fraction of first-layer neurons exhibit this behavior.

Data setting. Following Section 2 notation, we use $K = 10$ clusters in $d = 40$. We choose the cluster centers and predictive directions to lie along separate coordinate axes: for $j \in \{1, \dots, d\}$, $[\mu_c]_j = 20$ if $j = c$ and 0 otherwise, while $[v_c]_j = 1$ if $j = K + c$ and 0 otherwise. We further set $\Sigma_c = I_d/d$ and $\varepsilon \sim \mathcal{N}(0, 0.005^2)$. All cluster link functions are the third-Hermite polynomial.

Specialization measures. For a first-layer weight w_j, we measure predictive-direction alignment by $\max_{c\in[K]} |\langle w_j, v_c\rangle|/(\|w_j\|_2\|v_c\|_2)$. Table 1 reports the percentage of active neurons with cosine at least 0.71 or 0.90, corresponding to at least 50% of the squared weight norm along one predictive direction. We also report predictive-coordinate dominance: whether the largest predictive coordinate exceeds every other coordinate in magnitude, or twice that magnitude. This captures specialization obscured by many small coordinates; see Appendix A.1 for details.

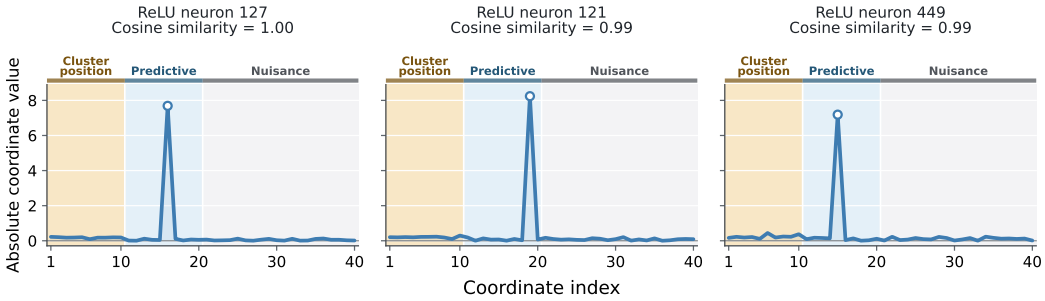


Figure 1: **Representative specialized neurons.** Specialized neurons concentrate nearly all of their weights on one cluster-specific predictive coordinate, with little weight on the cluster-position and nuisance coordinates. Because every predictive direction v_c is a standard basis vector, concentration on a predictive coordinate is equivalent to alignment with the corresponding predictive direction.

		Specialization measures (% of active neurons)			
Model/weight	**Active neurons (out of 2048)**	**Max. predictive cosine $\geq .71$ ↑**	**Max. predictive cosine $\geq .90$ ↑**	**Predictive coord. is largest ↑**	**Predictive coord. $> 2\times$ every other ↑**
ReLU	469 ± 57	56.75 ± 3.91	33.22 ± 1.92	92.6 ± 0.9	58.8 ± 3.1
GELU	$1{,}747 \pm 83$	30.15 ± 2.48	18.40 ± 1.20	71.2 ± 0.7	34.6 ± 2.2
ReGLU gate	200 ± 56	51.89 ± 1.22	33.82 ± 1.27	92.7 ± 0.6	54.2 ± 1.3
ReGLU value	200 ± 56	24.98 ± 0.52	3.94 ± 2.24	69 ± 1	22.0 ± 1.4
SwiGLU gate	432 ± 5	25.06 ± 2.29	12.26 ± 1.27	72.3 ± 0.9	23.3 ± 2.0
SwiGLU value	432 ± 5	17.88 ± 2.88	4.62 ± 0.96	86.5 ± 2.6	17.9 ± 2.1

Table 1: **First-layer neurons are monosemantic.** A large fraction of neurons aligns with a single cluster-specific predictive direction. Entries show mean $\pm$ standard deviation over three runs, resampling the training, validation, and test data, label noise, network initialization, and minibatches.

Details and additional experiments. The appendix motivates predictive-coordinate dominance as a complement to cosine similarity and illustrates it with ReLU and GELU neurons (Appendix A.1.1), defines active neurons (Appendix A.1.2), and reports the random-direction baseline (Appendix A.1.3). Training details are provided in Appendix A.1.4.

3.2 MLPs Jointly Learn Cluster Structure and Cluster-Specific Predictive Functions

We follow the single-index model of Section 2, with predictive directions and link functions varying across clusters. We compare MLPs with global predictors fit to all training data and local predictors fit separately within each true cluster; only the latter receive cluster identities.

Main findings. Figure 2 and Table 2 highlight two main conclusions:

1. **Sample complexity as the number of clusters grows.** Figure 2 shows that as the number of clusters increases, MLPs exhibit better sample complexity than global RFM. Remarkably, their sample complexity approaches that of local predictors. MLPs achieve this without observing the cluster identities. Although ReLU is already more sample efficient than global RFM, gated activations such as ReGLU improve sample efficiency further.

2. **Specialization beyond axis-aligned orthogonal directions.** Table 2 shows that a substantial fraction of active first-layer neurons specialize to the cluster-specific predictive directions. In this experiment, the predictive directions are neither axis-aligned nor constrained to be mutually orthogonal. As a baseline, we show that the alignment of the same active neurons with random directions in the same predictive subspace is negligible compared with their alignment with the true predictive directions.

Data setting. Following Section 2, we consider $K \in \{1, 2, 10, 50\}$ clusters in $d = 20$. We divide the input into ten cluster-identifying coordinates and ten predictive coordinates. The cluster centers are unit vectors $s_c \in \mathbb{S}^9 := \{s \in \mathbb{R}^{10} : \|s\|_2 = 1\}$, embedded as $\mu_c = (s_c, 0_{10})$. The cluster-specific predictive directions are sampled independently and uniformly from the unit sphere in the predictive subspace, with $\widetilde{v}_c \sim \mathrm{Unif}(\mathbb{S}^9)$ and $v_c = (0_{10}, \widetilde{v}_c)$, where 0_{10} denotes the ten-dimensional zero vector. We use covariance $\Sigma_c = \mathrm{diag}(\sigma_K^2 I_{10}, I_{10})$ and observation noise $\varepsilon \sim \mathcal{N}(0, 0.02^2)$. The centers and σ_K are chosen to keep clusters well separated as K varies; Appendix A.2.3 provides the details. Each cluster is also assigned a link function g_c, sampled uniformly from four choices: the second- and third-order Hermite polynomial, the sine function, and the hyperbolic tangent function.

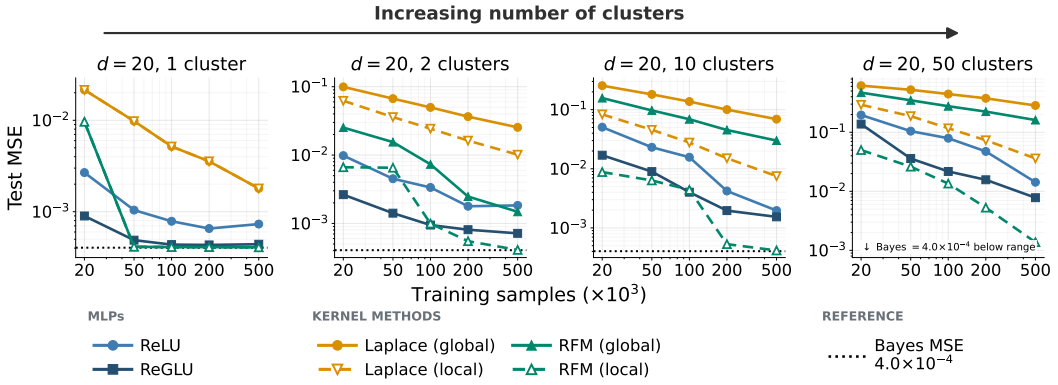


Figure 2: **MLPs jointly learn clustering and local predictive functions.** As the number of clusters grows, MLPs, especially ReGLU, maintain strong performance close to local RFM, while global RFM becomes increasingly less sample efficient. Curves show loss averaged over three runs with different link-function assignments, data, label noise, model initialization, and minibatches.

		Specialization measures (% of active neurons)			
Model/weight	**Active neurons (out of 4096)**	**True directions cosine $\geq .71$ ↑**	**Random directions cosine $\geq .71$**	**True directions cosine $\geq .90$ ↑**	**Random directions cosine $\geq .90$**
ReLU	558 ± 36	12.72 ± 3.54	0.29 ± 0.49	3.02 ± 0.79	0.00 ± 0.02
GELU	287 ± 15	16.87 ± 0.79	0.83 ± 0.85	8.01 ± 0.66	0.00 ± 0.04
ReGLU gate	133 ± 44	16.87 ± 4.31	0.56 ± 1.08	5.02 ± 1.31	0.00 ± 0.04
ReGLU value	133 ± 44	79.65 ± 4.15	6.76 ± 7.44	60.12 ± 6.44	0.05 ± 0.46
SwiGLU gate	248 ± 6	5.39 ± 1.15	0.17 ± 0.29	2.00 ± 0.77	0.00 ± 0.02
SwiGLU value	248 ± 6	16.14 ± 0.79	0.67 ± 0.85	7.16 ± 2.43	0.00 ± 0.04

Table 2: **First-layer neurons specialization.** In the single-index experiment of Fig. 2, the fraction of active neurons aligned with a true predictive direction is substantially larger than the fraction aligned with random directions in the same predictive subspace. Results use $K = 10$ clusters and $n = 500{,}000$ training samples. True-direction entries and active-neuron counts report mean $\pm$ sample standard deviation over the same three runs as Fig. 2. Appendix A.2.1 describes the active-neuron criterion and provides details of the random-direction baseline.

Details and additional experiments. Appendix A.2.3 provides the protocol and GELU and SwiGLU results. Appendix A.2.1 details the specialization measure and random-direction baseline.

Appendix A.2.2 examines performance recovery after pruning to active neurons. Appendix A.2.4 reports results with a common third-Hermite link function to match our theoretical setting.

3.3 Trained MLPs Encode Cluster Structure

In this experiment, we show that an MLP's first-layer representation captures both cluster identity and cluster-specific predictive structure. Figure 10 in Appendix A.3 provides a visual overview.

We train a ReLU MLP without cluster labels and construct representations using only its active first-layer neurons. We apply K-means to these representations. For each learned cluster, we use the corresponding active neurons' first-layer weights to construct features for an independent local Laplace or RFM predictor. At test time, each input is routed to its nearest learned centroid and evaluated by the corresponding local predictor, yielding the *MLP-gated Laplace* and *MLP-gated RFM* methods. Appendix A.3 provides experiment details.

Main findings. Figure 3 shows that fitting local Laplace and RFM predictors using clusters extracted from the MLP representation substantially improves performance over their global counterparts and achieves performance comparable to the MLP.

Data setting. We use the clustered-data construction of Section 3.2 with $K \in \{2, 5, 10, 50\}$. Each cluster's link function is sampled uniformly from the normalized third-Hermite polynomial, sine, and hyperbolic tangent.

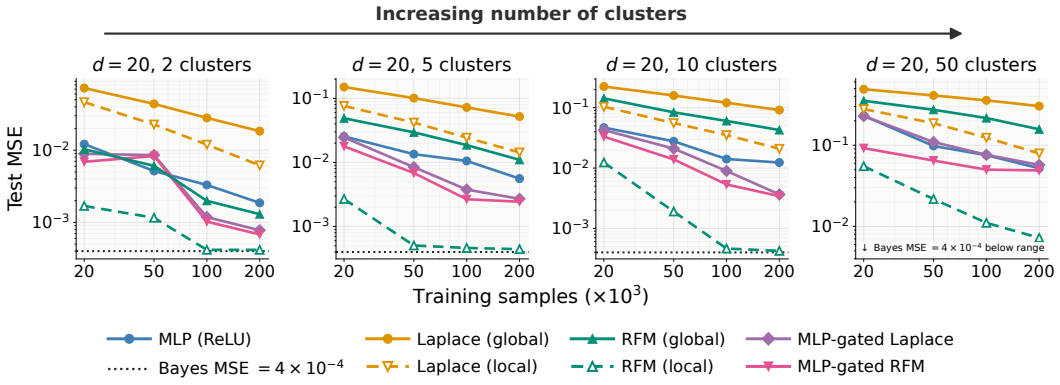


Figure 3: **Trained MLP representations encode cluster structure.** MLP-gated Laplace and RFM improve over their global counterparts and achieve performance comparable to the MLP. Curves show averaged loss over three runs with different cluster-specific link assignments, training and test examples, label noise, model and K-means initializations, and optimization minibatch orders.

3.4 MLPs Retain Their Advantage for Clustered Multi-Index Models

We extend the experiment of Section 3.2 to clustered multi-index models and show that MLPs retain their advantage over other methods. Appendix A.4 provides the experiment details.

Main findings. Figure 4 shows that MLPs retain their advantage over global Laplace and RFM as the number of clusters increases. The gated architectures generally perform best and approach local RFM at larger sample sizes. Table 3 further shows that active neurons align with cluster-specific predictive spans. Across all activations, Table 3 shows substantially greater active-neuron alignment with the cluster-specific predictive spans than with random three-dimensional subspaces of the predictive subspace; see Appendix A.4.3 for details.

Data setting. We use the multi-index model defined in Section 2, with $d = 50$, $K \in \{1, 2, 10, 25\}$, three predictive directions per cluster, and noise standard deviation $\sigma = 0.02$. The input comprises twenty cluster-identifying coordinates, twenty predictive coordinates, and ten nuisance coordinates.

Each cluster independently receives one link function, sampled uniformly from the normalized second-order Hermite polynomial, sine, and hyperbolic tangent.

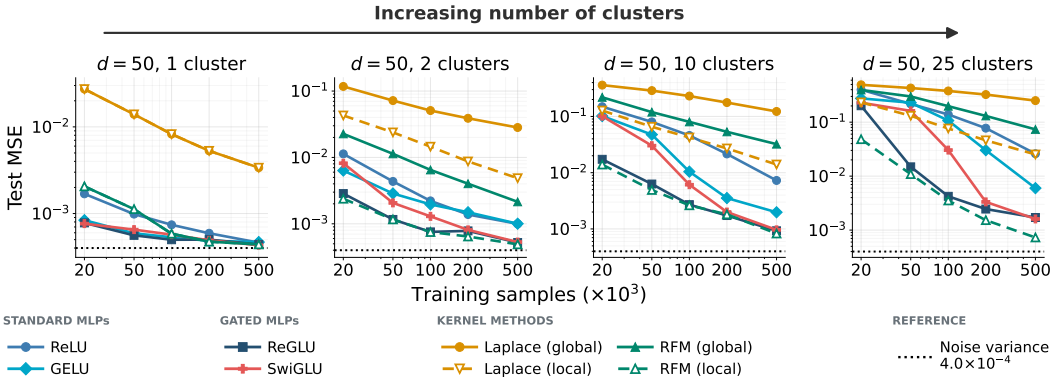


Figure 4: **MLPs retain their advantage for clustered multi-index targets.** As the number of clusters grows, MLPs, particularly ReGLU and SwiGLU, remain more sample efficient than global RFM and approach local RFM performance. Curves show mean test MSE over three runs with different link-function assignments, data, label noise, model initialization, and minibatches.

		Span cosine ≥ 0.71		Span cosine ≥ 0.90	
Model/weight	**Active neurons (out of 2048)**	**True subspace (%)**	**Random subspaces (%)**	**True subspaces (%)**	**Random subspaces (%)**
ReLU	1875 ± 126	13.07 ± 6.69	0.05 ± 0.08	1.80 ± 1.33	0.00 ± 0.00
GELU	661 ± 332	11.35 ± 5.62	0.12 ± 0.20	4.68 ± 3.47	0.00 ± 0.00
ReGLU gate	274 ± 10	8.59 ± 2.34	0.06 ± 0.15	1.59 ± 0.87	0.00 ± 0.00
ReGLU value	274 ± 10	47.46 ± 3.14	1.32 ± 0.97	27.13 ± 7.45	0.00 ± 0.01
SwiGLU gate	487 ± 21	7.29 ± 0.88	0.10 ± 0.16	3.17 ± 0.73	0.00 ± 0.00
SwiGLU value	487 ± 21	12.70 ± 1.67	0.12 ± 0.19	4.75 ± 1.09	0.00 ± 0.00

Table 3: **First-layer neurons specialize to cluster-specific predictive spans in the multi-index targets.** A substantially larger fraction of active neurons aligns with the true predictive spans than with random subspaces. Results are obtained using the same models as in Fig. 4, with $K = 10$ clusters and $n = 500{,}000$ training samples.

4 Theory

We prove in Theorem 1 that small-initialization training produces cluster-specialized MLP neurons and show in Theorem 2 that this specialization yields a sample-complexity advantage over rotationally invariant kernel methods and Recursive Feature Machines (RFM).

4.1 MLP Neurons Specialize When Learning on Gaussian Mixture Model Data

First, we show that neurons trained on well-separated Gaussian mixture data specialize to cluster-specific predictive directions at early times of training. We study K-cluster Gaussian mixture data in $d = 2K$ dimensions. Let $e_1, \ldots, e_K$ denote the standard basis of $\mathbb{R}^K$. We use the first K coordinates to separate the cluster means and the last K coordinates for the cluster-specific predictive directions.

Data Setting 1 — Symmetric Gaussian mixture

Let $R > 0$ be a cluster separation parameter and, for $c \in [K]$, let $\mu_c = [Re_c; 0]$ and $v_c = [0; e_c]$. Set each cluster to be isotropic with $\Sigma_c = I_d$ and use the common link function $g_c = h_3$, where $h_3(t) = (t^3 - 3t)/\sqrt{6}$ is the third Hermite polynomial. In the notation of Section 2, the data distribution is $c \sim \mathrm{Unif}([K])$, $x \mid c \sim \mathcal{N}(\mu_c, \Sigma_c)$, and $y = g_c(\langle x, v_c \rangle) = h_3(\langle x, v_c \rangle)$.

We will set $R > 0$ large, so that the clusters are well separated. Within cluster c, the response depends only the projection to v_c. The choice of Hermite link simplifies the analysis since $\mathbb{E}[y\,\phi(\omega^\top x)]$ has an explicit expression, letting us characterize the directions to which neurons converge.

Training Setup 1 — Two-layer ReLU population gradient flow

We train the two-layer ReLU network $f_\theta(x) = \sum_{j=1}^m a_j \phi(w_j^\top x)$, with $\phi(t) = t_+$, by population gradient flow on the squared loss $\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}[(f_\theta(x) - y)^2]$.

Initialization 1 — Small random hidden weights and zero output layer

The hidden directions are initialized uniform i.i.d., scaled by a parameter $\varepsilon > 0$, and the output layer is initialized at zero: $\omega_j^0 \overset{\text{iid}}{\sim} \text{Unif}(\mathbb{S}^{d-1})$, $w_j(0) = \varepsilon\omega_j^0$, and $a_j(0) = 0$.

Under this initialization and data distribution, the following theorem proves that neurons specialize to the clusters of the Gaussian mixture. This theoretically supports the empirics of Section 3.1.

Theorem 1 (Randomly initialized neurons specialize)**.** *Under Data Setting 1, Training Setup 1, and Initialization 1, there are universal constants $R_0, C < \infty$ such that, for every $R \geq R_0$, and almost every draw of $\omega_1^0, \ldots, \omega_m^0$, there are cluster labels $J_1, \ldots, J_m \in [K]$ and orientations $\tau_1, \ldots, \tau_m \in \{\pm 1\}$ such that, for every $\delta > 0$, there exist a finite time T_δ and $\varepsilon_0 > 0$ for which*

$$0 < \varepsilon \leq \varepsilon_0 \quad \Longrightarrow \quad \left\| \frac{w_j(T_\delta)}{\|w_j(T_\delta)\|_2} - \tau_j v_{J_j} \right\|_2 \leq \delta + \frac{C}{R} \qquad \text{for every } j \in [m].$$

Thus every neuron specializes, up to orientation and a vanishing $O(R^{-1})$ routing component, to the predictive direction of one cluster. Morever, the labels $J_1, \ldots, J_m$ are i.i.d. uniform on $[K]$, so every cluster is covered by a specialized neuron with probability at least $1 - Ke^{-m/K}$. In particular, if $m \geq K\log(K/\eta)$, then all K clusters are covered with probability at least $1 - \eta$.

Proof sketch. We reduce training to effectively independent neuron dynamics, which is a strategy used in prior work on early-time and small-initialization regimes (Abbe et al., 2022; Min et al., 2024; Glasgow, 2024). Initially, the network output is zero and for any fixed early-time interval the full network output is $O(m\varepsilon^2)$, so there is little interaction between neurons, and by a Grönwall argument a neuron-wise dynamics is a good approximation. In this dynamics, $\dot{a}_j$ is proportional to $\Phi(\omega_j) = \mathbb{E}[y\,\phi((\omega_j)^\top x)]$, so a_j immediately acquires the appropriate sign. ReLU homogeneity makes the hidden direction follow a time-reparameterized gradient ascent on the corresponding signed correlation. Using an explicit expression for $\Phi(\omega)$ based on h_3, we show each neuron converges to a one-cluster solution where it aligns with $\pm v_c$ and uses only an $O(R^{-1})$ component in the associated routing direction to place its ReLU threshold within cluster c; proof in Appendix B.1.

4.2 Specialization Yields a Sample-Complexity Gap Between MLPs and Adaptable Kernel Methods

We establish that for clustered data two-layer MLPs can have much better sample efficiency than the Recursive Feature Machines (RFM) adaptive kernel method. We study the many-cluster regime where $K \to \infty$, and show polynomial-sample consistency for MLPs, based on their ability to specialize. In contrast, we prove inconsistency of RFM under every fixed polynomial-sample scaling.

Data Setting 2 — Gaussian mixture with noiseless routing

We consider clustered data, where the clusters can now be separated noiselessly: the means are $\mu_c = [e_c; 0]$ and the predictive directions are $v_c = [0; e_c]$ in $d = 2K$ dimensions. The first K coordinates are noiseless: $\Sigma_c = \text{diag}(0_{K\times K}, I_K)$. Every cluster has the same non-constant, bounded, Lipschitz link function $g : \mathbb{R} \to \mathbb{R}$ which is independent of K. Thus, in the notation of Section 2, $c \sim \text{Unif}([K])$, $x \mid c \sim \mathcal{N}(\mu_c, \Sigma_c)$, and $y = g(\langle x, v_c\rangle)$.

Instead of studying a network trained through gradient-based updates, we study a two-layer MLP fit by empirical risk minimization subject to a Frobenius-norm constraint on its weights.

Training Setup 2 — Frobenius-constrained MLP ERM

We fit the two-layer ReLU network $f_\theta(x) = \sum_{j=1}^m a_j \phi(w_j^\top x)$, with $\phi(t) = t_+$, using empirical risk minimization. For a width m and budget $B > 0$, define $\mathcal{F}_{B,m} := \{f_\theta : \|\theta\|_{\mathrm{F}}^2 \le B\}$, where $\|\theta\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_{j=1}^m (a_j^2 + \|w_j\|_2^2)$. Given n samples, squared-loss empirical risk minimization (ERM) gives $\widehat{f}_{\mathrm{MLP}} \in \arg\min_{f \in \mathcal{F}_{B,m}} \frac{1}{n}\sum_{i=1}^n \big(f(x_i) - y_i\big)^2$.

We compare MLP ERM with standard kernel ridge regression and with RFM, a supervised kernel method (Radhakrishnan et al., 2024; 2025). To isolate the limitation of one global feature metric, we grant RFM the ground-truth population AGOP, eliminating metric-estimation error.

Training Setup 3 — Kernel methods and RFM with ground-truth AGOP

Let $\mathcal{K}$ be a rotationally invariant kernel. Given n samples, $\widehat{f}_{\mathrm{Kernel}}$ is kernel ridge regression with kernel $\mathcal{K}(x, x')$. We give the RFM comparator the ground-truth population AGOP (using gradients tangent to the data support), $M := \mathbb{E}[\nabla_x f^\star(x) \nabla_x f^\star(x)^\top]$. For $\rho \ge 0$, define the regularized AGOP $M_\rho := M + \rho I_d$ and corresponding kernel $\mathcal{K}_{M_\rho}(x, x') := \mathcal{K}(\sqrt{M_\rho}x, \sqrt{M_\rho}x')$. $\widehat{f}_{\mathrm{RFM}}$ is kernel ridge regression with kernel $\mathcal{K}_{M_\rho}$. Both estimators may use any kernel ridge parameter $\lambda_n \ge 0$.

The theorem below proves polynomial-sample consistency of the MLP and inconsistency of both kernel comparators under every fixed polynomial sample scaling. Appendix B.2 proves a more general fixed-dimensional multi-index result; here we state its single-index specialization.

Theorem 2 (Specialization enables data-efficient learning)**.** *Under Data Setting 2, the target function $f^\star$ admits approximation by cluster-specialized two-layer ReLU networks with polynomially growing width and Frobenius norm, and with approximation error tending to zero as $K \to \infty$.*

Thus, there are polynomially-growing width and norm-budget sequences (m_K) and (B_K) such that, with polynomially many samples $n_K = \Omega(K^7)$, the MLP estimator in Training Setup 2 is consistent:

$$\lim_{K\to\infty} \mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}} - f^\star\|_{L^2(P_x)}^2\right] = 0.$$

On the other hand, for every $A < \infty$, every sequence $n_K = O(K^A)$, and every sequence $\rho_K \ge 0$,

$$\liminf_{K\to\infty} \mathbb{E}\left[\|\widehat{f} - f^\star\|_{L^2(P_x)}^2\right] > 0, \qquad \widehat{f} \in \left\{\widehat{f}_{\mathrm{Kernel}}, \widehat{f}_{\mathrm{RFM}}\right\}.$$

Proof sketch. Since g is Lipschitz and bounded, it can be approximated arbitrarily well in Gaussian L^2 by a finite shallow network of ridge-ramp functions, each of which is a difference of two ReLUs. For each cluster c, we can instantiate a ridge-ramp function that is only active on that cluster as $\mathrm{ReLU}(\langle v, x\rangle + \langle e_c, x\rangle) - \mathrm{ReLU}(\langle v, x\rangle)$. This lets us approximate f^*.

Since our construction gives a Frobenius-norm-bounded network with good approximation error, standard Barron-norm Rademacher-complexity bounds control the estimation error (E et al., 2022; Bach, 2017; Golowich et al., 2018). Because Gaussian covariates are unbounded, we derive the required sample-dependent version, using concentration to control MLP predictions on typical inputs.

We prove an obstruction for kernel methods and RFM using Hermite decomposition similar to (Ghorbani et al., 2021; 2020a; Mei et al., 2021). Write the Hermite expansion of the link function, $g(t) = \sum_{q=0}^\infty \widehat{g}_q h_q(t)$, where each $\widehat{g}_q$ is a Hermite coefficient. By the properties of g, there is positive Hermite energy above every finite degree. Moreover, $M = \frac{\alpha}{K} P_V$, where $P_V := \sum_{c=1}^K v_c v_c^\top$ and $\alpha = \mathbb{E}[g'(G)^2] > 0$, so both kernel metrics remain rotationally invariant within the predictive coordinates. At Hermite order q, the cluster's target contains a component in an irreducible harmonic subspace of dimension $\Theta_q(K^q)$, but the predictor lies in the span of at most n invariantly distributed kernel sections. So, when $n = O(K^A)$, every fixed order $q > A$ leaves asymptotically all of its target energy unrecovered. Thus, both kernel estimators retain nonzero error under every fixed polynomial sample scaling.

AI Disclosure

Generative AI tools were used to provide feedback on experimental design, assist with analysis and interpretation of experimental results, assist in developing and checking portions of the theoretical arguments and proofs, and assist with manuscript preparation. They were also used to draft and revise text, improve organization and readability, edit LaTeX, and format tables and figures. All AI-assisted mathematical arguments, experimental results, and manuscript content were independently reviewed and verified by the authors. The authors take responsibility for all claims, results, and artifacts in this work.

References

Emmanuel Abbe and Enric Boix-Adsera. On the non-universality of deep learning: quantifying the cost of symmetry. *Advances in Neural Information Processing Systems*, 35:17188–17201, 2022.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4782–4887. PMLR, 2022.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: Leap complexity and saddle-to-saddle dynamics. In *Proceedings of the 36th Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 2552–2623. PMLR, 2023.

Amirhesam Abedsoltan, Mikhail Belkin, and Parthe Pandit. Toward large kernel models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 61–78. PMLR, 2023.

Amirhesam Abedsoltan, Siyuan Ma, Parthe Pandit, and Misha Belkin. Fast training of large kernel models with delayed projections. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 37932–37946, 2022.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

Gérard Ben Arous, Reza Gheissari, Jiaoyang Huang, and Aukosh Jagannath. High-dimensional SGD aligns with emerging outlier eigenspaces. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d10d6b28d74c4f0fcab588feeb6fe7d6-Abstract-Conference.html.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Joan Bruna and Daniel Hsu. Survey on algorithms for multi-index models. *Statistical Science*, 40 (3):378–391, 2025.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23049–23062, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Alex Damian, Jason D Lee, and Joan Bruna. The generative leap: Sharp sample complexity for efficiently learning gaussian multi-index models. *arXiv preprint arXiv:2506.05500*, 2025.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on learning theory*, pp. 5413–5452. PMLR, 2022.

Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. How two-layer neural networks learn, one (giant) step at a time. *Journal of Machine Learning Research*, 25(349): 1–65, 2024.

Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9376–9396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.583. URL https://aclanthology.org/2023.emnlp-main.583/.

Weinan E, Chao Ma, and Lei Wu. The barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022. doi: 10.1007/s00365-021-09549-y.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett. Random feature amplification: Feature learning and generalization in neural networks. *Journal of Machine Learning Research*, 24(303): 1–49, 2023.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? In *Advances in Neural Information Processing Systems*, volume 33, pp. 14820–14830. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/a9df2255ad642b923d95503b9a7958d8-Abstract.html.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? In *Advances in Neural Information Processing Systems*, volume 33, pp. 14820–14830, 2020b.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 2021.

Margalit Glasgow. SGD finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the XOR problem. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HgOJlxzB16.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Proceedings of the 31st Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 297–299. PMLR, 2018.

Andrey Gromov. Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*, 2023.

Jianliang He, Leda Wang, Siyu Chen, and Zhuoran Yang. On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking. *arXiv preprint arXiv:2602.16849*, 2026a.

Jianliang He, Leda Wang, Fengzhuo Zhang, Siyu Chen, and Zhuoran Yang. Neural networks provably learn spectral representations for group composition. *arXiv preprint arXiv:2606.02993*, 2026b.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Devon Jarvis, Sebastian Lee, Clémentine Carla Juliette Dominé, Andrew M. Saxe, and Stefano Sarao Mannelli. A theory of initialisation's impact on specialisation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on learning theory*, pp. 1772–1798. PMLR, 2019.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Ryotaro Kawata, Kohsei Matsutani, Yuri Kinoshita, Naoki Nishikawa, and Taiji Suzuki. Mixture of experts provably detect and learn the latent cluster structure in gradient-based learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29390–29448. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kawata25a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

Siyuan Ma and Mikhail Belkin. Kernel machines that adapt to GPUs for effective large batch training. *Proceedings of Machine Learning and Systems*, 1:360–373, 2019.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3325–3334. PMLR, 2018.

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7379–7389. PMLR, 2021.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pp. 3351–3418. PMLR, 2021.

Hancheng Min, Enrique Mallada, and René Vidal. Early neuron alignment in two-layer ReLU networks with small initialization. In *The Twelfth International Conference on Learning Representations*, 2024.

Depen Morwani, Benjamin Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *International Conference on Learning Representations*, volume 2024, pp. 29077–29114, 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Elisa Oostwal, Michiel Straat, and Michael Biehl. Hidden unit specialization in layered neural networks: ReLU vs. sigmoidal activation. *Physica A: Statistical Mechanics and its Applications*, 564:125517, 2021. doi: 10.1016/j.physa.2020.125517.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024. doi: 10.1126/science.adi5639.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Dmitriy Drusvyatskiy. Linear recursive feature machines provably recover low-rank matrices. *Proceedings of the National Academy of Sciences*, 122(13):e2411325122, 2025. doi: 10.1073/pnas.2411325122. URL https://doi.org/10.1073/pnas.2411325122.

Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborova. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8936–8947. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/refinetti21b.html.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Mahdi Soltanolkotabi. Learning ReLUs via gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Yuandong Tian. Student specialization in deep rectified networks with finite width and input dimension. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9470–9480. PMLR, 2020.

Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Zhenyu Zhu, Fanghui Liu, and Volkan Cevher. How gradient descent balances features: A dynamical analysis for two-layer neural networks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=25j2ZEgwTj.

A Additional Experimental Details

A.1 Additional Details for First-Layer Neuron Specialization

This appendix supplements Section 3.1. Section A.1.1 motivates the complementary specialization measures and illustrates them with representative ReLU and GELU neurons. Section A.1.2 introduces the active-neuron criterion and the shared analysis protocol. Section A.1.3 introduces controls based on random directions and subspaces to assess specialization, and applies this comparison to the axis-aligned experiment. Finally, Section A.1.4 gives the third-Hermite training protocol, and Section A.1.5 presents the additional second-Hermite experiment.

A.1.1 Specialization Measures and Representative Neurons

Why predictive-coordinate dominance? For an incoming weight vector w_j, define its maximum predictive-direction cosine as

$$A_j = \max_{1 \le c \le K} \frac{|\langle w_j, v_c \rangle|}{\|w_j\|_2 \, \|v_c\|_2}.$$

Cosine alignment measures how much of a neuron's total weight norm lies in a single predictive direction. It can therefore be conservative when one predictive coordinate is large but many individually small coefficients collectively contribute appreciably to the norm. Let

$$\widehat{c}_j \in \arg\max_{c \in [K]} |[w_j]_{K+c}|$$

index the predictive coordinate with the largest absolute weight. Because $v_c = e_{K+c}$ in this experiment, this is equivalent to selecting the predictive direction with the largest absolute inner product with w_j. Define the predictive-coordinate dominance ratio as

$$D_j = \frac{|[w_j]_{K+\widehat{c}_j}|}{\max\limits_{\substack{1 \le \ell \le d \\ \ell \ne K+\widehat{c}_j}} |[w_j]_\ell|}.$$

Thus, $D_j > 1$ means that the largest coordinate is predictive, while $D_j > 2$ means that this coordinate is more than twice as large as every competing coordinate. Unlike cosine alignment, this ratio asks whether a predictive coordinate dominates each alternative coordinate individually. The two measures therefore provide complementary views of specialization.

ReLU examples. Figure 5 illustrates this distinction for ReLU neurons trained with the normalized third-Hermite link function. Although the displayed neurons have only moderate cosine alignment because their norms include many smaller coefficients, each has a pronounced spike on one cluster-specific predictive coordinate. Predictive-coordinate dominance therefore captures clear specialization that a strict cosine threshold can miss.

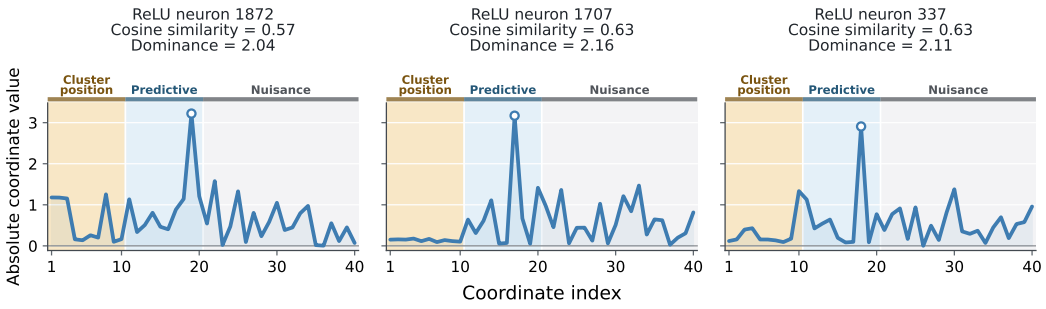


Figure 5: **Predictive-coordinate dominance complements cosine alignment.** This plot illustrates why predictive-coordinate dominance can be a more informative measure than cosine similarity with a strict threshold. A neuron may have only moderate cosine alignment because its norm includes many smaller coefficients, while still exhibiting a pronounced spike along a single cluster-specific predictive direction v_c. Recall that here, each v_c is chosen as a standard basis vector, so concentration on a predictive coordinate is equivalent to alignment with the corresponding predictive direction.

GELU examples. Figure 6 shows the corresponding behavior for GELU. The top row contains neurons with high cosine alignment, whereas the bottom row contains neurons with moderate cosine alignment but a single strongly dominant predictive coordinate. This confirms that the distinction between the two specialization measures is not specific to ReLU.

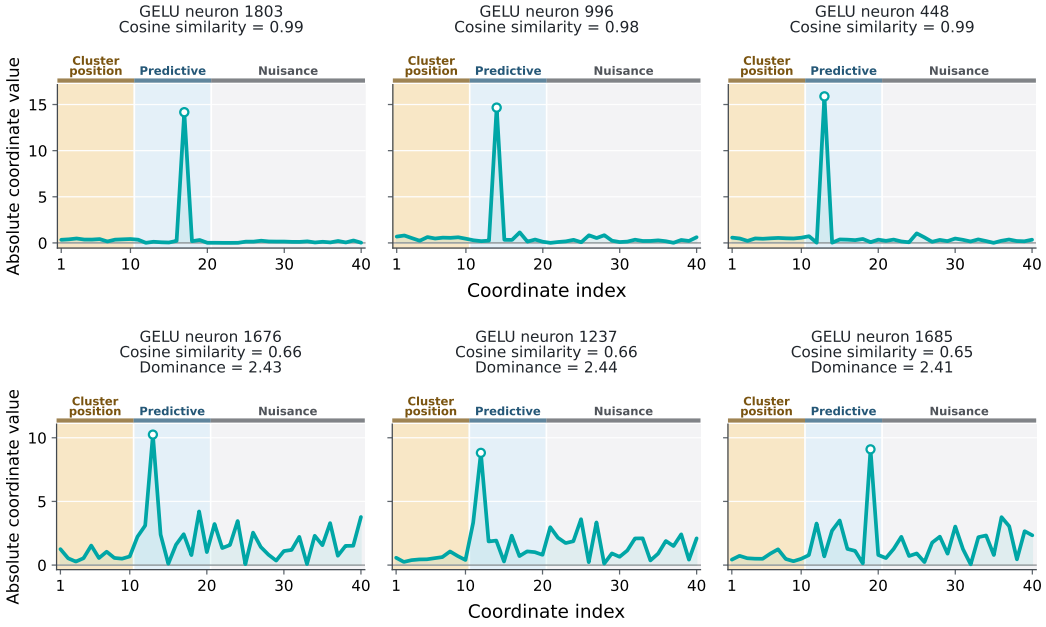


Figure 6: **Normalized third-Hermite specialization in GELU neurons.** Top: examples whose first-layer weights are nearly entirely concentrated on one predictive coordinate. Bottom: examples for which many smaller coefficients reduce cosine alignment, while one predictive coordinate remains more than twice as large as every competing coordinate.

A.1.2 Active Neurons and Specialization

Active-neuron criterion. We restrict the specialization analysis to neurons that contribute materially to the network output. For an ordinary ReLU or GELU neuron, we define its importance as

$$I_j = |a_j| \, \|w_j\|_2,$$

where a_j is its output weight and w_j is its incoming weight. For a gated ReGLU or SwiGLU neuron, we use

$$I_j = |a_j| \, \|w_{g,j}\|_2 \, \|w_{v,j}\|_2,$$

where $w_{g,j}$ and $w_{v,j}$ denote its gate and value weights.

We sort neurons in decreasing order of importance I_j, then add their importance scores in that order until the cumulative sum reaches at least 99.9% of the total importance, $\sum_j I_j$, summed over all neurons. We retain all neurons up to and including the one that reaches this threshold and refer to them as *active*. For gated architectures, the same active-neuron set is used when analyzing the gate and value weights separately.

The 99.9% threshold is a descriptive sparsification rule rather than a statistical significance threshold. Its purpose is to prevent the specialization statistics from being dominated by neurons with negligible influence on the network output.

Specialization statistics and reporting. For each active neuron, we compute the cosine-alignment score A_j and the predictive-coordinate dominance ratio D_j defined above. Absolute inner products are used because alignment with v_c and $-v_c$ represents the same predictive direction. Table 1 and the top panel of Fig. 7 report the fractions of active neurons satisfying

$$A_j \geq 0.71, \qquad A_j \geq 0.90, \qquad D_j > 1, \qquad D_j > 2.$$

For ReGLU and SwiGLU, all statistics are computed and reported separately for the gate and value weights.

The individual neurons displayed in the bottom panel of Fig. 7 are the highest-importance active GELU neurons satisfying $A_j \geq 0.90$. Their coordinate-wise absolute incoming weights are shown so that specialization is visible independently of sign. For the normalized third-Hermite link function, Fig. 1 and the top panel of Fig. 6 show high-alignment ReLU and GELU examples, respectively. Figure 5 and the bottom panel of Fig. 6 instead show neurons with only moderate A_j but large D_j, making the complementary roles of the two measures explicit for both activations.

A.1.3 Comparing Alignment with True and Random Directions

In this section, we introduce a control experiment to determine whether the observed neuron alignment reflects specialization to the true predictive directions. If we replace these directions with random directions in the same predictive subspace, would a similar fraction of active neurons still exceed the same alignment threshold?

We answer this question by comparing alignment with the true directions against alignment with randomly generated alternatives. For multi-index models, we make the analogous comparison using predictive subspaces. The random directions or subspaces match their true counterparts in number, dimension, and coordinate support, while respecting the geometric constraints of each experiment. A substantially larger fraction of neurons aligned with the true directions or subspaces provides evidence that the learned weights capture the specific predictive structure of the problem.

We use this idea in the specialization experiments of Sections 3.1, 3.2, and 3.4. The appropriate random object depends on the experiment: it is a set of directions for the single-index models and a set of three-dimensional subspaces for the multi-index model.

Shared post-hoc protocol. We first train each network and select its active neurons using the criterion reported for that experiment. We then keep both the learned weights and this active set fixed. For each trained seed and alignment threshold, we compare the observed percentage of active neurons exceeding the threshold with the corresponding percentages under $B = 5{,}000$ random sets of directions or subspaces. Thus, the control changes only the directions or subspaces against which the learned neurons are evaluated; it does not retrain or otherwise modify the network.

Model/weight	Active neurons (out of 2048)	Cosine $\geq .71$ (%)		Cosine $\geq .90$ (%)	
		True directions ↑	Random directions	True directions ↑	Random directions
ReLU	469 ± 57	56.75 ± 3.91	5.80 ± 3.96	33.22 ± 1.92	0.03 ± 0.24
GELU	$1{,}747 \pm 83$	30.15 ± 2.48	2.56 ± 2.05	18.40 ± 1.20	0.02 ± 0.12
ReGLU gate	200 ± 56	51.89 ± 1.22	5.58 ± 4.33	33.82 ± 1.27	0.05 ± 0.37
ReGLU value	200 ± 56	24.98 ± 0.52	1.08 ± 1.07	3.94 ± 2.24	0.00 ± 0.03
SwiGLU gate	432 ± 5	25.06 ± 2.29	3.30 ± 1.80	12.26 ± 1.27	0.02 ± 0.11
SwiGLU value	432 ± 5	17.88 ± 2.88	2.83 ± 1.19	4.62 ± 0.96	0.01 ± 0.04

Table 4: **Predictive-direction specialization exceeds a geometry-matched random-direction baseline.** Observed alignment with the true cluster-specific directions is substantially stronger than alignment with randomized orthonormal directions in the same predictive subspace. True-direction entries are means and sample standard deviations over three trained models. Random-direction entries summarize $15{,}000$ pooled comparisons ($5{,}000$ random bases for each trained model).

Within each training seed, the same random directions or subspaces are used for every architecture, so model comparisons are not affected by different random draws. The coordinate-model construction is given next; the non-axis-aligned single-index and multi-index constructions are given in Sections A.2.1 and A.4.3, respectively.

True-direction or true-span entries report means and sample standard deviations over three trained models. Random-baseline entries report the mean and sample standard deviation of all $15{,}000$ specialization percentages pooled across the three models and their $5{,}000$ random draws each. These are neither standard errors nor standard deviations of seed-level means. Specialization entries are percentages, with standard deviations in percentage points. Displayed values are rounded to two decimal places; a displayed zero may therefore represent a small nonzero value.

Application to the axis-aligned experiment. For the experiment in Section 3.1, we use the active-neuron criterion defined in Section A.1.2. The true predictive directions form an orthonormal coordinate basis of the ten-dimensional predictive subspace, so the randomized directions must preserve this joint geometry rather than being sampled independently. For each null draw, we sample a Haar-random orthogonal matrix $Q \in \mathbb{R}^{10\times 10}$ and embed its columns in the same ten predictive coordinates as the true directions. The randomized directions are therefore mutually orthonormal and span exactly the same predictive subspace as the true directions. This construction preserves each learned weight's norm and its total mass in the predictive subspace, while breaking its alignment with the particular direction assigned to each cluster.

We apply this randomization only to cosine-based specialization. The predictive-coordinate dominance measure depends on the chosen coordinate system and therefore does not have a rotation-invariant interpretation under this null.

A.1.4 NORMALIZED THIRD-HERMITE EXPERIMENT

Let He_3 denote the third probabilists' Hermite polynomial. We use its unit-variance normalization,

$$\mathrm{He}_3(t) = t^3 - 3t, \qquad h_3(t) = \frac{\mathrm{He}_3(t)}{\sqrt{3!}} = \frac{t^3 - 3t}{\sqrt{6}}.$$

For an example from cluster c, the response is

$$y = h_3\Big(\sqrt{d}\,\langle v_c, x\rangle\Big) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.005^2).$$

Under the data distribution in Section 3.1, $\langle v_c, x\rangle$ has conditional variance $1/d$. Thus, $\sqrt{d}\,\langle v_c, x\rangle$ has unit variance, and for $Z \sim \mathcal{N}(0,1)$, the normalization above gives $\mathbb{E}[h_3(Z)^2] = 1$.

Training protocol. All four architectures use width 2,048, a batch size of 2,048, trainable biases, the width-aware parameterization, and μP-scaled Adam (Kingma & Ba, 2015). Each run uses 100,000 training examples, 4,096 validation examples, and 2,048 test examples. For each architecture, we sweep over the learning rate and weight decay, selecting the configuration and checkpoint with the lowest validation MSE. Table 5 reports the selected configurations. The test set is never used for training, hyperparameter tuning, or checkpoint selection; it is used only for final evaluation.

Table 5: Training configurations for the normalized third-Hermite specialization experiment. Each learning rate follows a cosine schedule from the first value to the second.

Activation	Cosine LR	Weight decay
ReLU	$2 \times 10^{-3} \to 2 \times 10^{-4}$	10^{-6}
GELU	$1.6 \times 10^{-2} \to 1.6 \times 10^{-3}$	0
ReGLU	$1.6 \times 10^{-3} \to 1.6 \times 10^{-4}$	0
SwiGLU	$2 \times 10^{-3} \to 2 \times 10^{-4}$	0

The specialization statistics in Table 1 are reported as the mean and standard deviation over three independent seeds.

A.1.5 Normalized Second-Hermite Experiment

Figure 7 provides an additional experiment using a normalized second-Hermite link function. First-layer neurons remain strongly specialized across all four activation families. Thus, specialization also occurs beyond the third-Hermite setting analyzed in Section 4.1.

Experimental setting and link function. This experiment uses exactly the data geometry, input distribution, noise level, model architectures, and width from Section 3.1. The only change is the link function: in every cluster, we replace the normalized third-Hermite link function by the unit-variance normalized second-Hermite link function

$$h_2(t) = \frac{\mathrm{He}_2(t)}{\sqrt{2!}} = \frac{t^2 - 1}{\sqrt{2}}, \qquad \mathrm{He}_2(t) = t^2 - 1.$$

Thus, for an example from cluster c,

$$y = h_2\Big(\sqrt{d}\,\langle v_c, x\rangle\Big) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.005^2).$$

For $Z \sim \mathcal{N}(0,1)$, this normalization gives $\mathbb{E}[h_2(Z)^2] = 1$, matching the scale of the normalized third-Hermite link function. We generate the training, validation, and test sets independently; they contain 100,000, 4,096, and 2,048 examples, respectively.

		Specialization measures (% of active neurons)			
Model/weight	**Active neurons (out of 2048)**	$\mathbf{A_j \geq .71}$ ↑	$\mathbf{A_j \geq .90}$ ↑	$\mathbf{D_j > 1}$ ↑	$\mathbf{D_j > 2}$ ↑
ReLU	613	45.5	33.4	71.1	49.1
GELU	1,710	49.9	28.2	68.2	43.1
ReGLU gate	453	73.1	56.1	96.7	73.5
ReGLU value	453	28.5	3.8	69.1	24.7
SwiGLU gate	551	43.6	19.1	91.7	38.1
SwiGLU value	551	46.3	14.9	87.5	41.4

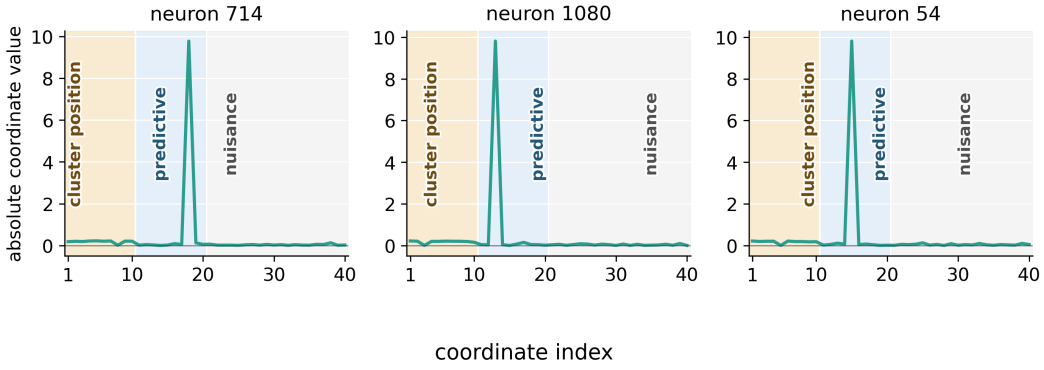


Figure 7: **Strong neuron specialization also occurs for the normalized second-Hermite link function.** Top: percentage of specialized neurons among active neurons. Across standard and gated activations, many neurons align strongly with a single cluster-specific predictive direction. Bottom: absolute first-layer weights of selected GELU neurons. Each neuron concentrates on one predictive coordinate, with comparatively little weight on the cluster-position and nuisance coordinates.

Training protocol. All models use μP-scaled Adam, width 2048, batch size 2048, zero weight decay, and the width-aware parameterization of Section A.1.4. The cosine learning-rate schedules are

$$\text{ReLU: } 4 \times 10^{-3} \to 4 \times 10^{-4},$$
$$\text{GELU: } 1.6 \times 10^{-2} \to 1.6 \times 10^{-3},$$
$$\text{ReGLU: } 1.6 \times 10^{-3} \to 1.6 \times 10^{-4},$$
$$\text{SwiGLU: } 2.56 \times 10^{-2} \to 2.56 \times 10^{-3}.$$

Each reported model uses its lowest-validation-MSE checkpoint. Test data are used only for final evaluation.

A.2 Additional Details for MLPs Jointly Learning Cluster Structure and Cluster-Specific Predictive Functions

This appendix provides the experimental details and additional results supporting Section 3.2. Section A.2.1 details the specialization measure and random-direction control used in Table 2. Section A.2.2 evaluates whether active neurons retain predictive performance after pruning, comparing fine-tuning the active neurons with updating only the readout while keeping the hidden parameters fixed. Section A.2.3 gives the complete protocol for the single-index experiment with mixed link functions and extends the comparison in Fig. 2 to all four MLP activations—ReLU, GELU, ReGLU, and SwiGLU—by adding GELU and SwiGLU (Fig. 8). Finally, Section A.2.4 presents the experiment in Fig. 9, which uses only the normalized third-order Hermite link function, as in our theoretical analysis in Section 4.1.

A.2.1 Neuron Specialization and Random-Direction Control

Table 2 complements the predictive-performance results in Fig. 2 by testing whether first-layer neurons align with the true cluster-specific predictive directions. The analysis uses the $K = 10$, $n = 500{,}000$ models and the active-neuron criterion defined in Section A.1.2. For gated architectures, the same active set is used to analyze gate and value weights separately.

Specialization measure. For each active first-layer weight w_j, we compute its maximum absolute cosine with the ten true predictive directions,

$$\max_{1 \le c \le K} \frac{|\langle w_j, v_c \rangle|}{\|w_j\|_2 \, \|v_c\|_2}.$$

The observed columns in Table 2 report the percentage of active neurons whose maximum cosine is at least 0.71 or 0.90. The denominator uses the full twenty-dimensional weight norm.

Random-direction control. We use the shared post-hoc protocol of Section A.1.3. Unlike the coordinate experiment in Section 3.1, the true directions here are independently sampled, are not axis-aligned, and are not constrained to be mutually orthogonal. We therefore sample random directions from the same distribution as the true directions. For each trained model, we generate $B = 5{,}000$ independent sets of random directions, indexed by $b \in \{1, \dots, B\}$. Within each set b, we independently sample one direction for each cluster c:

$$\widetilde{q}_c^{(b)} \sim \mathrm{Unif}(\mathbb{S}^9), \qquad q_c^{(b)} = \begin{bmatrix} 0 \\ \widetilde{q}_c^{(b)} \end{bmatrix} \in \mathbb{R}^{20}.$$

For every active weight, we replace the maximum cosine with the true directions by the maximum cosine with $q_1^{(b)}, \dots, q_K^{(b)}$, and then recompute the percentage exceeding each threshold. This experiment-specific construction preserves the sampling distribution of the predictive directions in addition to the properties preserved by the shared protocol.

The random-direction entries in Table 2 report the pooled mean and sample standard deviation defined in Section A.1.3, across all $15{,}000$ specialization percentages ($5{,}000$ random direction sets for each of the three trained models). At both thresholds, alignment with the true directions is substantially stronger than alignment with the randomized directions for every architecture and weight type.

A.2.2 Importance-Based Pruning and Fine-Tuning

The active-neuron criterion in Section A.1.2 is based on weight importance. We evaluate the selected neurons' predictive role through two pruning experiments: fine-tuning only the active neurons re-

covers approximately the full network's test MSE with similar neuron specialization. In the second experiment, we freeze both the first-layer weights and biases and fine-tune only the readout. This recovers much of the performance lost through pruning without changing the retained neurons' directional alignment. In both experiments, fine-tuning uses exactly the same training data as the original model, and evaluation uses the same held-out test data.

Fine-tuning the retained neurons. We retain and fine-tune only the active neurons. Table 6 shows that fine-tuning recovers similar or better test MSE than the full model, with similar neuron specialization (Table 7).

		Test MSE ($\times 10^{-3}$) ↓		
Model	**Active neurons (out of 4096)**	**Full model**	**Pruned**	**Fine-tuned**
ReLU	558 ± 36	1.97 ± 0.30	2.09 ± 0.29	1.94 ± 0.29
GELU	287 ± 15	1.42 ± 0.30	3.93 ± 0.75	1.38 ± 0.32
ReGLU	133 ± 44	1.54 ± 0.31	2.02 ± 0.19	1.49 ± 0.38
SwiGLU	248 ± 6	0.74 ± 0.13	0.92 ± 0.09	0.73 ± 0.14

Table 6: **Fine-tuning active neurons recovers performance after pruning.** Removing the remaining neurons increases test error, but fine-tuning all retained parameters restores approximately the full model's performance. Results use the same $K = 10$, $n = 500{,}000$ models as Table 2. Entries report mean ± sample standard deviation over the same three runs.

	Cosine ≥ 0.71 (%)		Cosine ≥ 0.90 (%)	
Model/weight	**Before**	**After**	**Before**	**After**
ReLU	12.72 ± 3.54	12.18 ± 3.57	3.02 ± 0.79	3.14 ± 0.73
GELU	16.87 ± 0.79	17.21 ± 0.79	8.01 ± 0.66	8.01 ± 0.66
ReGLU gate	16.87 ± 4.31	16.87 ± 4.31	5.02 ± 1.31	4.99 ± 1.25
ReGLU value	79.65 ± 4.15	79.65 ± 4.15	60.12 ± 6.44	59.92 ± 5.63
SwiGLU gate	5.39 ± 1.15	5.39 ± 0.93	2.00 ± 0.77	1.87 ± 0.98
SwiGLU value	16.14 ± 0.79	16.14 ± 0.79	7.16 ± 2.43	7.30 ± 2.67

Table 7: **Active neurons retain similar specialization after fine-tuning.** Percentages are measured among the same active neurons before and after fine-tuning all retained parameters, using alignment with the original predictive directions. Entries report mean ± sample standard deviation over three runs. The before-fine-tuning values and retained-neuron counts match Table 2.

Readout-only fine-tuning after pruning. To isolate whether the active neurons already provide useful predictive features, we again retain only the active neurons, freeze both their first-layer weights and biases, and update only the output weights and output bias. For ReGLU and SwiGLU, both the gate and value branches are frozen. Table 8 shows that readout adjustment recovers much of the performance lost through pruning.

Model	Active neurons (out of 4096)	Test MSE ($\times 10^{-3}$) ↓		
		Full model	Pruned	Readout-only fine-tuning
ReLU	558 ± 36	1.97 ± 0.30	2.09 ± 0.29	1.94 ± 0.27
GELU	287 ± 15	1.42 ± 0.30	3.93 ± 0.75	1.53 ± 0.31
ReGLU	133 ± 44	1.54 ± 0.31	2.02 ± 0.19	1.57 ± 0.28
SwiGLU	248 ± 6	0.74 ± 0.13	0.92 ± 0.09	0.79 ± 0.11

Table 8: **Readout adjustment recovers much of the performance lost through pruning.** Updating only the output weights and output bias reduces test error while keeping all retained hidden parameters fixed. Results use the same $K = 10$, $n = 500{,}000$ models and active-neuron sets as Table 6. Entries report mean $\pm$ sample standard deviation over the same three data and model-initialization runs.

Shared experimental setting and pruning. Both experiments use the same trained models as Table 2: ReLU, GELU, ReGLU, and SwiGLU networks of width 4096, with $d = 20$, $K = 10$, and $n = 500{,}000$. The data and original training protocols are given in Section A.2.3.

In both experiments, we retain the active neurons defined in Section A.1.2, using the same set as the specialization analysis in Section A.2.1. This is the smallest importance-ranked set carrying at least 99.9% of total importance. For gated architectures, each retained unit includes both its gate and value branches. Pruning removes all other neurons while preserving the retained parameters and the original width-dependent output scaling. The selected neuron identities remain fixed during fine-tuning; we do not reapply the importance criterion afterward. Both fine-tuning procedures start independently from the same pruned model.

Shared fine-tuning protocol. Both experiments use the original $500{,}000$ training examples. A fresh Adam optimizer runs for $20{,}000$ steps with minibatches of 4096 examples sampled with replacement. Cosine decay reduces the learning rate from 10^{-3} to 10^{-5} for ReLU, GELU, and SwiGLU, and from 10^{-4} to 10^{-6} for ReGLU. Weight decay is unchanged from the original models: 10^{-6} for ReLU and GELU, 10^{-5} for ReGLU, and zero for SwiGLU, applied only to trainable parameters. For the readout-only control, Adam uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We save checkpoints and evaluate full-training MSE every 1000 steps, including the pruned model at step zero. We select the checkpoint with the lowest training MSE. Neither validation nor test data are used for fine-tuning or checkpoint selection. Evaluation uses the original 4096 test examples per seed; results summarize the same three runs.

Specialization evaluation. We use the alignment measure in Section A.2.1, with the original predictive directions and the full 20-dimensional weight norm. The thresholds are 0.71 and 0.90, as in Table 2. Before and after fine-tuning, we evaluate the same retained neurons, with gate and value weights evaluated separately. Pruning itself leaves their alignment unchanged because it preserves their incoming weights.

A.2.3 Experimental Details for the Single-Index Experiment with Mixed Link Functions

Cluster and predictive geometry. We use $d = 20$, with the first ten coordinates containing cluster-identifying information and the final ten coordinates containing the predictive signal. We consider

$$K \in \{1, 2, 10, 50\}.$$

The finite datasets contain equal numbers of examples from each cluster whenever the sample size is divisible by K, and otherwise differ by at most one example per cluster.

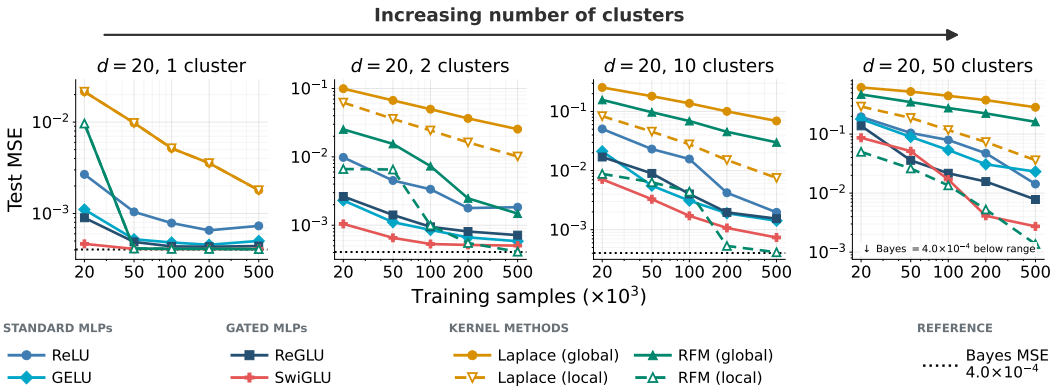


Figure 8: **Additional activations for the single-index model with mixed link functions.** Same plot as Figure 2, with GELU and SwiGLU added. As the number of clusters grows, MLPs, especially those with gated activations such as ReGLU and SwiGLU, maintain strong performance close to local Laplace and RFM, while global RFM becomes increasingly less sample efficient. Curves show loss averaged over three runs with different link-function assignments, data, label noise, model initialization, and optimization minibatches.

For each K, cluster-center directions are obtained by farthest-point sampling from $\max\{20{,}000, 500K\}$ independent Gaussian vectors normalized to unit norm, giving

$$s_1, \dots, s_K \in \mathbb{S}^9, \qquad \mathbb{S}^9 = \{s \in \mathbb{R}^{10} : \|s\|_2 = 1\}.$$

For $K > 1$, define

$$\rho_K = \max_{c \neq c'} s_c^\top s_{c'}$$

and set

$$\mu_c = \begin{bmatrix} s_c \\ 0 \end{bmatrix} \in \mathbb{R}^{20}.$$

For $K = 1$, we set $\mu_1 = 0$, since no cluster discrimination is required.

The standard deviation of the noise in the first ten coordinates is

$$\sigma_K = \frac{1 - \rho_K}{2\gamma}, \qquad \gamma = 4, \tag{1}$$

where we take $\rho_1 = 0$. Thus, the cluster noise adapts to the least-separated pair of centers and keeps the clusters well separated as K changes.

Each cluster independently receives a unit predictive direction

$$\widetilde{v}_c \sim \mathrm{Unif}(\mathbb{S}^9), \qquad v_c = \begin{bmatrix} 0 \\ \widetilde{v}_c \end{bmatrix} \in \mathbb{R}^{20}.$$

The cluster-identifying and predictive coordinates are therefore disjoint.

Input distribution. Given cluster c, an input is sampled as

$$x = \mu_c + \begin{bmatrix} \sigma_K z_{\text{cluster}} \\ z_{\text{pred}} \end{bmatrix}, \qquad z_{\text{cluster}}, z_{\text{pred}} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{10}). \tag{2}$$

Equivalently,

$$x \mid C = c \sim \mathcal{N}(\mu_c, \Sigma_c), \qquad \Sigma_c = \begin{pmatrix} \sigma_K^2 I_{10} & 0 \\ 0 & I_{10} \end{pmatrix}.$$

Table 9: Number of clusters assigned to each link function for the three runs shown in Fig. 8.

K	Seed	Hermite-2	Hermite-3	Sine	Tanh
1	1000	0	0	0	1
1	1001	0	1	0	0
1	1002	0	0	0	1
2	1000	0	0	0	2
2	1001	0	2	0	0
2	1002	0	1	0	1
10	1000	3	2	1	4
10	1001	2	3	3	2
10	1002	5	1	2	2
50	1000	17	13	12	8
50	1001	14	17	12	7
50	1002	11	14	14	11

Cluster-specific link functions. Each cluster independently receives a link function sampled uniformly with replacement from

$$\mathcal{G} = \left\{ t \mapsto \frac{t^2 - 1}{\sqrt{2}}, \quad t \mapsto \frac{t^3 - 3t}{\sqrt{6}}, \quad t \mapsto \sin(t), \quad t \mapsto \tanh(t) \right\}. \tag{3}$$

The first two functions are the normalized probabilists' Hermite polynomials of degrees two and three. Sampling with replacement allows several clusters to receive the same link function.

For an example from cluster c, the response is

$$y = g_c(\langle x, v_c \rangle) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.02^2). \tag{4}$$

Because v_c is supported only on the predictive coordinates,

$$\langle x, v_c \rangle = \widetilde{v}_c^\top z_{\text{pred}}.$$

Thus, both the predictive direction and the nonlinear link can change across clusters.

The realized link assignments for the three displayed seeds are reported in Table 9. For a fixed (K, seed), these assignments are shared across all methods and training sizes.

Training, validation, and test data. For each (K, seed), we train on datasets of sizes 20,000, 50,000, 100,000, 200,000, and 500,000. For each (K, n, seed), all methods use exactly the same training, validation, and test datasets. The validation and test sets are independent of the training data, and the test set contains 4096 examples. Validation data are used for model selection; test data are used only for final evaluation.

MLP architectures. We evaluate one-hidden-layer ReLU, GELU, ReGLU, and SwiGLU networks of width 4,096. ReLU and GELU use

$$h(x) = \phi(Wx + b),$$

while the gated models use

$$h(x) = \phi(W_g x + b_g) \odot (W_v x + b_v).$$

For ReGLU, $\phi = \mathrm{ReLU}$; for SwiGLU, $\phi = \mathrm{SiLU}$. All hidden weights, readout weights, and biases are trainable.

Hidden-weight entries are initialized independently with standard deviation $10^{-3}/\sqrt{d}$, and all biases are initialized to zero. For ordinary MLPs, the initial readout magnitude of neuron j is proportional to $\|w_j\|_2$, with balanced random signs. For gated MLPs, it is proportional to

$$\sqrt{\|w_{g,j}\|_2 \|w_{v,j}\|_2}.$$

The output multiplier is $1024/4096$.

Table 10: MLP hyperparameters used in Fig. 8. Each entry is (η_0, λ), denoting initial learning rate and weight decay.

Condition	ReLU	GELU	ReGLU	SwiGLU
$K = 1$, all n	$(10^{-2}, 10^{-5})$	$(10^{-2}, 10^{-6})$	$(10^{-3}, 10^{-5})$	$(3{\times}10^{-2}, 10^{-6})$
$K = 2$, all n	$(3.33{\times}10^{-3}, 10^{-5})$	$(10^{-2}, 10^{-6})$	$(10^{-3}, 10^{-5})$	$(3.33{\times}10^{-3}, 10^{-6})$
$K = 10,\ n = 20\text{k}$	$(3{\times}10^{-2}, 10^{-5})$	$(10^{-2}, 10^{-6})$	$(10^{-3}, 10^{-5})$	$(10^{-2}, 10^{-6})$
$K = 10,\ n \in \{50\text{k}, 100\text{k}\}$	$(3{\times}10^{-2}, 10^{-5})$	$(10^{-2}, 10^{-6})$	$(10^{-3}, 10^{-5})$	$(10^{-2}, 0)$
$K = 10,\ n \in \{200\text{k}, 500\text{k}\}$	$(10^{-2}, 10^{-6})$	$(10^{-2}, 10^{-6})$	$(10^{-3}, 10^{-5})$	$(10^{-2}, 0)$
$K = 50,\ n = 20\text{k}$	$(3{\times}10^{-2}, 0)$	$(3{\times}10^{-2}, 10^{-6})$	$(9{\times}10^{-3}, 10^{-6})$	$(3{\times}10^{-2}, 10^{-6})$
$K = 50,\ n = 50\text{k}$	$(3{\times}10^{-2}, 0)$	$(3{\times}10^{-2}, 10^{-6})$	$(9{\times}10^{-3}, 10^{-6})$	$(10^{-2}, 0)$
$K = 50,\ n = 100\text{k}$	$(3{\times}10^{-2}, 10^{-6})$	$(3{\times}10^{-2}, 10^{-6})$	$(9{\times}10^{-3}, 10^{-6})$	$(10^{-2}, 0)$
$K = 50,\ n \in \{200\text{k}, 500\text{k}\}$	$(10^{-2}, 0)$	$(3{\times}10^{-2}, 10^{-6})$	$(9{\times}10^{-3}, 10^{-6})$	$(10^{-2}, 0)$

MLP optimization and evaluation. All MLPs use Adam, batch size 4096, and cosine learning-rate decay with horizon 200,000 updates and final rate $\eta_0/100$. Table 10 gives the learning rates and weight decay used in each condition.

Global Laplace and RFM. We follow the RFM framework of Radhakrishnan et al. (2024) and its accompanying implementation[1] for Laplace-kernel fitting, computation of the average gradient outer product $\widehat{G}_t$, and updates to the feature metric M_t. We initialize $M_0 = I_{20}$, so iteration zero is the isotropic Laplace baseline.

Table 11 gives the configurations used across sample sizes and seeds. Each entry is (b, t, λ), where b is the bandwidth multiplier, t is the RFM iteration, and λ is the ridge parameter.

Table 11: Kernel configurations used in Fig. 8. Each tuple is (b, t, λ): bandwidth multiplier b, RFM iteration t (with $t = 0$ denoting the Laplace baseline), and ridge parameter λ.

Condition	Global Laplace	Global RFM	Local Laplace	Local RFM
$K = 1$, all n	$(2, 0, 10^{-6})$	$(2, 3, 10^{-6})$	same as global	same as global
$K = 2$, all n	$(2, 0, 10^{-6})$	$(1, 1, 10^{-6})$	$(2, 0, 10^{-8})$	$(2, 3, 10^{-8})$
$K = 10$, all n	$(2, 0, 10^{-6})$	$(2, 1, 10^{-6})$	$(2, 0, 10^{-8})$	$(2, 3, 10^{-6})$
$K = 50$, all n	$(2, 0, 10^{-6})$	$(2, 2, 10^{-6})$	$(2, 0, 10^{-8})$	$(2, 3, 10^{-8})$

Exact kernel solves are used when an individual kernel problem contains fewer than 11,000 training examples. Larger kernel problems are solved using EigenPro (Ma et al., 2018; Ma & Belkin, 2019; Abedsoltan et al., 2023; 2025). EigenPro uses a training-MSE target of 4×10^{-4}, equal to the observation-noise variance. The explicit ridge parameter applies directly to exact solves; EigenPro is regularized primarily through early stopping.

Local methods. The local Laplace and RFM methods receive the true cluster identity during both training and testing. They fit an independent predictor using the training examples within each cluster and evaluate each test example using the predictor associated with its true cluster.

Iteration zero of each local RFM is the local Laplace predictor. Each local RFM then performs at most three metric updates. For $K = 1$, the global and local methods coincide.

These local methods are diagnostic references rather than fair predictive baselines because the cluster identity is supplied directly. Their purpose is to measure the difficulty of learning the local predictive functions once the cluster assignment is known.

Bayes reference. The dotted line is the empirical test MSE of the Bayes predictor under the known data-generating distribution. It computes the posterior probability of each cluster given x and av-

[1] https://github.com/aradha/recursive_feature_machines

erages the corresponding noiseless local predictions. The irreducible observation-noise variance is

$$0.02^2 = 4 \times 10^{-4}.$$

Across the test sets in Fig. 8, the empirical Bayes MSE lies between approximately 3.97×10^{-4} and 4.14×10^{-4}, consistent with finite-test-set variation around the noise variance.

A.2.4 A Common Third-Hermite Link Function

Fixed third-Hermite single-index model. We use the clustered single-index geometry of Section A.2.3, with $d = 20$,

$$K \in \{1, 2, 10, 50\}, \qquad n \in \{20{,}000, 50{,}000, 100{,}000, 200{,}000\}.$$

Each cluster has an independently sampled predictive direction, but every cluster uses the common normalized third-Hermite link

$$g_c(t) = h_3(t) = \frac{t^3 - 3t}{\sqrt{6}}, \qquad c \in [K].$$

A common link function isolates the role of cluster-dependent predictive directions and supports the third-Hermite analysis in Section 4.1.

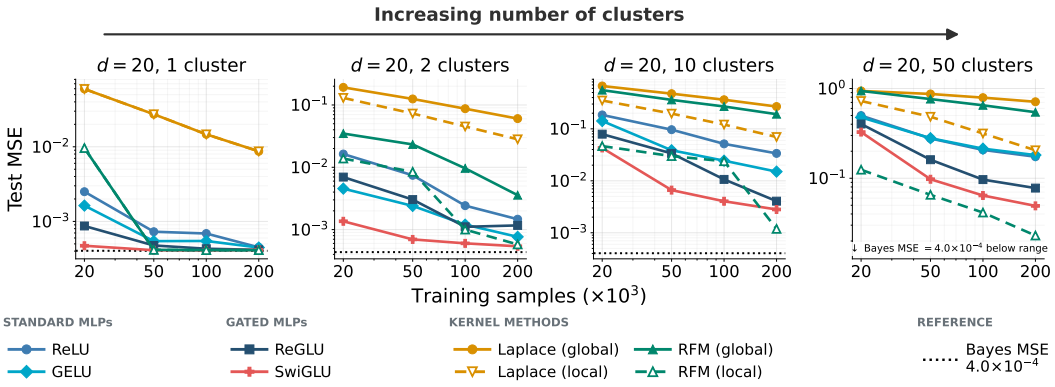


Figure 9: **Fixed third-Hermite single-index model.** All clusters use the normalized third-order Hermite link function with the clustered input geometry of Figure 2, providing a setting closer to the theory in Section 4.

Results. Figure 9 shows that the separation between the global methods and the MLPs grows with the number of clusters. Global Laplace and RFM become progressively less effective as the cluster-specific directions occupy more of the predictive subspace. The MLPs continue to exploit the cluster-dependent directions, and the gated architectures perform especially well in the many-cluster regime.

A.3 Additional Details for Trained MLPs Encoding Cluster Structure

This appendix supplements the experiment in Section 3.3, which builds on the clustered single-index setting of Section 3.2. We describe how cluster gates are extracted from a trained ReLU MLP and reused to construct independent local Laplace and RFM predictors. Figure 10 summarizes the procedure.

Clustered data with mixed link functions. We use the $d = 20$ input distribution in Eq. 2, with ten cluster-identifying and ten predictive coordinates, cluster-center radius $R = 1$, and $K \in \{2, 5, 10, 50\}$ equally represented clusters. Cluster-center and predictive directions follow Section A.2.3, with the cluster-noise rule in Eq. 1 and $\gamma = 4$. Each cluster's link function is sampled independently and uniformly from normalized Hermite-3, sine, and hyperbolic tangent. Responses follow Eq. 4, with observation-noise standard deviation 0.02.

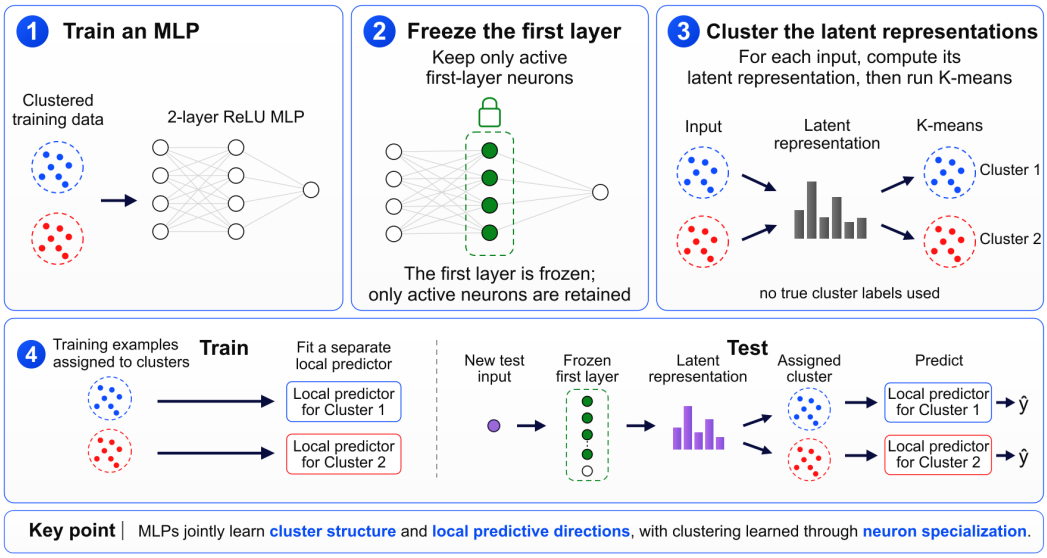


Figure 10: **Overview of the MLP-gated local-prediction procedure.** A ReLU MLP is trained without cluster labels, K-means is applied to its first-layer representations, and a local predictor is fit within each learned cluster. Test inputs are routed using the nearest learned centroid.

Datasets and repetitions. For every K, we generate a balanced training pool containing 200,000 examples and use nested prefixes of sizes

$$n \in \{20{,}000, 50{,}000, 100{,}000, 200{,}000\}.$$

Evaluation uses an independently generated test set containing 10,000 examples. Within each run, the same training and test data are used by the MLP, global kernel methods, ground-truth local methods, and MLP-gated methods.

Results are averaged over three seeds. For a fixed K, the cluster-position and predictive directions are held fixed, while each seed independently resamples the cluster-specific link-function assignments, training and test examples, label noise, and MLP initialization. No test labels are used to construct the gates.

Source ReLU MLP. For each K, sample size, and seed, we train a one-hidden-layer ReLU network of width 4,096,

$$\widehat{f}(x) = \sum_{j=1}^{4096} a_j \,\mathrm{ReLU}(w_j^\top x + b_j) + b_{\mathrm{out}}.$$

All weights and biases are trainable. Optimization uses Adam with minibatches of size 4,096, zero weight decay, and cosine learning-rate decay from 10^{-2} to 10^{-4}. Training runs for at least 20,000 steps and at most 200,000 steps. It terminates when the full training MSE reaches 4×10^{-4}, or when the training loss plateaus. We restore the checkpoint with the smallest full training MSE. Neither test MSE nor cluster identity is used for checkpoint selection.

Selecting active neurons. After training, we freeze the ReLU MLP and select the active-neuron set $\mathcal{J}$ using the shared importance-based criterion in Section A.1.2. Here, these neurons are used to construct contribution profiles for clustering inputs and gate-specific features for the local predictors, rather than to measure alignment with the predictive directions.

Contribution profiles and learned gates. For every training input x, we compute its normalized contribution profile over the active neurons:

$$p_j(x) = \frac{|a_j|\,\mathrm{ReLU}(w_j^\top x + b_j)}{\sum_{\ell \in \mathcal{J}} |a_\ell|\,\mathrm{ReLU}(w_\ell^\top x + b_\ell)}, \qquad j \in \mathcal{J}. \tag{5}$$

Thus $p(x)$ describes which active first-layer neurons contribute to the MLP prediction on input x, independent of the overall magnitude of the prediction.

We apply K-means++ to the training contribution profiles and use $M = K$ learned gates. Knowledge of the number of clusters is therefore provided, but the true cluster assignments are never observed. We use eight initializations and at most 30 Lloyd iterations, retaining the solution with the smallest training inertia. Every training example is assigned to exactly one gate through its nearest centroid.

Let q_c denote the centroid of learned gate c, with coordinate q_{cj} corresponding to active neuron j. Although examples receive hard gate assignments, the centroid coordinates provide a soft association between neurons and gates: the same neuron may contribute to several gates.

Constructing gate-specific features. For each learned gate c, we form the positive-semidefinite metric

$$G_c = \frac{\sum_{j \in \mathcal{J}} q_{cj} w_j w_j^\top}{\sum_{j \in \mathcal{J}} q_{cj}}. \tag{6}$$

The corresponding gate-specific representation is

$$z_c(x) = G_c^{1/2} x. \tag{7}$$

The MLP biases affect the contribution profiles and hence the gate assignments, while G_c itself is constructed from the incoming first-layer weight directions.

For every learned gate, we project only the training examples assigned to that gate and fit an independent local Laplace or RFM predictor in the resulting representation.

Test-time prediction. For a new test input x, we:

1. pass x through the frozen MLP and compute its contribution profile $p(x)$;
2. assign x to the nearest fixed training centroid;
3. transform it using the corresponding representation $z_c(x) = G_c^{1/2} x$; and
4. evaluate the local predictor fitted for that learned gate.

The test input is therefore assigned using only its MLP contribution profile. Neither its response nor its true cluster identity is used.

Local Laplace and RFM predictors. We use the Laplace-kernel fitting, RFM metric updates, and kernel solvers described in Section A.2.3, applied within each learned gate to the representation $z_c(x) = G_c^{1/2} x$. Here, the ridge parameter is 10^{-6}, and the bandwidth is the median pairwise distance within the gate, recomputed after each metric update. RFM performs three metric updates, with iteration zero providing the local Laplace baseline. The same fitting procedure is used for the global and ground-truth local RFM baselines.

Compared methods. Figure 3 compares:

- the source ReLU MLP;
- one global Laplace predictor fitted to all training examples;
- one global RFM fitted to all training examples;
- MLP-gated local Laplace, using the gates and projections extracted from the ReLU MLP;
- MLP-gated local RFM, using the same extracted gates and projections;
- local Laplace, which is given the true cluster identities at training and test time;
- local RFM, defined analogously; and
- the irreducible noise level, whose expected MSE is $0.02^2 = 4 \times 10^{-4}$.

The local methods using the true clusters are diagnostic references rather than fair deployable predictors because they receive the true cluster identity. By contrast, the MLP-gated methods receive only the training input–response pairs and the known number K of gates.

Aggregation. Every curve in Figure 3 is the arithmetic mean over three independent runs. Each run uses a newly sampled dataset from the same generative model and an independently initialized MLP.

A.4 Additional Details for Clustered Multi-Index Models

This appendix supplements Section 3.4. Section A.4.1 specifies the data distribution. Section A.4.2 reports the models and training settings, and Section A.4.3 extends the specialization analysis to three-dimensional predictive spans.

A.4.1 Data Generation

Cluster geometry. We follow the Gaussian-mixture construction in Section A.2.3, increasing the input dimension to $d = 50$, with coordinates $1{:}20$ identifying the cluster, $21{:}40$ containing the predictive variables, and $41{:}50$ containing nuisance variables. We consider

$$K \in \{1, 2, 10, 25\}, \qquad n \in \{20{,}000, 50{,}000, 100{,}000, 200{,}000, 500{,}000\}.$$

Clusters have equal sampling probabilities, and training sets contain exactly n/K examples from each cluster.

For $K > 1$, cluster centers are

$$\mu_c = (s_c, 0_{20}, 0_{10}), \qquad s_c \in \mathbb{R}^{20}, \qquad \|s_c\|_2 = 1.$$

The vectors s_c are selected from $20{,}000$ random unit-vector candidates in $\mathbb{R}^{20}$ using the farthest-point procedure of Section A.2.3. We use the same cluster-noise rule in Eq. 1, with $\gamma = 4$, and the same $K = 1$ convention. With the enlarged blocks and additional nuisance variables, the conditional input distribution becomes

$$x \mid c \sim \mathcal{N}\big(\mu_c,\ \mathrm{diag}(\sigma_K^2 I_{20}, I_{20}, I_{10})\big)\,.$$

Predictive spans and responses. Independently for each cluster, we generate a 20×3 matrix with independent standard Gaussian entries and compute its thin QR decomposition. If $q_{c,1}, q_{c,2}, q_{c,3}$ are the resulting orthonormal columns, we set

$$v_{c,r} = (0_{20}, q_{c,r}, 0_{10}), \qquad r = 1, 2, 3.$$

Directions are orthonormal within a cluster; frames from different clusters are neither shared nor constrained to be orthogonal.

Each cluster independently receives one link function, sampled uniformly with replacement from the normalized second-order Hermite polynomial, sine, and hyperbolic tangent. Responses follow the three-index model in Section 2, with the same observation-noise standard deviation 0.02 as in Section A.2.3. Each predictive projection has unit variance.

Shared data and repeated runs. For each K, cluster centers and predictive frames are held fixed across three runs. Across runs, we vary the link-function assignments, sampled inputs, observation noise, network initialization, and optimization minibatches.

All training sets are nested, cluster-balanced subsets of a single $500{,}000$-example pool for each K and run. Every run uses independently sampled validation and test sets of sizes $20{,}000$ and $4{,}096$, respectively. All methods share the same training, validation, and test data for each (K, n) and run. Validation data are used for model selection, and test data only for final evaluation. We report the mean test MSE over three runs without standard-deviation bands.

A.4.2 Models and Training

MLP architecture and initialization. We use the ReLU, GELU, ReGLU, and SwiGLU architectures described in Section A.2.3, with width 2048 and input dimension $d = 50$. Hidden and output biases are trainable. Incoming weights are initialized with standard deviation $10^{-3}/\sqrt{50}$, and biases are initialized to zero. Readout magnitudes follow the norm-based formulas in Section A.2.3, with proportionality constant one and independent random signs. The output multiplier is $1024/2048$.

Optimization and checkpoint selection. We use Adam with batch size 4096 and cosine learning-rate decay. For each architecture, we sweep over the initial learning rate and weight decay, selecting the configuration and checkpoint with the lowest validation MSE.

Kernel methods. We use the global and local Laplace and RFM predictors and kernel-fitting procedures described in Section A.2.3. We select the RFM iteration by validation MSE, including iteration zero as the Laplace baseline, and evaluate the selected predictors on the held-out test set.

A.4.3 Specialization Analysis and Random-Subspace Control

Active neurons. We analyze the validation-selected models at $K = 10$ and $n = 500{,}000$, using the active-neuron criterion in Section A.1.2, as in Section A.2.1. This includes the same 99.9% importance threshold and separate gate/value analyses on the same active set.

Span-specialization measure. We extend the direction-alignment measure in Section A.2.1 to alignment with an entire cluster-specific predictive span. The orthogonal projector onto cluster c's predictive span is

$$P_c = \sum_{r=1}^{3} v_{c,r} v_{c,r}^\top.$$

For an incoming weight w_j, define

$$S_j = \max_{1 \le c \le K} \frac{\|P_c w_j\|_2}{\|w_j\|_2}. \tag{8}$$

The denominator includes all fifty coordinates, including cluster-identifying and nuisance coordinates. We report the percentage of active neurons satisfying $S_j \ge 0.71$ or $S_j \ge 0.90$. True-span entries and active-neuron counts report mean and sample standard deviation over three trained models. This measure detects alignment with a cluster's predictive span, not necessarily with an individual direction within it.

Random-subspace control. We use the shared protocol in Section A.1.3, replacing the random directions of Section A.2.1 with random three-dimensional subspaces. For each of the $B = 5{,}000$ draws per trained model, we independently generate ten such subspaces within the twenty-dimensional predictive subspace, using the Gaussian QR construction in Section A.4.1. We then substitute their projectors for the true projectors in Eq. 8 and recompute the percentages at the same thresholds.

The random-subspace entries in Table 3 report the pooled mean and sample standard deviation defined in Section A.1.3, across all $15{,}000$ specialization percentages ($5{,}000$ random subspace collections for each of the three trained models). The same random subspace collections are used across activations and weight types within each seed.

B Proofs for Section 4

B.1 Proof of Theorem 1

We first prove Theorem 1 invoking Lemmas 3 to 9, whose statements and proofs are provided after the main proof. A roadmap of their dependencies is shown in fig. 11.

Throughout this subsection, the normalized means and predictive directions

$$\frac{\mu_1}{R}, \dots, \frac{\mu_K}{R}, v_1, \dots, v_K$$

form an orthonormal basis of $\mathbb{R}^d$, where $d = 2K$. In the notation of Section 2, we may write

$$c \sim \mathrm{Unif}([K]), \qquad x = \mu_c + z, \qquad z \sim \mathcal{N}(0, I_d),$$

and

$$y = h_3\big(v_c^\top z\big), \qquad h_3(t) = \frac{t^3 - 3t}{\sqrt{6}}.$$

For $\omega \in \mathbb{S}^{d-1}$, define the weight's correlations with the cluster means and predictive directions

$$b_c(\omega) = \langle \omega, \mu_c \rangle, \qquad \rho_c(\omega) = \langle \omega, v_c \rangle.$$

Write the ReLU activation function and the Gaussian density as

$$\phi(t) = t_+, \qquad \varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2},$$

and define the unsigned and signed population-correlation objectives, respectively, by

$$\Phi(\omega) = \mathbb{E}\left[y\, \phi(\omega^\top x)\right], \qquad \Psi_\zeta(\omega) = \zeta \Phi(\omega), \qquad \zeta \in \{\pm 1\}.$$

Proof of Theorem 1. For each initial direction ω_j^0, let $(\bar{u}_j, \bar{\omega}_j)$ be the corresponding self-selected teacher-only trajectory from Lemma 7.

By Lemmas 7 and 5, almost surely there are a cluster label $J_j \in [K]$ and an orientation $\tau_j \in \{\pm 1\}$ such that

$$\limsup_{t \to \infty} \left\| \bar{\omega}_j(t) - \tau_j v_{J_j} \right\|_2 \le \frac{C}{R}$$

for a universal constant C.

Since $m < \infty$ and every teacher-only direction converges, for every $\delta > 0$ there is a finite time T_δ such that, simultaneously for all $j \in [m]$,

$$\left\| \bar{\omega}_j(T_\delta) - \tau_j v_{J_j} \right\|_2 \le \frac{\delta}{2} + \frac{C}{R}.$$

Apply Lemma 9 on $[0, T_\delta]$. For sufficiently small ε,

$$\left\| \frac{w_j(T_\delta)}{\|w_j(T_\delta)\|_2} - \bar{\omega}_j(T_\delta) \right\|_2 \le L_{T_\delta} m \varepsilon^2 \le \frac{\delta}{2}$$

for every $j \in [m]$. The triangle inequality therefore gives

$$\left\| \frac{w_j(T_\delta)}{\|w_j(T_\delta)\|_2} - \tau_j v_{J_j} \right\|_2 \le \delta + \frac{C}{R}.$$

The independence and uniformity of the labels $J_1, \dots, J_m$, together with the cluster-coverage probability, follow from Lemma 8. □

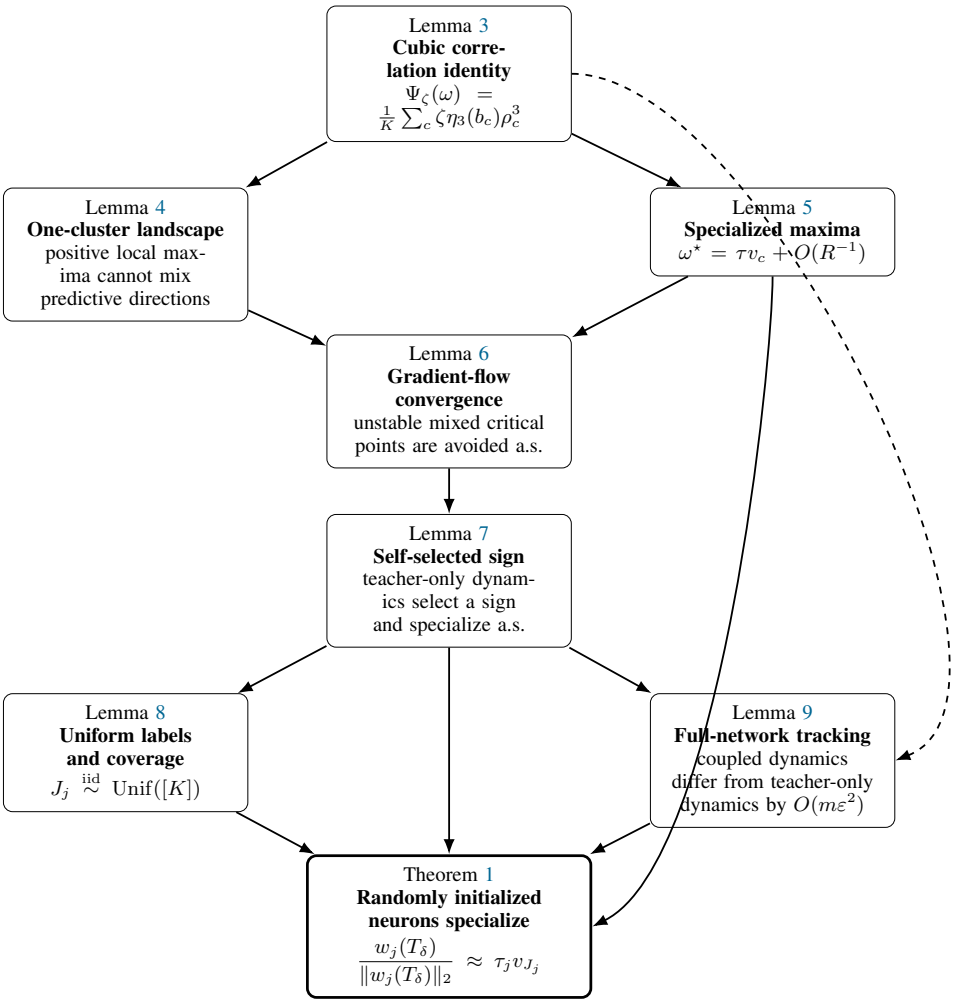


Figure 11: Dependency graph for the proof of Theorem 1. Solid arrows indicate the main logical dependencies. The dashed arrow records that Lemma 3 is also used to establish smoothness of the teacher-only vector field in the tracking argument.

B.1.1 The Population-Correlation Landscape

In this subsection, we study the critical points of the optimization problem

$$\max_{\omega \in \mathbb{S}^{d-1}} \Psi_\zeta(\omega)\,. \tag{9}$$

The use of homogeneity to characterize parameter directions is part of a broader literature on the implicit bias of gradient methods, beginning with homogeneous linear predictors and extending to homogeneous neural networks (Soudry et al., 2018; Ji & Telgarsky, 2019; Lyu & Li, 2020; Ji & Telgarsky, 2020). These results motivate the directional viewpoint used here, but do not directly yield the squared-loss, small-initialization approximation in our setting; we establish the required approximation directly in Lemma 9. (In the subsequent section of the appendix, we will show how the sign ζ is determined for each neuron at very early times of training.) We show that the critical points of equation 9 are specialized to clusters of the data distribution.

First, we provide an explicit formula for the signed population-correlation objective in terms of a weight's correlations with the cluster means and predictive directions. This lemma is stated for directions ω on the unit sphere, but is later applied to the directions of small weights using the homogeneity of ReLU networks.

Lemma 3 (Cubic correlation identity)**.** *For every $\omega \in \mathbb{S}^{d-1}$ and $\zeta \in \{\pm 1\}$,*

$$\Psi_\zeta(\omega) = -\frac{\zeta}{K\sqrt{6}} \sum_{c=1}^{K} b_c(\omega)\varphi\big(b_c(\omega)\big)\rho_c(\omega)^3.$$

Proof. Fix $\omega \in \mathbb{S}^{d-1}$. Conditional on c,

$$\omega^\top x = b_c(\omega) + \omega^\top z, \qquad y = h_3\big(v_c^\top z\big).$$

Hence the contribution of cluster c to $\Phi(\omega) = \mathbb{E}\left[y\,\phi(\omega^\top x)\right]$ is

$$\mathbb{E}\left[h_3\big(v_c^\top z\big)\big(b_c(\omega) + \omega^\top z\big)_+\right].$$

Set

$$U = v_c^\top z, \qquad V = \omega^\top z, \qquad \rho = \rho_c(\omega) = \langle v_c, \omega\rangle.$$

Since $z \sim \mathcal{N}(0, I_d)$ and v_c, ω are unit vectors, U and V are standard Gaussian, and

$$\mathbb{E}[UV] = v_c^\top \mathbb{E}[zz^\top]\omega = \langle v_c, \omega\rangle = \rho.$$

Thus (U, V) is a jointly standard Gaussian pair with correlation ρ.

For $|\rho| < 1$, define

$$Z := \frac{U - \rho V}{\sqrt{1-\rho^2}}.$$

Because (U, V) is jointly Gaussian, (Z, V) is also jointly Gaussian. Moreover,

$$\mathbb{E}[Z] = 0, \qquad \mathbb{E}[Z^2] = 1,$$

and

$$\mathbb{E}[ZV] = \frac{\mathbb{E}[UV] - \rho\mathbb{E}[V^2]}{\sqrt{1-\rho^2}} = 0.$$

Jointly Gaussian random variables with zero covariance are independent, so $Z \sim \mathcal{N}(0,1)$ is independent of V. Therefore

$$U = \rho V + \sqrt{1-\rho^2}\, Z.$$

When $|\rho| = 1$, the same representation holds with the second term equal to zero.

Conditioning on V and using the independence of Z gives

$$\mathbb{E}[U \mid V] = \rho V$$

and

$$\mathbb{E}[U^3 \mid V] = \mathbb{E}\left[\left(\rho V + \sqrt{1-\rho^2}\, Z\right)^3 \,\middle|\, V\right] = \rho^3 V^3 + 3\rho(1-\rho^2)V,$$

since $\mathbb{E}[Z] = \mathbb{E}[Z^3] = 0$ and $\mathbb{E}[Z^2] = 1$. Recalling that

$$h_3(x) = \frac{x^3 - 3x}{\sqrt{6}},$$

we obtain

$$\mathbb{E}[h_3(U) \mid V] = \frac{1}{\sqrt{6}}\left(\mathbb{E}[U^3 \mid V] - 3\mathbb{E}[U \mid V]\right) = \rho^3 h_3(V).$$

Since $(b_c(\omega) + V)_+$ depends only on V, the tower property now gives

$$\begin{aligned}\mathbb{E}\left[h_3(U)\big(b_c(\omega)+V\big)_+\right] &= \mathbb{E}\left[\big(b_c(\omega)+V\big)_+ \mathbb{E}[h_3(U) \mid V]\right] \\ &= \rho_c(\omega)^3 \mathbb{E}_{G\sim\mathcal{N}(0,1)}\left[h_3(G)\big(b_c(\omega)+G\big)_+\right].\end{aligned}$$

It remains to compute the one-dimensional expectation. For any $b \in \mathbb{R}$,

$$\mathbb{E}\left[h_3(G)(b+G)_+\right] = \frac{1}{\sqrt{6}} \int_{-b}^{\infty} (x^3 - 3x)(b+x)\varphi(x)\, dx.$$

Using

$$\frac{d}{dx}\left[(x^2-1)\varphi(x)\right] = -(x^3 - 3x)\varphi(x),$$

integration by parts yields

$$\int_{-b}^{\infty} (x^3 - 3x)(b+x)\varphi(x)\, dx = \int_{-b}^{\infty} (x^2-1)\varphi(x)\, dx,$$

where the boundary term vanishes because $b + x = 0$ at $x = -b$ and $\varphi(x) \to 0$ as $x \to \infty$. Since

$$\frac{d}{dx}\big(x\varphi(x)\big) = (1-x^2)\varphi(x),$$

we further obtain

$$\int_{-b}^{\infty} (x^2-1)\varphi(x)\, dx = -b\varphi(b).$$

Therefore

$$\mathbb{E}\left[h_3(G)(b+G)_+\right] = -\frac{b\varphi(b)}{\sqrt{6}}.$$

Applying this with $b = b_c(\omega)$, the contribution of cluster c to $\Phi(\omega)$ is

$$-\frac{1}{\sqrt{6}} b_c(\omega)\varphi\big(b_c(\omega)\big)\rho_c(\omega)^3.$$

Since c is uniform on $[K]$,

$$\Phi(\omega) = -\frac{1}{K\sqrt{6}} \sum_{c=1}^{K} b_c(\omega)\varphi\big(b_c(\omega)\big)\rho_c(\omega)^3.$$

Finally, $\Psi_\zeta(\omega) = \zeta\Phi(\omega)$, and hence

$$\Psi_\zeta(\omega) = -\frac{\zeta}{K\sqrt{6}} \sum_{c=1}^{K} b_c(\omega)\varphi\big(b_c(\omega)\big)\rho_c(\omega)^3,$$

as claimed. □

Because the μ_c/R and v_c vectors form an orthonormal basis, the sphere constraint is

$$\sum_{c=1}^{K} \frac{b_c^2}{R^2} + \sum_{c=1}^{K} \rho_c^2 = 1.$$

Next, we consider the optimization problem of signed population-correlation maximization between the value of the neuron and the ground truth: $\max_{\omega \in \mathbb{S}^{d-1}} \Psi_\zeta(\omega)$. This is written and analyzed below under the linear change of coordinates

$$b_c = \langle \omega, \mu_c \rangle, \qquad \rho_c = \langle \omega, v_c \rangle ,$$

where we show that the neuron maximizes correlation with the response by specializing to one cluster. Here, we use the fact that the link function is the cubic Hermite polynomial. (A similar argument would hold for all higher-degree Hermite link functions as well.)

Lemma 4 (Positive local maxima use one cluster)**.** *Consider the constrained maximization problem*

$$\max_{b_1,\dots,b_K,\rho_1,\dots,\rho_K} F(b_1, \dots, b_K, \rho_1, \dots, \rho_K),$$

where

$$F(b_1, \dots, b_K, \rho_1, \dots, \rho_K) := -\frac{\zeta}{K\sqrt{6}} \sum_{c=1}^{K} b_c \varphi(b_c) \rho_c^3,$$

subject to

$$\sum_{c=1}^{K} \frac{b_c^2}{R^2} + \sum_{c=1}^{K} \rho_c^2 = 1.$$

Every constrained local maximum of F with positive objective value has exactly one nonzero predictive coordinate ρ_c. Moreover, if $\rho_c = 0$, then the corresponding routing coordinate b_c also vanishes.

Proof. At a constrained local maximum, first-order stationarity of the Lagrangian

$$F(b_1, \dots, b_K, \rho_1, \dots, \rho_K) - \lambda \left(\sum_{c=1}^{K} \frac{b_c^2}{R^2} + \sum_{c=1}^{K} \rho_c^2 - 1 \right)$$

with respect to b_c and ρ_c gives, for every $c \in [K]$,

$$-\frac{\zeta}{K\sqrt{6}} (1 - b_c^2) \varphi(b_c) \rho_c^3 = \frac{2\lambda}{R^2} b_c, \tag{B.1}$$

$$-\frac{3\zeta}{K\sqrt{6}} b_c \varphi(b_c) \rho_c^2 = 2\lambda \rho_c. \tag{B.2}$$

Here we used

$$\frac{d}{db}\big(b\varphi(b)\big) = (1 - b^2)\varphi(b).$$

Multiplying equation B.2 by ρ_c and summing over c gives

$$3F(b_1, \dots, b_K, \rho_1, \dots, \rho_K) = 2\lambda \sum_{c=1}^{K} \rho_c^2.$$

Since the objective value is positive, at least one predictive coordinate is nonzero. Hence both the left-hand side and $\sum_c \rho_c^2$ are positive, so

$$\lambda > 0.$$

If $\rho_c = 0$, then equation B.1 reduces to

$$0 = \frac{2\lambda}{R^2} b_c.$$

Since $\lambda > 0$, this implies $b_c = 0$.

It remains to rule out two nonzero predictive coordinates. Suppose that $\rho_c \neq 0$ and $\rho_r \neq 0$ for two distinct clusters $c \neq r$. Keep all routing coordinates and all other predictive coordinates fixed, and consider

$$\rho_c(t) = \operatorname{sign}(\rho_c)\sqrt{\rho_c^2 + t}, \qquad \rho_r(t) = \operatorname{sign}(\rho_r)\sqrt{\rho_r^2 - t}.$$

For sufficiently small $|t|$, this perturbation is well defined, preserves the signs of the two coordinates, and satisfies

$$\rho_c(t)^2 + \rho_r(t)^2 = \rho_c^2 + \rho_r^2.$$

Thus it preserves the constraint exactly.

Differentiating F along this feasible curve at $t = 0$ gives

$$F'(0) = -\frac{3\zeta}{2K\sqrt{6}} \left(b_c \varphi(b_c)\rho_c - b_r \varphi(b_r)\rho_r \right).$$

Since $\rho_c, \rho_r \neq 0$, dividing equation B.2 by the corresponding predictive coordinate gives

$$-\frac{3\zeta}{K\sqrt{6}} b_c \varphi(b_c)\rho_c = 2\lambda, \qquad -\frac{3\zeta}{K\sqrt{6}} b_r \varphi(b_r)\rho_r = 2\lambda.$$

Therefore

$$F'(0) = 0.$$

The second derivative along the same feasible curve is

$$F''(0) = -\frac{3\zeta}{4K\sqrt{6}} \left(\frac{b_c \varphi(b_c)}{\rho_c} + \frac{b_r \varphi(b_r)}{\rho_r} \right).$$

Using the same KKT identities gives

$$F''(0) = \frac{\lambda}{2} \left(\frac{1}{\rho_c^2} + \frac{1}{\rho_r^2} \right) > 0.$$

Thus a critical point with two nonzero predictive coordinates has a feasible direction of positive curvature and cannot be a constrained local maximum.

Hence at most one predictive coordinate is nonzero. Since the objective value is positive, at least one is nonzero, so exactly one is active. As shown above, every routing coordinate corresponding to an inactive predictive coordinate also vanishes. □

Returning to our objective $\max_{\omega \in \mathbb{S}^{d-1}} \Psi_\zeta(\omega)$, let c denote the unique active cluster and let

$$\tau = \operatorname{sign}(\rho_c) \in \{\pm 1\}.$$

Then

$$b_j = \rho_j = 0 \qquad \text{for every } j \neq c,$$

and the constraint becomes

$$\frac{b_c^2}{R^2} + \rho_c^2 = 1.$$

Hence

$$\rho_c = \tau \sqrt{1 - \frac{b_c^2}{R^2}}.$$

Substituting this into F, and writing $b = b_c$, reduces the problem to the one-dimensional objective

$$-\frac{\tau\zeta}{K\sqrt{6}} b\varphi(b) \left(1 - \frac{b^2}{R^2} \right)^{3/2}, \qquad |b| < R.$$

Lemma 5 (Specialized maxima)**.** *Fix $c \in [K]$ and $\zeta, \tau \in \{\pm 1\}$. On the one-cluster feasible branch, write*

$$f(b) := -\frac{\tau\zeta}{K\sqrt{6}} b\varphi(b) \left(1 - \frac{b^2}{R^2} \right)^{3/2}, \qquad |b| < R,$$

and let

$$I := \{ b \in (-R, R) : f(b) > 0 \}.$$

Then I is a connected open interval. For all sufficiently large R, f has a unique critical point $b^\star \in I$, which is a strict local maximum and satisfies

$$b^\star = -\tau\zeta + O(R^{-2}).$$

The corresponding direction

$$\omega^\star = \tau\sqrt{1 - \frac{(b^\star)^2}{R^2}}\, v_c + \frac{b^\star}{R^2}\mu_c$$

is a strict local maximum of Ψ_ζ and obeys

$$\omega^\star = \tau v_c - \frac{\tau\zeta}{R^2}\mu_c + O(R^{-2}).$$

Proof. For every $b \in (-R, R)$,

$$\frac{1}{K\sqrt{6}} > 0, \qquad \varphi(b) > 0, \qquad \left(1 - \frac{b^2}{R^2}\right)^{3/2} > 0.$$

Hence the sign of $f(b)$ is determined entirely by $-\tau\zeta b$:

$$f(b) > 0 \quad \Longleftrightarrow \quad -\tau\zeta b > 0.$$

Since $\tau\zeta \in \{\pm 1\}$, it follows that

$$I = \begin{cases} (-R, 0), & \tau\zeta = 1, \\ (0, R), & \tau\zeta = -1. \end{cases}$$

Thus I is connected.

For $b \in I$, set

$$t = -\tau\zeta b.$$

The preceding characterization of I shows that this is a bijective linear change of variable from I onto $(0, R)$. Since φ is even and $(\tau\zeta)^2 = 1$, we obtain

$$f(b) = \frac{1}{K\sqrt{6}} t\varphi(t)\left(1 - \frac{t^2}{R^2}\right)^{3/2}.$$

Therefore the critical points of f on I correspond exactly to the critical points on $(0, R)$ of

$$t \longmapsto t\varphi(t)\left(1 - \frac{t^2}{R^2}\right)^{3/2}.$$

This function is strictly positive on $(0, R)$, so its critical points can be found from its logarithmic derivative:

$$\frac{d}{dt}\log\left[t\varphi(t)\left(1 - \frac{t^2}{R^2}\right)^{3/2}\right] = \frac{1}{t} - t - \frac{3t}{R^2 - t^2}.$$

Thus a critical point satisfies

$$\frac{1}{t} - t - \frac{3t}{R^2 - t^2} = 0.$$

Multiplying by $t(R^2 - t^2)$ gives

$$t^4 - (R^2 + 4)t^2 + R^2 = 0.$$

Solving this quadratic equation in t^2 gives

$$t^2 = \frac{R^2 + 4 \pm \sqrt{R^4 + 4R^2 + 16}}{2}.$$

The root with the plus sign is larger than R^2, whereas the root with the minus sign lies in $(0, R^2)$. Hence there is exactly one critical point in $(0, R)$, and it satisfies

$$t^2 = \frac{R^2 + 4 - \sqrt{R^4 + 4R^2 + 16}}{2} = 1 + O(R^{-2}).$$

Since $t > 0$,

$$t = 1 + O(R^{-2}).$$

Returning to $b = -\tau\zeta t$ gives

$$b^\star = -\tau\zeta + O(R^{-2}).$$

The function

$$t\varphi(t)\left(1 - \frac{t^2}{R^2}\right)^{3/2}$$

is positive on $(0, R)$ and tends to zero as $t \to 0$ or $t \to R$. Since it has exactly one critical point in $(0, R)$, this point is its unique strict maximum. Hence $b^\star$ is the unique maximizer of f on I.

It remains to show that the corresponding point is a local maximum of the full constrained objective F. Since the feasible set is compact, F attains a global maximum. The point constructed above has positive objective value, so the global maximum is positive. By Lemma 4, every global maximizer must have exactly one nonzero predictive coordinate and therefore lies on one of the one-cluster feasible branches.

For every choice of the active cluster c and orientation τ, the change of variable

$$t = -\tau\zeta b$$

reduces the positive part of the corresponding one-cluster objective to

$$\frac{1}{K\sqrt{6}} t\varphi(t)\left(1 - \frac{t^2}{R^2}\right)^{3/2}.$$

Thus all one-cluster branches have the same maximal value, attained uniquely at the point identified above. Consequently, every corresponding $\omega^\star$ is a global maximizer of Ψ_ζ.

There are only finitely many such maximizers, one for each $c \in [K]$ and $\tau \in \{\pm 1\}$. Hence each is isolated, and therefore each $\omega^\star$ is a strict local maximum of Ψ_ζ.

□

B.1.2 Random Initialization, Self-Selected Signs, and Cluster Coverage

In the previous section, we considered the signed population-correlation objective Ψ_ζ for each neuron. In this section, we note that the sign ζ for each neuron is learned at very early times of training and does not need to be fixed. Consider the teacher-only joint dynamics

$$\dot{u} = \Phi(\omega), \qquad \dot{\omega} = \tanh(u)\nabla_{\mathbb{S}^{d-1}}\Phi(\omega). \tag{10}$$

These equations arise from the two-layer parameterization when the second-layer weight is initially zero. Crucially, they neglect interactions between neurons, which we will later show is a fine approximation because the network is small at initialization.

Lemma 6 (Positive signed-correlation flows specialize)**.** *Let ω^0 have an absolutely continuous distribution on $\mathbb{S}^{d-1}$, fix $\zeta \in \{\pm 1\}$, and assume $\Psi_\zeta(\omega^0) > 0$. For all sufficiently large R, spherical gradient ascent*

$$\dot{\omega} = \nabla_{\mathbb{S}^{d-1}}\Psi_\zeta(\omega), \qquad \omega(0) = \omega^0,$$

converges almost surely to one of the specialized maxima in Lemma 5.

Proof. Along spherical gradient ascent,

$$\frac{d}{dt}\Psi_\zeta(\omega(t)) = \|\nabla_{\mathbb{S}^{d-1}}\Psi_\zeta(\omega(t))\|_2^2 \geq 0.$$

Since the sphere is compact, Ψ_ζ is bounded above. Hence

$$\int_0^\infty \|\nabla_{\mathbb{S}^{d-1}}\Psi_\zeta(\omega(t))\|_2^2 \, dt < \infty.$$

Moreover, Ψ_ζ is smooth on the sphere, so its gradient and Hessian are bounded. It follows that $\|\nabla_{\mathbb{S}^{d-1}}\Psi_\zeta(\omega(t))\|_2^2$ has bounded derivative. A nonnegative integrable function with bounded derivative must converge to zero, and therefore

$$\|\nabla_{\mathbb{S}^{d-1}}\Psi_\zeta(\omega(t))\|_2 \longrightarrow 0.$$

We next note that Ψ_ζ has only finitely many positive critical points. Indeed, at any positive critical point, the stationarity conditions equation B.1–equation B.2 imply that $\lambda > 0$, and hence $b_c = 0$ whenever $\rho_c = 0$. For every active coordinate $\rho_c \neq 0$, eliminating λ from the two stationarity equations gives

$$(1 - b_c^2)\rho_c^2 = \frac{3b_c^2}{R^2}.$$

Thus $0 < |b_c| < 1$ and

$$\rho_c^2 = \frac{3b_c^2}{R^2(1 - b_c^2)}.$$

Equation B.2 further shows that

$$|b_c|\varphi(b_c)|\rho_c|$$

has the same value for every active coordinate. Substituting the preceding expression for ρ_c^2, this quantity is proportional to

$$\frac{|b_c|^2\varphi(|b_c|)}{\sqrt{1 - |b_c|^2}},$$

which is strictly increasing for $0 < |b_c| < 1$, since

$$\frac{d}{dt}\log\left(\frac{t^2\varphi(t)}{\sqrt{1-t^2}}\right) = \frac{2}{t} + \frac{t^3}{1-t^2} > 0.$$

Hence all active $|b_c|$ are equal. If there are s active coordinates, the sphere constraint then gives

$$s\,\frac{b_c^2(4 - b_c^2)}{R^2(1 - b_c^2)} = 1,$$

which has a unique solution for $b_c^2 \in (0, 1)$. Thus, for each choice of the active coordinates and their signs, there is at most one positive critical point. Since there are only finitely many such choices, the set of positive critical points is finite.

Because $\Psi_\zeta(\omega(0)) > 0$ and the objective is nondecreasing, every accumulation point of the trajectory has positive objective value. By compactness, accumulation points exist, and the convergence of the gradient to zero implies that every accumulation point is a positive critical point. Since there are only finitely many such points, the trajectory must converge to one of them.

Finally, the argument in Lemma 4 gives an unstable tangent direction at every positive critical point with more than one active predictive coordinate. Since there are only finitely many such critical points, the center-stable manifold theorem implies that the set of initial conditions converging to any of them has measure zero. Lemma 5 shows that the remaining positive critical points are precisely the specialized maxima. Hence an absolutely continuous initialization converges almost surely to a specialized maximum. □

For a single neuron, write

$$\omega(t) = \frac{w(t)}{\|w(t)\|_2}.$$

ReLU homogeneity implies that gradient flow preserves

$$\|w(t)\|_2^2 - a(t)^2 = \|w(0)\|_2^2 - a(0)^2 = \varepsilon^2.$$

We therefore define the scalar amplitude coordinate

$$u(t) := \operatorname{arsinh}\left(\frac{a(t)}{\varepsilon}\right),$$

so that, equivalently,

$$a(t) = \varepsilon \sinh u(t), \qquad \|w(t)\|_2 = \varepsilon \cosh u(t).$$

Under this parametrization, the target-only part of the directional dynamics is

$$\dot{u} = \Phi(\omega), \qquad \dot{\omega} = \tanh(u)\nabla_{\mathbb{S}^{d-1}}\Phi(\omega). \tag{11}$$

Lemma 7 (Self-selected sign and almost-sure specialization)**.** *Let ω^0 have an absolutely continuous distribution on $\mathbb{S}^{d-1}$, and let $(u(t), \omega(t))$ solve equation 11 with*

$$u(0) = 0, \qquad \omega(0) = \omega^0.$$

For almost every ω^0, define

$$\zeta = \operatorname{sign}(\Phi(\omega^0)) \in \{\pm 1\}.$$

Then, for every $t > 0$,

$$\zeta u(t) > 0, \qquad \zeta \Phi(\omega(t)) \geq |\Phi(\omega^0)|.$$

Moreover, $\omega(t)$ is a positive time reparameterization of spherical gradient ascent on $\Psi_\zeta = \zeta\Phi$, and, for all sufficiently large R, it converges almost surely to one of the specialized maxima in Lemma 5.

Proof. Since Φ is real analytic and not identically zero, its zero set has spherical measure zero. Thus, for almost every ω^0, $\Phi(\omega^0) \neq 0$. Fix such a ω^0 and let

$$\zeta = \operatorname{sign}(\Phi(\omega^0)).$$

At $t = 0$,

$$\frac{d}{dt}\big(\zeta u(t)\big)\Big|_{t=0} = \zeta\Phi(\omega^0) = |\Phi(\omega^0)| > 0,$$

so $\zeta u(t) > 0$ for all sufficiently small $t > 0$.

As long as $\zeta u(t) > 0$, $\tanh(u(t))$ has sign ζ, and hence

$$\zeta \tanh(u(t)) = |\tanh(u(t))|.$$

Therefore

$$\frac{d}{dt}\big(\zeta\Phi(\omega(t))\big) = |\tanh(u(t))|\, \|\nabla_{\mathbb{S}^{d-1}}\Phi(\omega(t))\|_2^2 \geq 0.$$

It follows that

$$\zeta\Phi(\omega(t)) \geq \zeta\Phi(\omega^0) = |\Phi(\omega^0)|.$$

Consequently,

$$\frac{d}{dt}\big(\zeta u(t)\big) = \zeta\Phi(\omega(t)) \geq |\Phi(\omega^0)|,$$

and therefore

$$\zeta u(t) \geq |\Phi(\omega^0)| t.$$

In particular, $\zeta u(t)$ cannot return to zero, so the preceding inequalities hold for every $t > 0$. This proves

$$\zeta u(t) > 0, \qquad \zeta\Phi(\omega(t)) \geq |\Phi(\omega^0)|.$$

Since $\Psi_\zeta = \zeta\Phi$ and $\tanh(u(t))$ has sign ζ, the directional dynamics satisfy

$$\dot{\omega}(t) = |\tanh(u(t))| \nabla_{\mathbb{S}^{d-1}} \Psi_\zeta(\omega(t)).$$

Thus $\omega(t)$ follows the same orbit as spherical gradient ascent on Ψ_ζ, up to a positive reparameterization of time. Moreover, $\zeta u(t) \geq |\Phi(\omega^0)| t$ implies $|u(t)| \to \infty$, and hence $|\tanh(u(t))| \to 1$. In particular,

$$\int_0^\infty |\tanh(u(t))|\, dt = \infty,$$

so this reparameterization covers the entire forward gradient-ascent trajectory.

Finally,

$$\Psi_\zeta(\omega^0) = \zeta\Phi(\omega^0) = |\Phi(\omega^0)| > 0.$$

Lemma 6 therefore implies that $\omega(t)$ converges almost surely to one of the specialized maxima in Lemma 5. □

Lemma 8 (Uniform selected labels and coverage)**.** *Let $\omega_1^0, \dots, \omega_m^0$ be independent and uniform on $\mathbb{S}^{d-1}$, and let J_j be the cluster selected by the self-selected flow in Lemma 7. Then $J_1, \dots, J_m$ are independent and uniform on $[K]$. Consequently,*

$$\mathbb{P}\big(\{J_1, \dots, J_m\} \neq [K]\big) \leq K\left(1 - \frac{1}{K}\right)^m \leq Ke^{-m/K}.$$

Proof. For a permutation π of $[K]$, let P_π be the orthogonal map satisfying

$$P_\pi \frac{\mu_c}{R} = \frac{\mu_{\pi(c)}}{R}, \qquad P_\pi v_c = v_{\pi(c)}.$$

The data distribution, target, and self-selected vector field are equivariant under P_π. Hence, if ω^0 selects cluster c, then $P_\pi \omega^0$ selects cluster $\pi(c)$. Uniform spherical measure is invariant under P_π, so all selection probabilities are equal. By Lemma 7, they sum to one, and each is therefore $1/K$. Independence follows because each teacher-only label is a deterministic function of an independent initial direction.

A fixed cluster is missed with probability $(1-1/K)^m$. A union bound yields the coverage estimate. □

B.1.3 Tracking by the Full Small-Initialization Dynamics

Finally, we put the above ingredients analyzing the trajectories of individual neurons together, and show that they describe the trajectory of the neurons in a neural network (up to rescaling) when the network is initialized small. The proof strategy of studying feature learning through effectively independent neuron dynamics in early-time or small-initialization regimes has been put forward in prior work (Abbe et al., 2022; Min et al., 2024; Glasgow, 2024). The population loss is

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}\left[\left(f_\theta(x) - y\right)^2\right],$$

and its gradient-flow equations are

$$\dot{a}_j = \mathbb{E}\left[(y - f_\theta(x))\phi(w_j^\top x)\right],$$
$$\dot{w}_j = a_j \mathbb{E}\left[(y - f_\theta(x))\mathbf{1}_{\{w_j^\top x > 0\}} x\right].$$

Lemma 9 (Early-time tracking from a zero output layer)**.** *Initialize*

$$a_j(0) = 0, \qquad w_j(0) = \varepsilon\omega_j^0, \qquad \omega_j^0 \in \mathbb{S}^{d-1}.$$

Let $(\bar{u}_j, \bar{\omega}_j)$ solve the self-selected teacher-only system

$$\dot{\bar{u}}_j = \Phi(\bar{\omega}_j), \qquad \dot{\bar{\omega}}_j = \tanh(\bar{u}_j)\nabla_{\mathbb{S}^{d-1}}\Phi(\bar{\omega}_j),$$

with

$$(\bar{u}_j(0), \bar{\omega}_j(0)) = (0, \omega_j^0).$$

For every fixed $T < \infty$, there are constants $L_T < \infty$ and $\varepsilon_T > 0$ such that, for $0 < \varepsilon \le \varepsilon_T$,

$$\sup_{0 \le t \le T} \left(|u_j(t) - \bar{u}_j(t)| + \|\omega_j(t) - \bar{\omega}_j(t)\|_2\right) \le L_T m \varepsilon^2,$$

where

$$a_j(t) = \varepsilon \sinh(u_j(t)), \qquad \omega_j(t) = \frac{w_j(t)}{\|w_j(t)\|_2}.$$

Proof. ReLU homogeneity gives the invariant

$$\frac{d}{dt}\left(\|w_j\|_2^2 - a_j^2\right) = 0.$$

Under the stated initialization,

$$\|w_j(t)\|_2^2 - a_j(t)^2 = \varepsilon^2.$$

Since $a_j = \varepsilon \sinh u_j$, it follows that

$$\|w_j\|_2 = \varepsilon \cosh u_j.$$

Substituting

$$w_j = \varepsilon \cosh(u_j)\omega_j$$

into the gradient-flow equations and using ReLU homogeneity gives

$$\dot{u}_j = \Phi(\omega_j) - \mathbb{E}\left[f_\theta(x)\phi({\omega_j}^\top x)\right],$$
$$\dot{\omega}_j = \tanh(u_j)\Bigg(\nabla_{\mathbb{S}^{d-1}}\Phi(\omega_j) - \left(I - \omega_j\omega_j^\top\right)\mathbb{E}\left[f_\theta(x)\mathbf{1}_{\{{\omega_j}^\top x>0\}}x\right]\Bigg).$$

Lemma 3 shows that the teacher-only vector field is smooth on bounded u-intervals and on the sphere. On every fixed interval $[0,T]$, the quantities $u_j(t)$ remain uniformly bounded for sufficiently small ε. Consequently, there is a constant $C_T < \infty$ such that

$$|a_j(t)| + \|w_j(t)\|_2 \le C_T\varepsilon, \qquad 0 \le t \le T.$$

Since x has finite moments of every order,

$$\|f_{\theta(t)}\|_{L^2(P_x)} \le C_T \sum_{j=1}^{m} |a_j(t)|\|w_j(t)\|_2 \le C_T m\varepsilon^2.$$

Cauchy–Schwarz therefore bounds both interaction terms in the (u_j, ω_j) equations by $C_T m\varepsilon^2$.

The full and teacher-only systems have the same initial conditions, and their vector fields are Lipschitz on the relevant compact set. Grönwall's inequality yields

$$\sup_{0\le t\le T}\left(|u_j(t) - \bar{u}_j(t)| + \|\omega_j(t) - \bar{\omega}_j(t)\|_2\right) \le L_T m\varepsilon^2.$$

□

Perturbations of the cubic link function. The Hermite link function isolates the specialization mechanism and makes the correlation landscape exact. More generally, one could perturb the induced correlation objective and expect specialization to persist. In particular, the specialized maxima and their attraction basins should be stable under sufficiently small perturbations near both the maxima and the strict-saddle regions.

B.2 Proof of Theorem 2

We first prove the positive MLP result in a more general fixed-dimensional multi-index setting. We then recover the single-index MLP result in Theorem 2 as a corollary and prove the kernel and RFM lower bounds in that single-index setting.

B.2.1 A General Multi-Index Specialized MLP Construction

Fix an integer $r_0 \ge 1$ and a constant $R \ge 1$. For each K, let $d = K + Kr_0$. Let $E_c \in \mathbb{R}^{Kr_0\times r_0}$ embed $\mathbb{R}^{r_0}$ into the cth block of $\mathbb{R}^{Kr_0}$, so that

$$E_c^\top E_{c'} = \mathbf{1}_{\{c=c'\}} I_{r_0}.$$

Define

$$\mu_c = \begin{bmatrix} Re_c \\ 0 \end{bmatrix} \in \mathbb{R}^d, \qquad V_c = \begin{bmatrix} 0 \\ E_c \end{bmatrix} \in \mathbb{R}^{d\times r_0}.$$

The data distribution is

$$C \sim \mathrm{Unif}([K]), \qquad X = \mu_C + \begin{bmatrix} 0 \\ Z \end{bmatrix}, \qquad Z \sim \mathcal{N}(0, I_{Kr_0}).$$

Let $g : \mathbb{R}^{r_0} \to \mathbb{R}$ be fixed, bounded, and Lipschitz, with

$$\|g\|_\infty \le G_0, \qquad |g(u) - g(u')| \le L\|u - u'\|_2,$$

and define

$$f^\star_{r_0}(X) = g(V_C^\top X).$$

Thus, within cluster c, the target may depend on all r_0 coordinates in the cluster-specific predictive block $V_c^\top X$.

For a two-layer ReLU network, define the path norm

$$\|f\|_{\mathcal{P}} = \inf\left\{\sum_j |a_j|\|w_j\|_2 : f(x) = \sum_j a_j\phi(w_j^\top x)\right\}.$$

For a width-m parameterization $\theta = ((a_j, w_j))_{j=1}^m$, define

$$\|\theta\|_{\mathrm{F}}^2 := \frac{1}{2}\sum_{j=1}^m \left(a_j^2 + \|w_j\|_2^2\right),$$

and let

$$\mathcal{F}_{B,m} := \left\{f_\theta(x) = \sum_{j=1}^m a_j\phi(w_j^\top x) : \|\theta\|_{\mathrm{F}}^2 \le B\right\}.$$

Define the ramp function

$$\psi(t) := \phi(t) - \phi(t-1).$$

Every occurrence of ψ below is implemented by two ordinary ReLU neurons; the network itself has neither hidden biases nor output clipping.

Lemma 10 (Ridge-ramp approximation)**.** *Let $\gamma_{r_0} = \mathcal{N}(0, I_{r_0})$. Finite ridge-ramp expansions of the form*

$$p(u) = b + \sum_{j=1}^J a_j\psi(q_j^\top u - t_j)$$

are dense in $L^2(\gamma_{r_0})$. Consequently, there is a sequence $(p_s)_{s\ge1}$ of such functions satisfying

$$\|p_s - g\|_{L^2(\gamma_{r_0})} \longrightarrow 0.$$

Proof. Suppose $h \in L^2(\gamma_{r_0})$ is orthogonal to every ridge-ramp function, and define the finite signed measure

$$\nu(A) := \int_A h(u)\, d\gamma_{r_0}(u).$$

For every $q \neq 0$, $t \in \mathbb{R}$, and $\lambda > 0$,

$$\int h(u)\psi\big(\lambda(q^\top u - t)\big)\; d\gamma_{r_0}(u) = 0.$$

Because $0 \le \psi \le 1$, dominated convergence gives, as $\lambda \to \infty$,

$$\nu\big(\{u : q^\top u > t\}\big) = 0.$$

Halfspaces determine finite Borel measures on $\mathbb{R}^{r_0}$, so $\nu = 0$ and therefore $h = 0$ in $L^2(\gamma_{r_0})$. This proves density. □

Lemma 11 (Routing a multi-index approximant)**.** *Let*

$$p(u) = b + \sum_{j=1}^J a_j\psi(q_j^\top u - t_j).$$

There is a two-layer ReLU network

$$F_{K,p} = \sum_{c=1}^K F_{K,p,c}$$

with no hidden biases such that, on an input from cluster c_0,

$$F_{K,p,c}(x) = \mathbf{1}_{\{c=c_0\}} p(V_c^\top x).$$

In particular, each cluster-specific group cancels on all other clusters. Every hidden weight in $F_{K,p,c}$ lies in

$$\operatorname{span}\big(\{\mu_c\} \cup \operatorname{range}(V_c)\big).$$

The network has width at most $K(2J+1)$ and path norm at most

$$K\,\mathcal{C}_R(p),$$

where

$$\mathcal{C}_R(p) := \frac{|b|}{R} + \sum_{j=1}^{J} |a_j| \left[\sqrt{\|q_j\|_2^2 + \frac{t_j^2}{R^2}} + \sqrt{\|q_j\|_2^2 + \frac{(t_j+1)^2}{R^2}} \right].$$

It also admits a parameterization satisfying

$$\|\theta\|_{\mathrm{F}}^2 \le K\,\mathcal{C}_R(p).$$

Proof. Define

$$F_{K,p,c}(x) := b\,\phi\left(\frac{\mu_c^\top x}{R^2}\right) + \sum_{j=1}^{J} a_j \phi\left(\left(V_c q_j - \frac{t_j}{R^2}\mu_c\right)^\top x\right) - \sum_{j=1}^{J} a_j \phi\left(\left(V_c q_j - \frac{t_j+1}{R^2}\mu_c\right)^\top x\right).$$

On the support of the data distribution,

$$\frac{\mu_c^\top x}{R^2} = \mathbf{1}_{\{C=c\}}.$$

Suppose x belongs to cluster c_0. If $c \neq c_0$, the two ReLUs in each pair have the same argument and cancel, while the routing neuron is zero. If $c = c_0$, then

$$F_{K,p,c_0}(x) = b + \sum_{j=1}^{J} a_j \left[\phi(q_j^\top V_{c_0}^\top x - t_j) - \phi(q_j^\top V_{c_0}^\top x - t_j - 1)\right] = p(V_{c_0}^\top x).$$

This proves the specialization claim.

The routing weight has norm $1/R$. The two hidden weights associated with (a_j, q_j, t_j) have norms

$$\sqrt{\|q_j\|_2^2 + \frac{t_j^2}{R^2}} \quad \text{and} \quad \sqrt{\|q_j\|_2^2 + \frac{(t_j+1)^2}{R^2}}.$$

Summing the corresponding path-norm contributions over all clusters gives the stated bound. Finally, positive homogeneity allows each neuron to be balanced so that $|a_j| = \|w_j\|_2$. For this parameterization, its contribution to $\|\theta\|_{\mathrm{F}}^2$ equals its contribution to the path norm. □

The finite-network path norm is the discrete analogue of the Barron or variation norm for two-layer networks (Bach, 2017; E et al., 2022). The following sample-dependent estimate is a version of standard path-norm Rademacher-complexity bounds (E et al., 2022; Golowich et al., 2018).

Lemma 12 (Path-norm generalization without clipping)**.** *For any deterministic sample $x_1, \dots, x_n \in \mathbb{R}^d$,*

$$\widehat{\mathfrak{R}}_n(\mathcal{F}_{B,m}) \le \frac{2B}{n}\left(\sum_{i=1}^n \|x_i\|_2^2\right)^{1/2}.$$

For the multi-index distribution above, let $\widehat{f}_{\mathrm{MLP}}^{(r_0)}$ be an empirical squared-loss minimizer over $\mathcal{F}_{B,m}$. There is a universal constant $C < \infty$ such that

$$\begin{aligned}\mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}}^{(r_0)} - f_{r_0}^\star\|_{L^2(P_x)}^2\right] \le \inf_{f\in\mathcal{F}_{B,m}} \|f - f_{r_0}^\star\|_{L^2(P_x)}^2 \\ + CB^2\sqrt{\frac{(Kr_0+R^2)(Kr_0+R^2+\log(en))}{n}} \\ + CBG_0\sqrt{\frac{Kr_0+R^2}{n}}.\end{aligned}$$

Proof. By the arithmetic–geometric mean inequality, every $f \in \mathcal{F}_{B,m}$ has path norm at most B. By positive homogeneity, every such function can be represented as

$$f(x) = \sum_j c_j \phi(\omega_j^\top x), \qquad \|\omega_j\|_2 = 1, \qquad \sum_j |c_j| \le B.$$

Thus $\mathcal{F}_{B,m}$ is contained in B times the absolutely convex hull of unit-norm ReLU atoms. Contraction and Cauchy–Schwarz give

$$\begin{aligned}\widehat{\mathfrak{R}}_n(\mathcal{F}_{B,m}) &\le \frac{B}{n}\mathbb{E}_\epsilon \sup_{\|\omega\|_2\le 1}\left|\sum_{i=1}^n \epsilon_i \phi(\omega^\top x_i)\right| \\ &\le \frac{2B}{n}\mathbb{E}_\epsilon\left\|\sum_{i=1}^n \epsilon_i x_i\right\|_2 \\ &\le \frac{2B}{n}\left(\sum_{i=1}^n \|x_i\|_2^2\right)^{1/2}.\end{aligned}$$

For the squared-loss bound, let

$$M_n := \max_{1\le i\le n}\|x_i\|_2.$$

The path-norm bound implies

$$|f(x_i)| \le BM_n \qquad \text{for every } f \in \mathcal{F}_{B,m}.$$

Since $|y_i| \le G_0$, the squared loss is $2(BM_n+G_0)$-Lipschitz in the prediction over the range attained on the sample. The standard ERM symmetrization and contraction argument therefore gives

$$\begin{aligned}\mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}}^{(r_0)} - f_{r_0}^\star\|_{L^2(P_x)}^2\right] \le \inf_{f\in\mathcal{F}_{B,m}} \|f - f_{r_0}^\star\|_{L^2(P_x)}^2 \\ + C\,\mathbb{E}\left[\frac{B(BM_n+G_0)}{n}\left(\sum_{i=1}^n \|x_i\|_2^2\right)^{1/2}\right].\end{aligned}$$

Under the multi-index distribution,

$$\|x_i\|_2^2 = R^2 + \|Z_i\|_2^2, \qquad Z_i \sim \mathcal{N}(0, I_{Kr_0}).$$

Hence

$$\mathbb{E}\sum_{i=1}^n \|x_i\|_2^2 = n(Kr_0+R^2),$$

and Gaussian norm concentration gives

$$\mathbb{E}[M_n^2] \le C\big(Kr_0 + R^2 + \log(en)\big).$$

The result follows from Cauchy–Schwarz. □

Theorem 13 (Specialized multi-index MLP approximation and consistency)**.** *Fix r_0, R, and a bounded Lipschitz function $g : \mathbb{R}^{r_0} \to \mathbb{R}$ as above. There is a sequence of cluster-specialized two-layer ReLU networks (F_K) satisfying*

$$m(F_K) \le \left\lceil K^{9/8} \right\rceil, \qquad \|\theta(F_K)\|_{\mathrm{F}}^2 \le K^{9/8},$$

and

$$\|F_K - f_{r_0}^\star\|_{L^2(P_x)}^2 \longrightarrow 0.$$

Moreover, if

$$m_K = \left\lceil K^{9/8} \right\rceil, \qquad B_K = K^{9/8}, \qquad n_K = \lceil K^7 \rceil,$$

then any empirical squared-loss minimizer over $\mathcal{F}_{B_K, m_K}$ satisfies

$$\lim_{K\to\infty} \mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}}^{(r_0)} - f_{r_0}^\star\|_{L^2(P_x)}^2\right] = 0.$$

Proof. By Lemma 10, choose ridge-ramp networks $(p_s)_{s\ge 1}$ such that

$$\|p_s - g\|_{L^2(\gamma_{r_0})} \longrightarrow 0.$$

Write J_s for the number of ridge-ramp units in p_s, and define

$$D_s := \max\{2J_s + 1, \mathcal{C}_R(p_s), 1\}, \qquad A_s := \max_{1\le j\le s} D_j.$$

Let $p_0 = 0$ and set

$$s_K := \max\left(\{0\} \cup \{s \le K : A_s \le K^{1/8}\}\right).$$

Every A_s is finite, so $s_K \to \infty$.

Apply Lemma 11 to p_{s_K}. The resulting network F_K has width at most

$$K(2J_{s_K} + 1) \le K^{9/8}$$

and admits a parameterization satisfying

$$\|\theta(F_K)\|_{\mathrm{F}}^2 \le K\mathcal{C}_R(p_{s_K}) \le K^{9/8}.$$

It is cluster-specialized in the sense of Lemma 11, and

$$\|F_K - f_{r_0}^\star\|_{L^2(P_x)}^2 = \|p_{s_K} - g\|_{L^2(\gamma_{r_0})}^2 \longrightarrow 0.$$

Applying Lemma 12 with $B_K = K^{9/8}$ and $n_K = \lceil K^7 \rceil$ gives

$$\begin{aligned}\mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}}^{(r_0)} - f_{r_0}^\star\|_{L^2(P_x)}^2\right] \le \|F_K - f_{r_0}^\star\|_{L^2(P_x)}^2 &\\ + CB_K^2\sqrt{\frac{(Kr_0 + R^2)(Kr_0 + R^2 + \log(en_K))}{n_K}} &\\ + CB_KG_0\sqrt{\frac{Kr_0 + R^2}{n_K}}.&\end{aligned}$$

Because r_0 and R are fixed, the two estimation terms are respectively $O(K^{-1/4})$ and $O(K^{-15/8})$. Both vanish, proving consistency. □

Corollary 14 (Single-index specialized MLP upper bound)**.** *Under Data Setting 2, the target $f^\star$ admits approximation by cluster-specialized two-layer ReLU networks with polynomially growing width and squared Frobenius norm and with approximation error tending to zero. Moreover, the MLP estimator with*

$$m_K = \left\lceil K^{9/8} \right\rceil, \qquad B_K = K^{9/8}, \qquad n_K = \lceil K^7 \rceil$$

is consistent.

Proof. Take $r_0 = 1$ and $R = 1$ in Theorem 13. Then V_c reduces to the single predictive direction v_c, and the multi-index distribution reduces to Data Setting 2. □

B.2.2 The Single-Index Kernel and RFM Lower Bounds

For the remainder of the proof, we return to the single-index setting. Throughout, $d = 2K$ and $R \geq 1$ is fixed. We use

$$\mu_c = [Re_c; 0], \qquad v_c = [0; e_c], \qquad \Sigma_c = \mathrm{diag}(0_{K\times K}, I_K).$$

Equivalently,

$$c \sim \mathrm{Unif}([K]), \qquad x = \mu_c + \begin{bmatrix} 0 \\ z \end{bmatrix}, \qquad z \sim \mathcal{N}(0, I_K),$$

and

$$f^\star(x) = g(\langle x, v_c\rangle),$$

where $g : \mathbb{R} \to \mathbb{R}$ is fixed, nonconstant, bounded, and Lipschitz.

B.2.3 The Ground-Truth AGOP Removes the Routing Block

Define the ground-truth population AGOP across all input coordinates by

$$M := \mathbb{E}\left[\nabla_x f^\star(x) \nabla_x f^\star(x)^\top\right].$$

Lemma 15 (The AGOP kernel loses the cluster identity)**.** *Let $P_V := \sum_{c=1}^K v_c v_c^\top$ be the orthogonal projector onto the predictive subspace, let $G \sim \mathcal{N}(0,1)$, and set*

$$\alpha := \mathbb{E}[g'(G)^2].$$

Then $0 < \alpha \leq L^2$, and

$$M = \frac{\alpha}{K} P_V.$$

Moreover, every measurable function h of $\sqrt{M}x$ satisfies

$$\mathbb{E}\left[\left(h(\sqrt{M}x) - f^\star(x)\right)^2\right] \geq \left(1 - \frac{1}{K}\right) \mathrm{Var}(g(G)).$$

Proof. On the support of cluster c, the regression function is $f^\star(x) = g(\langle x, v_c\rangle)$ and hence, almost everywhere,

$$\nabla_x f^\star(x) = g'(\langle x, v_c\rangle) v_c.$$

Since g is Lipschitz, $|g'| \leq L$ almost everywhere, so $\alpha \leq L^2$. If $\alpha = 0$, then $g' = 0$ almost everywhere with respect to Gaussian measure and therefore almost everywhere with respect to Lebesgue measure. Since a Lipschitz function is absolutely continuous, this would make g constant. Hence $\alpha > 0$.

Averaging the corresponding outer product over the input and the uniform cluster index gives

$$M = \frac{1}{K}\sum_{c=1}^K \mathbb{E}\left[g'(\langle x, v_c\rangle)^2 \mid c\right] v_c v_c^\top = \frac{\alpha}{K}\sum_{c=1}^K v_c v_c^\top = \frac{\alpha}{K} P_V.$$

Therefore $\sqrt{M} = \sqrt{\alpha/K}\, P_V$. In particular, $\sqrt{M}x$ contains the predictive coordinates $S_j := \langle x, v_j\rangle$, $j \in [K]$, but none of the routing coordinates. Under the data distribution, $S = (S_1, \ldots, S_K) \sim \mathcal{N}(0, I_K)$ and is independent of the uniform cluster index c, while $f^\star(x) = g(S_c)$. Thus, conditional on $\sqrt{M}x$,

$$\mathbb{E}[f^\star(x) \mid \sqrt{M}x] = \frac{1}{K}\sum_{j=1}^K g(S_j).$$

The conditional-expectation characterization of squared-loss regression implies that every $h(\sqrt{M}x)$ has risk at least

$$\begin{aligned}\mathbb{E}\Big[\mathrm{Var}\big(f^\star(x) \mid \sqrt{M}x\big)\Big] &= \mathbb{E}\left[\frac{1}{K}\sum_{j=1}^K g(S_j)^2 - \left(\frac{1}{K}\sum_{j=1}^K g(S_j)\right)^2\right] \\ &= \left(1 - \frac{1}{K}\right)\mathrm{Var}(g(G)).\end{aligned}$$

Because g is continuous and nonconstant and the Gaussian distribution has full support, $\mathrm{Var}(g(G)) > 0$.

□

B.2.4 An All-Orders Lower Bound for Kernel Methods and Regularized RFM

We next study the full family in Training Setup 3. Let $P_R := I_d - P_V$ denote the projector onto the routing subspace. By Lemma 15,

$$M_\rho = \rho P_R + \left(\rho + \frac{\alpha}{K}\right) P_V.$$

Thus both the standard metric I_d and the regularized RFM metric M_ρ commute with every rotation of the predictive subspace. The lower bound below is uniform over $\rho \geq 0$, applies to any rotationally invariant base kernel $\mathcal{K}$, and is independent of the empirical ridge parameter.

This degree-by-degree rotational-invariance obstruction is closely related to high-dimensional results showing that rotationally invariant kernel methods with polynomially many samples recover only bounded-degree components (Ghorbani et al., 2021; 2020a). Related group-based analyses of invariant kernel and random-feature models appear in Mei et al. (2021). More broadly, Abbe & Boix-Adsera (2022) develop symmetry-based limitations for equivariant learning algorithms.

Let He_r denote the probabilists' Hermite polynomial and $h_r = \mathrm{He}_r / \sqrt{r!}$ its normalized version. Write

$$g(t) = \sum_{r=0}^{\infty} \widehat{g}_r h_r(t), \qquad \widehat{g}_r = \mathbb{E}[g(G)h_r(G)].$$

Lemma 16 (Bounded nonconstant links have infinite Hermite support)**.** *The Hermite coefficients of g are not eventually zero. Equivalently,*

$$\sum_{r > r_0} \widehat{g}_r^2 > 0 \qquad \textit{for every finite } r_0.$$

Proof. Suppose instead that, for some finite r_0,

$$\widehat{g}_r = 0 \qquad \text{for all } r > r_0.$$

Then, in $L^2(\mathcal{N}(0,1))$,

$$g(t) = \sum_{r=0}^{r_0} \widehat{g}_r h_r(t) =: p(t),$$

where p is a polynomial. Both g and p are continuous, and the Gaussian distribution has full support. Hence equality almost everywhere implies $g(t) = p(t)$ for every $t \in \mathbb{R}$. But g is bounded, so p is a bounded polynomial and must be constant, contradicting the assumption that g is nonconstant. Parseval's identity then gives the stated positive-tail property. □

For $r \geq 2$, let $\mathcal{V}_{r,K}$ be the traceless order-r Gaussian-chaos space. Equivalently, under the standard isometry between the rth Gaussian chaos and symmetric order-r tensors, $\mathcal{V}_{r,K}$ corresponds to the kernel of the tensor trace map. It is the irreducible $O(K)$ representation of harmonic degree-r polynomials.

Lemma 17 (Harmonic dimension and directional energy)**.** *The dimension of $\mathcal{V}_{r,K}$ is*

$$H_{r,K} = \binom{K+r-1}{r} - \binom{K+r-3}{r-2}.$$

Let $P_{r,K}$ be orthogonal projection onto $\mathcal{V}_{r,K}$. For every unit vector $q \in \mathbb{R}^K$,

$$\left\| P_{r,K} h_r(q^\top z) \right\|_{L^2}^2 = \alpha_{r,K},$$

where, writing $r = 2m$ or $r = 2m+1$,

$$\alpha_{2m,K} = \prod_{j=0}^{m-1} \frac{K+2j-1}{K+2m+2j-2},$$

and

$$\alpha_{2m+1,K} = \prod_{j=0}^{m-1} \frac{K+2j-1}{K+2m+2j}.$$

For every fixed r, $H_{r,K} = \Theta_r(K^r)$ and $\alpha_{r,K} \to 1$ as $K \to \infty$.

Proof. The dimension of symmetric order-r tensors is $\binom{K+r-1}{r}$. The trace map onto symmetric order-$(r-2)$ tensors is surjective, and its kernel is the traceless subspace. This gives the stated dimension.

Under the Gaussian-chaos/tensor isometry, $h_r(q^\top z)$ corresponds to $q^{\otimes r}$. By rotational invariance, take $q = e_1$. The orthogonal projection of the homogeneous polynomial x_1^r onto harmonic degree r is

$$\mathcal{H}_r[x_1^r] = \sum_{j=0}^{\lfloor r/2 \rfloor} \frac{(-1)^j \Gamma(r-j+K/2-1)}{4^j j!\,\Gamma(r+K/2-1)} \|x\|^{2j} \Delta^j x_1^r.$$

Since

$$\Delta^j x_1^r = \frac{r!}{(r-2j)!} x_1^{r-2j},$$

the coefficient of x_1^r in this harmonic projection is

$$\sum_{j=0}^{\lfloor r/2 \rfloor} \frac{(-1)^j r!\,\Gamma(r-j+K/2-1)}{4^j j!(r-2j)!\,\Gamma(r+K/2-1)}.$$

In the symmetric-tensor inner product, this coefficient equals $\langle e_1^{\otimes r}, P_{r,K} e_1^{\otimes r} \rangle$, which equals the squared norm of the projection. Simplifying the finite sum gives the two product formulas displayed above. The asymptotic statements follow immediately from those formulas and the dimension expression. □

Lemma 18 (Invariant random-subspace bound)**.** *Let V be a D-dimensional irreducible orthogonal representation of a compact group. Let $S \subseteq V$ be a random subspace with invariant law and $\dim S \le N$ almost surely. Then, for every fixed $v \in V$,*

$$\mathbb{E}\,\mathrm{dist}(v,S)^2 \ge \left(1 - \frac{N}{D}\right)_+ \|v\|_2^2.$$

Proof. Let Π_S be orthogonal projection onto S. Invariance implies that $\mathbb{E}\Pi_S$ commutes with the group action. Schur's lemma gives $\mathbb{E}\Pi_S = \beta I_V$. Taking traces yields $\beta D = \mathbb{E}\dim S \le N$. Therefore

$$\mathbb{E}\,\mathrm{dist}(v,S)^2 = \|v\|_2^2 - \mathbb{E}\|\Pi_S v\|_2^2 = (1-\beta)\|v\|_2^2 \ge \left(1 - \frac{N}{D}\right)_+ \|v\|_2^2.$$

□

Lemma 19 (All-orders lower bound for regularized AGOP kernel methods)**.** *For every rotationally invariant kernel $\mathcal{K}$, every sample size n, every $\rho \ge 0$, and every empirical ridge parameter $\lambda_n \ge 0$, both estimators from Training Setup 3 satisfy*

$$\mathbb{E}\left[\|\widehat{f} - f^\star\|_{L^2(P_x)}^2\right] \ge \sum_{r=2}^{\infty} \widehat{g}_r^2 \alpha_{r,K} \left(1 - \frac{n}{H_{r,K}}\right)_+, \qquad \widehat{f} \in \left\{\widehat{f}_{\mathrm{Kernel}}, \widehat{f}_{\mathrm{RFM}}\right\}.$$

Proof. Let $A = I_d$ for $\widehat{f}_{\mathrm{Kernel}}$ and $A = M_\rho$ for $\widehat{f}_{\mathrm{RFM}}$. By the representer theorem, for a fixed training sample the predictor belongs to the span of the n transformed-kernel sections centered at the training inputs. For a test point in cluster c, restrict these sections to that cluster's support and write them as functions of its predictive argument $\xi \in \mathbb{R}^K$:

$$\psi_{c,i}^A(\xi) := \mathcal{K}\left(\sqrt{A}x_i, \sqrt{A}\left(\mu_c + \begin{bmatrix} 0 \\ \xi \end{bmatrix}\right)\right), \qquad 1 \le i \le n.$$

At Hermite order $r \geq 2$, define

$$S_{c,r} := \operatorname{span}\left\{P_{r,K}\psi^A_{c,i} : 1 \leq i \leq n\right\} \subseteq \mathcal{V}_{r,K}.$$

Then $\dim S_{c,r} \leq n$. The predictive coordinates of the training inputs are standard Gaussian, A commutes with predictive rotations, and $\mathcal{K}$ is rotationally invariant. Hence the law of $S_{c,r}$ is invariant under the $O(K)$ action on $\mathcal{V}_{r,K}$ for both estimators and every $\rho \geq 0$.

The order-r target component in cluster c is

$$\gamma_{c,r} = \widehat{g}_r P_{r,K} h_r(e_c^\top \xi),$$

whose squared norm is $\widehat{g}_r^2 \alpha_{r,K}$ by Lemma 17. The projected predictor lies in $S_{c,r}$. Since distinct Gaussian-chaos orders are orthogonal, for every finite L_0,

$$\|\widehat{f}_c - g(e_c^\top \cdot)\|^2_{L^2} \geq \sum_{r=2}^{L_0} \operatorname{dist}(\gamma_{c,r}, S_{c,r})^2.$$

Apply Lemma 18 and then let $L_0 \to \infty$ by monotone convergence. This gives, for every cluster c,

$$\mathbb{E}\left[\|\widehat{f}_c - g(e_c^\top \cdot)\|^2_{L^2}\right] \geq \sum_{r=2}^{\infty} \widehat{g}_r^2 \alpha_{r,K} \left(1 - \frac{n}{H_{r,K}}\right)_+.$$

Averaging over the uniform test cluster proves the result. □

Lemma 20 (Polynomial sample budgets leave a nonzero kernel/RFM error)**.** *Fix $A < \infty$, let $n_K = O(K^A)$, and let $\rho_K \geq 0$ be any sequence. Then, for each*

$$\widehat{f} \in \left\{\widehat{f}_{\mathrm{Kernel}}, \widehat{f}_{\mathrm{RFM}}\right\},$$

$$\liminf_{K\to\infty} \mathbb{E}\left[\|\widehat{f} - f^\star\|^2_{L^2(P_x)}\right] \geq \sum_{\substack{r\geq 2 \\ r>A}} \widehat{g}_r^2 > 0.$$

Proof. Fix an order $r > A$. By Lemma 17, $H_{r,K} = \Theta_r(K^r)$ and $\alpha_{r,K} \to 1$, so

$$\frac{n_K}{H_{r,K}} \longrightarrow 0.$$

Apply Lemma 19. For any finite collection of orders satisfying $r > A$, each corresponding summand converges to $\widehat{g}_r^2$. Taking the lower limit and then increasing the finite collection gives, by monotone convergence,

$$\liminf_{K\to\infty} \mathbb{E}\left[\|\widehat{f} - f^\star\|^2_{L^2(P_x)}\right] \geq \sum_{\substack{r\geq 2 \\ r>A}} \widehat{g}_r^2.$$

The right-hand side is strictly positive by Lemma 16.

□

Proof of Theorem 2. The specialized MLP approximation and consistency conclusions follow from Corollary 14. In particular, the choices

$$m_K = \left\lceil K^{9/8} \right\rceil, \qquad B_K = K^{9/8}, \qquad n_K = \lceil K^7 \rceil$$

are polynomial in K and give

$$\mathbb{E}\left[\|\widehat{f}_{\mathrm{MLP}} - f^\star\|^2_{L^2(P_x)}\right] \longrightarrow 0.$$

For the lower bound, fix any $A < \infty$, any sequence $n_K = O(K^A)$, and any sequence $\rho_K \geq 0$. Lemma 20 gives

$$\liminf_{K\to\infty} \mathbb{E}\left[\|\widehat{f} - f^\star\|^2_{L^2(P_x)}\right] \geq \sum_{\substack{r\geq 2 \\ r>A}} \widehat{g}_r^2 > 0, \qquad \widehat{f} \in \left\{\widehat{f}_{\mathrm{Kernel}}, \widehat{f}_{\mathrm{RFM}}\right\}.$$

The lower bound is uniform over ρ_K and over the empirical ridge parameters allowed in Training Setup 3. This proves the theorem. □

C Additional Prior Work

C.1 Monosemanticity

Interpretability works on language models have found that some individual neurons are monosemantic, responding primarily to a single interpretable concept or pattern, while others are *polysemantic*, responding to multiple distinct concepts or patterns (Bills et al., 2023; Elhage et al., 2022). More recent work has argued that monosemantic features need not align with individual neurons, and has used sparse dictionary learning to recover more monosemantic feature directions from neural representations (Bricken et al., 2023; Templeton et al., 2024). Our work shows that, at least in simple MLPs, monosemantic features can emerge directly at the level of individual neurons and that this neuron-level specialization can improve sample efficiency compared with methods based on a global low-dimensional representation.

C.2 Mixture-of-Experts Models

A complementary line of work studies specialization and latent-cluster recovery in mixture-of-experts (MoE) architectures. An MoE router can learn cluster-center features that partition a clustered classification problem into simpler subproblems handled by different experts (Chen et al., 2022). Related theoretical and empirical results show that a learned router can route inputs according to latent clusters (Dikkala et al., 2023). Particularly close to our statistical setting, recent work studies nonlinear regression with an underlying cluster structure of single-index models and shows that an MoE trained by SGD can detect the latent organization and divide the task into cluster-specific subproblems; under its assumptions, a vanilla neural network does not detect this organization within the same polynomial complexity regime (Kawata et al., 2025). These works build specialization into the architecture through an explicit router and separate experts. In contrast, we study a standard MLP with no explicit routing or expert decomposition, and show that both the implicit clustering and the corresponding local predictive-feature specialization can emerge through the specialization of individual neurons.