Deep Learning as Neural Low-Degree Filtering: A Spectral Theory of Hierarchical Feature Learning

Yatin Dandi[1,2], **Matteo Vilucchio**[1], **Luca Arnaboldi**[1], **Hugo Tabanelli**[1], **Florent Krzakala**[1]

[1]Information Learning and Physics Laboratory, EPFL, Switzerland
[2]Statistical Physics of Computation Laboratory, EPFL, Switzerland

firstname.lastname@epfl.ch

Abstract

Understanding how deep neural networks learn useful internal representations from data remains a central open problem in the theory of deep learning. We introduce *Neural Low-Degree Filtering* (Neural LoFi), a stylized limit of gradient-based training in which hierarchical feature learning becomes an explicit iterative spectral procedure. In this limit, the dynamics at each layer decouple: given the current representation, the next layer selects directions with maximal accessible low-degree correlation to the label. This yields a tractable surrogate mechanism for deep learning, together with a natural kernel-space interpretation. Neural LoFi provides a mathematically explicit framework for studying multi-layer feature learning beyond the lazy regime. It predicts how representations are selected layer by layer, explains how emergence of concepts arises with given sample complexity, and gives a concrete mechanism by which depth progressively constructs new features from old ones through low-degree compositionality. We complement the theory with mechanistic experiments on fully connected and convolutional architectures, showing that Neural LoFi improves over lazy random-feature baselines, recovers meaningful structured filters, and predicts representations aligned with early gradient-descent feature discovery with real datasets.

1 Introduction

One of the most striking features of modern deep learning [1] is the ability of neural networks to learn complex high-dimensional functions from data with extraordinary effectiveness [2]. Yet this empirical success still outpaces our theoretical understanding. A large body of empirical work suggests that deep networks do not merely fit input-output relations, but progressively build structured representations. In convnets, feature-visualization studies have shown that successive layers extract patterns of increasing complexity, from local edge-like detectors to more semantic motifs [3]. Transfer-learning experiments indicate that earlier layers tend to contain more general features, whereas deeper layers become increasingly specialized to the task [4]. The *platonic representation* viewpoint has emphasized that models often learn surprisingly aligned representations across architectures and training procedures [5]. These observations point to a central challenge:

Can we understand which features are discovered during training, in which order, and at what sample complexity? Can we understand why some functions are learned efficiently with multi-layer models?

Several theoretical perspectives have clarified important parts of this picture. In the lazy regime [6], or equivalently in the Neural Tangent Kernel (NTK) limit [7, 8], training is described as kernel regression in an essentially fixed feature space, giving a powerful theory of optimization and generalization but largely bypassing feature learning. Mean-field analyses of two-layer networks capture genuine parameter evolution and representation learning [9–12], while tensor-program and dynamical mean-field theory approaches provide a general framework for infinite-width limits, including feature-learning regimes under suitable parametrizations [13–16]. In parallel, stylized high-dimensional

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

models, such as single-index and multi-index settings, have made it possible to analyze when neural networks recover latent structure from data [17–20], and recent work has begun to clarify the computational role of depth in synthetic toy models [21–26]. Still, we lack a simple predictive mechanism for how deep networks select, organize, and refine features across layers.

In this work, we propose such a mechanism, which we call *Neural Low-Degree Filtering* (Neural LoFi). Neural LoFi is a stylized, mathematically tractable iterative spectral surrogate for feature learning: given a current representation, each layer forms a label-weighted moment operator on the current features, selects its leading eigendirections, and lifts the resulting projected features through a nonlinear random feature map. This procedure is motivated by a stylized small-initialization limit of gradient-based training, in which the feature-learning component of the layerwise dynamics is governed by a Hessian-like, label-weighted second-order operator.

The analysis of Neural LoFi leads to two main lessons. First, at a fixed representation, feature learning is governed by a *relevance–complexity* trade-off: the next layer selects features that have large low-degree correlation with the label while remaining simple in the geometry induced by the current representation. Neural LoFi makes this trade-off explicit through a variational problem, where relevance is measured by supervised low-degree correlation and complexity by the RKHS norm. This variational view also yields an *explicit, data-driven criterion for feature emergence*: a new direction becomes learnable when its population correlation rises above the empirical noise floor, whose scale is controlled by the residual effective dimension of the current kernel.

Second, depth makes this process powerful because the selected features are lifted into a new representation, changing the geometry in which the next layer searches for signal. Thus a deep network can turn structure that is high-degree in the input into structure that is low-degree in an intermediate representation. This leads to a principle of *low-degree compositionality*: deep learning is efficient when each stage of the target is not only compositional, but visible through low-degree correlations in the current representation. The same viewpoint also gives a kernel interpretation of feature learning: Neural LoFi constructs a sequence of task-adaptive kernels, one per layer, rather than a single fixed kernel chosen before seeing the labels. In this sense, depth allows for an adaptive multilayer kernel construction driven by supervised low-degree feature selection. Although we use Neural LoFi primarily as a theoretical lens, the resulting procedure is also of independent interest: it is layerwise, backpropagation-free, and based only on label-weighted low-degree moments of the current representation. In this sense, it provides a simple spectral primitive for feature discovery, sitting between fixed random-feature models, which do not adapt their representation, and full end-to-end backpropagation, whose dynamics are much harder to analyze.

The rest of the paper develops this picture. We first introduce Neural LoFi and motivate its label-weighted moment operator from the early feature-learning dynamics of gradient descent. We then derive an RKHS variational characterization, which makes explicit the relevance–complexity trade-off solved at each layer. This leads to a criterion for feature emergence, expressed through the residual effective dimension of the current kernel, and to the principle of low-degree compositionality. We conclude with a study of a solvable mathematical model that illustrates out main point (emergence of features and low-degree compositionality), and with real-data experiments, including fully connected and convolutional architectures, illustrating the predicted mechanisms arising in practice. The code for all experiments is available at `https://github.com/IdePHICS/Neural-LoFi-Theory`.

2 Neural Low-Degree Filtering

2.1 Setup and guiding approximation

We now define Neural LoFi and present its feature-space and kernel interpretations. Its derivation as a surrogate for early gradient-based feature learning is given in App. B. Consider supervised data

$$\mathcal{D}_n = \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^n, \qquad \boldsymbol{x}_\mu \in \mathbb{R}^d, \quad y_\mu \in \mathbb{R},$$

with scalar labels for simplicity (The extension to vector-valued outputs is discussed in App. D). We assume throughout this section that the labels are centered, and write $\widehat{\mathbb{E}}_n$ for the empirical average over the training set. A depth-L neural network builds a sequence of representations

$$\boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{x}, \qquad \boldsymbol{h}_\ell(\boldsymbol{x}) = \boldsymbol{W}_\ell \boldsymbol{z}_{\ell-1}(\boldsymbol{x}), \qquad \boldsymbol{z}_\ell(\boldsymbol{x}) \in \mathbb{R}^{p_\ell} = \sigma(\boldsymbol{h}_\ell(\boldsymbol{x})), \qquad \ell = 1, \dots, L, \quad (1)$$

for sequence of widths $p_1, \cdots, p_L$. For structured architectures, such as convolutional networks, the same notation should be read locally: $\boldsymbol{z}_{\ell-1}(\boldsymbol{x})$ is replaced by the local patch or feature vector seen

by a filter, and the corresponding estimator is averaged over spatial locations. The model's output is then defined by a readout applied to the final layer

$$\hat{f}(x) = \langle \boldsymbol{a}_L, \boldsymbol{z}_L(x) \rangle. \tag{2}$$

Fix a layer ℓ, and consider the preactivation $h_{\ell,i}(\boldsymbol{x}) = \langle \boldsymbol{w}_{\ell,i}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$. We show in Appendix B, that under a layer-wise vanishing initialization scheme $a_L \ll W_{L-1} \ll \cdots \ll W_1 \ll 1$, the training of different neurons in a layer, can be described up to leading approximation through a linear dynamics consisting of two terms: (i) a constant drift along a mean direction given by $\hat{\boldsymbol{u}}^\ell \in \mathbb{R}^{p_\ell}$ and (ii), a linear projection along the matrix $\widehat{\boldsymbol{C}}^{(\ell)} \in \mathbb{R}^{p_\ell \times p_\ell}$, with $\hat{\boldsymbol{u}}^\ell, \widehat{\boldsymbol{C}}^{(\ell)}$ defined by:

$$\hat{\boldsymbol{u}}^\ell := \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu), \quad \widehat{\boldsymbol{C}}^{(\ell)} := \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top. \tag{3}$$

This result is formalized below, which is a consequence of Taylor's theorem and the fact that under a layer-wise vanishing scaling of initializations, interactions between neurons within the same layer and between the present layer and the subsequent, untrained layers, are suppressed:

Proposition 1 (Informal)**.** *Suppose that $\hat{f}(x)$ is trained through layer-wise gradient descent on squared loss $\mathcal{L} = \frac{1}{2n} \sum_{\mu=1}^{n} (y_\mu - \hat{f}(\boldsymbol{x}_\mu))^2$ with initialization scale $a_L \ll W_{L-1} \ll \cdots \ll W_1 \ll 1$. Then, for any $\eta > 0$, and layer ℓ, any fixed neuron $w_{\ell,i}$ for $i \in p_\ell$, satisfies for small enough time:*

$$\boldsymbol{w}_{\ell,i}(t+1) - \boldsymbol{w}_{\ell,i}(t) \approx \eta \, c_{0,i} \hat{\boldsymbol{u}}_\ell + \eta \, c_{1,i} \widehat{C}^{(\ell)} \, \boldsymbol{w}_{\ell,i}(t) + O(\left\| \boldsymbol{w}_{\ell,i}(0) \right\|^2), \tag{4}$$

where $c_{0,i}, c_{1,i} \in \mathbb{R}$ are initialization-dependent random coefficients, independent across neurons i.

Appendix I gives a complementary explanation for this suppression. In Proposition 1, higher-order terms are ordered by the layer-wise small-initialization scales, while Appendix I instead isolates the effect of averaging over data: after testing the averaged gradient against a direction, its k-th contribution is a correlation between two degree-k polynomial components, both orthogonal to lower-degree terms. If the current representation has m_ℓ degrees of freedom, this degree-k space has dimension of order m_ℓ^k, so a random initialization direction has typical squared overlap m_ℓ^{-k} with a fixed label-weighted component. Thus higher-order interactions may be present on individual samples but are suppressed in the averaged gradient.

The first term in Eq. 4 is the linear correlation between the current representation and the label. It can be removed by centering, by fitting the best linear predictor in the current representation, or simply interpreted as the degree-one component of the dynamics. The second term, however, is the first genuinely multiple feature-learning component. Let $\widehat{C}^{(\ell)} = \hat{\boldsymbol{V}} \Lambda \hat{\boldsymbol{V}}^\top$ denote the eigendecomposition. For small step-size, the linear term $c_{1,i} \widehat{C}^{(\ell)} \, \boldsymbol{w}_{\ell,i}(t)$ yields the per-neuron approximation $\boldsymbol{w}_{\ell,i}(t) \approx \exp(c_{1,i} \widehat{\boldsymbol{C}}^{(\ell)} t) \, \boldsymbol{w}_{\ell,i}(0)$. Stacking the p_ℓ neuron weights into the row matrix $W_\ell(t) \in \mathbb{R}^{p_\ell \times p_{\ell-1}}$ with i-th row $\boldsymbol{w}_{\ell,i}(t)^\top$, the dynamics splits into two interpretable pieces:

$$W_\ell(t) \;\approx\; W_\ell(0) \, \exp\big(c_1 \, \widehat{\boldsymbol{C}}^{(\ell)} \, t\big) \;=\; \underbrace{W_\ell(0) \, \hat{\boldsymbol{V}}}_{\text{random transformation}} \;\times\; \underbrace{\exp(c_1 \, \Lambda \, t) \, \hat{\boldsymbol{V}}^\top}_{\text{spectral filter} \to \text{top-eigenvector projection}} . \tag{5}$$

The second part depends only on $\widehat{\boldsymbol{C}}^{(\ell)}$: it projects onto the eigen-basis of $\widehat{\boldsymbol{C}}^{(\ell)}$ and reweighs each direction by $\exp(c_1 \lambda_r^{(\ell)} t)$. Since the coefficients $c_{1,i}$ are random and may have either sign, the amplified directions are selected by large *absolute* eigenvalues; this is why Neural LoFi orders the spectrum by $|\lambda_r^{(\ell)}|$. The first part $W_\ell(0) \, \hat{\boldsymbol{V}}$ is t-independent: it is a fixed random linear transformation of those eigenvectors, inherited from the i.i.d. initialization of the neuron weights. The post-training feature map $\sigma\big(W_\ell(t) \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\big)$ at each layer is therefore, to leading order, the composition of (i) a spectral filter that selects the leading eigen-directions of $\widehat{\boldsymbol{C}}^{(\ell)}$ and (ii) a random per-neuron mixing of those directions: small-initialization training acts, to leading order, as a *spectral filtering* of the representation.

While the exponential weighting in Equation 5 holds under vanishing initialization and small time-scales, we expect, in general, the network to maintain a preference towards features corresponding to larger eigenvalue magnitudes for other training regimes. We discuss how weighting regimes arise under different training settings for two-layer networks in App. C. The simplest choice of such

Algorithm 1 Neural Low-Degree Filtering

Require: Dataset $\{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^n$, depth L, ranks $\{k_\ell\}_{\ell=1}^L$, widths $\{p_\ell\}_{\ell=1}^L$

Initialize $\boldsymbol{z}_0(\boldsymbol{x}) \leftarrow \boldsymbol{x}$ and $p_0 = d$.

for $\ell = 1, \dots, L$ **do**

  $\hat{\boldsymbol{u}}^\ell \leftarrow {}^1\!/_n \sum_{\mu=1}^n y_\mu \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)$, $\hat{\boldsymbol{v}}_0^\ell \leftarrow \hat{\boldsymbol{u}}^\ell / \|\hat{\boldsymbol{u}}^\ell\|$ ▷ Estimate linear component, normalize

  Form the label-weighted moment operator:

$$\widehat{\boldsymbol{C}}^{(\ell)} \leftarrow \frac{1}{n} \sum_{\mu=1}^n y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top \in \mathbb{R}^{p_{\ell-1} \times p_{\ell-1}}.$$

  $\widehat{\boldsymbol{V}}_\ell = [\hat{\boldsymbol{v}}_0^\ell, \hat{\boldsymbol{v}}_1^{(\ell)}, \dots, \hat{\boldsymbol{v}}_{k_\ell}^{(\ell)}] \in \mathbb{R}^{p_{\ell-1} \times k_\ell}$, with ordered k_ℓ eigenvectors by decreasing $|\hat{\lambda}|$.

  $\boldsymbol{g}_\ell(\boldsymbol{x}) \leftarrow \widehat{\boldsymbol{V}}_\ell^\top \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \in \mathbb{R}^{k_\ell}$. ▷ Project onto the learned feature subspace

  $\boldsymbol{z}_\ell(\boldsymbol{x}) \leftarrow {}^1\!/_{\sqrt{p_\ell}} \sigma(\boldsymbol{R}_\ell \boldsymbol{g}_\ell(\boldsymbol{x}))$, $\boldsymbol{R}_\ell \in \mathbb{R}^{p_\ell \times k_\ell}$. ▷ Lift by nonlinear random feature map

Fit a final linear or logistic readout $\boldsymbol{a}$ on $\boldsymbol{z}_L(\boldsymbol{x})$.

return Final predictor $\hat{f}(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{z}_L(\boldsymbol{x}) \rangle$

weights is top-k selection by $|\lambda_r^{(\ell)}|$: $w(\lambda_r^{(\ell)}) = 1$ for $r \le k$ after magnitude ordering, and 0 otherwise, which we use in Algorithm 1.

Unlike the general gradient descent dynamics involving complex interactions across layers and neurons, the regime identified by Proposition 1 is *sequential* across layers and *decoupled* across neurons. We call this the *Neural LoFi regime* and put forward the following hypothesis:

*The **Neural LoFi regime** described by Proposition 1 provides a tractable surrogate towards understanding hierarchical learning in deep neural networks.*

2.2 The Neural Low-Degree Filtering algorithm

The decomposition in Eq. (5) directly motivates the two operations of Neural LoFi (Algorithm 1), which replace the implicit dynamics of the Neural LoFi regime (Proposition 1) with two concrete, decoupled steps per layer. First, a *filter* step projects the current representation onto the leading eigen-directions of the label-weighted moment operator $\widehat{\boldsymbol{C}}^{(\ell)}$, making the spectral filter explicit as $\widehat{\boldsymbol{V}}_\ell$. Second, a *lift* step applies a nonlinear random feature map $\boldsymbol{R}_\ell$ that emulates the per-neuron randomness of the left factor $W_\ell(0)\hat{\boldsymbol{V}}$ in Eq. (5), producing the next representation.

The matrix $\widehat{\boldsymbol{V}}_\ell$ defines the *learned feature projection* at layer ℓ. The projections

$$\boldsymbol{g}_\ell(\boldsymbol{x}) = \widehat{\boldsymbol{V}}_\ell^\top \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \,, \tag{6}$$

are the features selected by low-degree filtering. The random nonlinear lift $\sigma(\boldsymbol{R}_\ell \boldsymbol{g}_\ell)$ then re-expands these selected coordinates into a richer representation, so that the next layer can search for new low-degree correlations in the transformed feature space. The projection step can be interpreted as a *weighted principal component analysis*: the linear component $\hat{\boldsymbol{u}}^\ell$ gives the direction in the feature space $\boldsymbol{z}_{\ell-1}(\boldsymbol{x})$ maximizing the linear correlation with y, and the top eigenvectors of $\widehat{\boldsymbol{C}}^{(\ell)}$ extract the most relevant directions[1] in feature space $\boldsymbol{z}_{\ell-1}(\boldsymbol{x})$ weighted by their 2$^{\text{nd}}$-order correlation with y.

Figure 1 gives a first sanity check for Neural LoFi as a surrogate for gradient-based feature learning. We compare it with end-to-end and layerwise GD on the binary CIFAR-10 animal-vs.-vehicle task, for both fully connected and convolutional architectures, with the first layer of the GD network frozen at the Neural LoFi random features. Neural LoFi, a one-pass spectral procedure, matches GD trained for a few hundred steps (per layer), and in the low-data regime of the fully connected network it is ahead of GD; longer GD training overtakes it at large n. We return to these experiments in detail in Section 4.

2.3 Function-space and kernel interpretation

The projection components $[\hat{\boldsymbol{v}}_0^\ell, \hat{\boldsymbol{v}}_1^{(\ell)}, \dots, \hat{\boldsymbol{v}}_{k_\ell}^{(\ell)}]$ reside in the dual space w.r.t. the features $\boldsymbol{z}_{\ell-1}$ and are themselves not directly interpretable and depend on the choice of random projection

[1] In some experiments, we did not even kept the linear component, as the second-order components are numerous and seem to suffice for extracting the most relevant features.

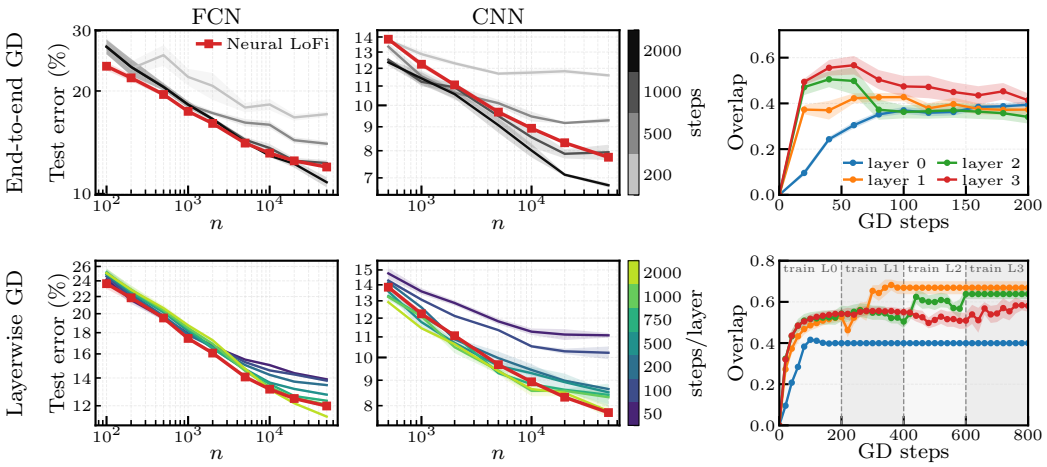


Figure 1: **Neural LoFi versus gradient descent (GD) on binary CIFAR-10.** *Left:* Test error versus training-set size for fully connected and convolutional networks. Red: Neural LoFi. Bands: one standard error over seeds. *Right:* Normalized overlap gain between GD (top-SVD) and LoFi feature subspaces. *Top:* End-to-end GD. *Bottom:* Layerwise GD; shaded intervals on the right mark the layer being trained. Neural LoFi matches early GD performance and captures increasingly aligned feature subspaces. See Appendices K.2–K.3 and L.1 for details.

weights $R_{\ell-1}, \cdots, R_1$. However, we show below that the resulting features $\langle \hat{\boldsymbol{u}}^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$ and $\langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$, for $j = 1, \ldots, k_\ell$, admit a clear description in terms of low-degree projections of y onto the RKHS defined by the previous layer. Furthermore, the features converge to deterministic infinite-width limits. Let

$$K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{z}_{\ell-1}(\boldsymbol{x}), \boldsymbol{z}_{\ell-1}(\boldsymbol{x}') \rangle \tag{7}$$

be the kernel induced by the current representation, and let $\mathcal{H}_{\ell-1}$ be its Reproducing Kernel Hilbert Space (RKHS [27, 28]). Through an analogue of the classical representer theorem (Lemma 1), we show that for any $j \in k_\ell$, the features $\varphi_j(\boldsymbol{x}) := \langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$, lie in the subspace spanned by $\{K_{\ell-1}(\boldsymbol{x}_\mu, \cdot)\}_{\mu=1}^n$, and satisfy the equivalence of norms given by $\|\varphi_j(\boldsymbol{x})\|_{\mathcal{H}_{\ell-1}} = \|\hat{\boldsymbol{v}}_j^{(\ell)}\|_2$. Since $\hat{\boldsymbol{v}}_1^{(\ell)}$ maximizes $\boldsymbol{v}^\top \widehat{\boldsymbol{C}}^{(\ell)} \boldsymbol{v}$ subject to the $\|\boldsymbol{v}\|_2 = 1$, we obtain the equivalent criterion:

$$\hat{\boldsymbol{v}}_1^{(\ell)} \in \underset{\|\boldsymbol{v}\|_2=1}{\arg\max} \left| \frac{1}{n} \sum_{\mu=1}^n y_\mu \langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \rangle^2 \right| \to \varphi_1^{(\ell)} \in \underset{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1}{\arg\max} \left| \frac{1}{n} \sum_{\mu=1}^n y_\mu \, \varphi(\boldsymbol{x}_\mu)^2 \right|. \tag{8}$$

Maximizing this correlation with the label, together with the norm constraint in feature space, leads to the fundamental conclusion:

Neural LoFi searches for features that are **simple** *in the geometry induced by the previous layer, but* **predictive** *through their low-degree correlation with the task.*

This is the sense in which the method is a low-degree filter. We show in App.F, how this can be turned into a practical kernel version of our approach. This is formalized through the following result, which also shows that the selected features converge to the corresponding variational maximizers for the limiting infinite-width kernel as the layer width grows:

Theorem 1 (Variational characterization and infinite-width convergence)**.** *Let $\hat{\boldsymbol{v}}_0^\ell$ denote the linear component and $[\hat{\boldsymbol{v}}_1^{(\ell)}, \ldots, \hat{\boldsymbol{v}}_{k_\ell}^{(\ell)}]$ the ordered eigenvectors from Algorithm 1. Define the linear feature*

$$\psi^\ell(\boldsymbol{x}) := \langle \hat{\boldsymbol{v}}_0^\ell, \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle,$$

and the second-order features $\varphi_j(\boldsymbol{x}) := \langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$ for $j = 1, \ldots, k_\ell$. Then:

(i) ***Linear feature.*** *ψ^ℓ is the unit-norm maximizer of the empirical first-order correlation with y:*

$$\psi^\ell \;=\; \underset{\psi:\, \|\psi\|_{\mathcal{H}_{\ell-1}}=1}{\arg\max} \left| \widehat{\mathbb{E}}_n \left[y \, \psi(\boldsymbol{x}) \right] \right|. \tag{9}$$

(ii) ***Second-order features.*** *For each $k = 1, \dots, k_\ell$, φ_k is the unit-norm maximizer of the empirical second-order correlation, successively orthogonalized to the previously selected features:*

$$\varphi_k \;=\; \underset{\substack{\varphi:\, \|\varphi\|_{\mathcal{H}_{\ell-1}}=1 \\ \varphi \perp \varphi_1, \dots, \varphi_{k-1}}}{\arg\max} \left| \widehat{\mathbb{E}}_n \left[y\, \varphi(\boldsymbol{x})^2 \right] \right|. \tag{10}$$

(iii) ***Infinite-width convergence.*** *Let $K_{\ell-1}^\infty$ denote the infinite-width limiting kernel obtained as the layer width $p_{\ell-1} \to \infty$, and let $\{\phi_k^\infty\}_{k \geq 1}$ be the successive maximizers of the objective in* (10)*, with the RKHS norm and orthogonality induced by $K_{\ell-1}^\infty$. Assume that the eigenvalues associated with the selected maximizers in this limiting label-weighted variational problem are separated from the rest of its spectrum. Then, for any pseudo-lipschitz (of finite order) $\sigma(\cdot)$, and any fixed k, as $p_{\ell-1} \to \infty$,*

$$\varphi_k \;\longrightarrow\; \phi_k^\infty,$$

where convergence is in L^2 under the data distribution.

The above characterization in feature space translates to an explicit recursion over the kernels defined by each layer, once the selected features $\varphi_1^{(\ell)}, \dots, \varphi_{k_\ell}^{(\ell)}$ are obtained, define $\boldsymbol{g}_\ell(\boldsymbol{x}) = \big(\varphi_1^{(\ell)}(\boldsymbol{x}), \dots, \varphi_{k_\ell}^{(\ell)}(\boldsymbol{x})\big)$. If the next layer is a random nonlinear lift, then the induced kernel is

$$K_\ell(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{r}} \left[\sigma(\boldsymbol{r}^\top \boldsymbol{g}_\ell(\boldsymbol{x})) \sigma(\boldsymbol{r}^\top \boldsymbol{g}_\ell(\boldsymbol{x}')) \right]. \tag{11}$$

This recursion is the kernel-level form of feature learning. Unlike a *fixed* kernel method, with a single geometry before seeing the labels, Neural LoFi constructs a sequence of task-adaptive kernels $K_0 \longrightarrow K_1 \longrightarrow \cdots \longrightarrow K_L$, where each transition is supervised: the next kernel is built from the low-complexity features in $\mathcal{H}_{\ell-1}$ whose squared activations are most correlated with the label. Neural LoFi thus turns deep learning into an explicit procedure for adaptive kernel construction. For vector-valued labels, Appendix D describes the analogous construction: the scalar first-order criterion in (9) is replaced by the vector objective $\|\mathbb{E}[\boldsymbol{y}\, \phi(\boldsymbol{x})]\|_2^2$. After orthogonalizing the selected scalar features, this already gives up to c feature directions when $\boldsymbol{y} \in \mathbb{R}^c$; for linear candidates $\phi(\boldsymbol{x}) = \langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$, the criterion is equivalent to SVD on $\mathbb{E}[\boldsymbol{y}\, \boldsymbol{z}_{\ell-1}^\top]$. Appendix H describes additional corrections that account for the covariance of the current feature map, through partial whitening of both the first- and second-order LoFi estimators and a covariance drift in the second-order power iteration.

3 Neural LoFi lessons: emergence and low-degree compositionality

Neural LoFi is not proposed here as a replacement for backpropagation, but as a tractable, simpler, surrogate for understanding learning. This section now attempts to draw lessons from the surrogate.

3.1 Emergence of concepts, effective dimension, and a criterion

A growing body of empirical evidence suggests that learning is often not a smooth process: training dynamics can display long plateaus followed by abrupt transitions, with new directions in representation space *emerging* sequentially rather than all at once [29–32]. Neural LoFi provides a simple mechanism for this phenomenon. By Theorem 1, the kth feature extracted by layer ℓ maximizes the following second-order *empirical* correlation:

$$\widehat{\rho}_\ell^{(k)} := \sup_{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1, \varphi \perp \varphi_1, \cdots, \varphi_{k-1}} \left| \widehat{\mathbb{E}}_n \left[y\, \varphi(\boldsymbol{x})^2 \right] \right|, \tag{12}$$

where $\mathcal{H}_{\ell-1}$ is the RKHS induced by the representation after $\ell - 1$ layers, and $\widehat{\mathbb{E}}_n$ is the empirical average over data. This variational form gives a direct criterion for feature emergence. Suppose that the features $\varphi_1, \cdots, \varphi_{k-1}$ have been extracted and define the set of candidate features as:

$$\mathcal{S}_k^\ell := \left\{ \varphi \in \mathcal{H}_{\ell-1} : \|\varphi\|_{\mathcal{H}_{\ell-1}} = 1,\, \varphi \perp \varphi_1, \cdots, \varphi_{k-1} \right\}. \tag{13}$$

For a fixed unit-norm feature $\varphi \in \mathcal{S}_k^\ell$, define

$$c_\ell(\varphi) := \mathbb{E} \left[y\, \varphi(\boldsymbol{x})^2 \right], \qquad \widehat{c}_{\ell,n}(\varphi) := \widehat{\mathbb{E}}_n \left[y\, \varphi(\boldsymbol{x})^2 \right]. \tag{14}$$

The empirical correlation $\widehat{c}_{\ell,n}(\varphi)$ fluctuates around its population value $c_\ell(\varphi)$. Since Neural LoFi optimizes over a class of candidate features, the relevant measure of variance is the uniform fluctuation over $\mathcal{S}_k^\ell$,

$$\tau_\ell^k(n) \coloneqq \sup_{\varphi \in \mathcal{S}_k^\ell} \left| \widehat{c}_{\ell,n}(\varphi) - c_\ell(\varphi) \right|. \tag{15}$$

To recover the population minimizer feature, the variance must be small w.r.t. the population correlation

$$\rho_\ell^{(k)} \coloneqq \sup_{\varphi \in \mathcal{S}_k^\ell} \left| \mathbb{E}\left[y\, \varphi(\boldsymbol{x})^2 \right] \right|. \tag{16}$$

Hence, the criterion for the emergence of a feature at layer ℓ becomes:

$$\rho_\ell^{(k)} \gg \tau_\ell^k(n). \tag{17}$$

Below this threshold, the leading empirical direction is dominated by noise. Above it, a task-dependent direction separates from the noise and becomes learnable: Feature emergence is thus akin to a phase transition *à la* Baik-Ben Arous-Péché/Edwards-Jones (BBP/EJ) [33, 34]. We show in Appendix G, using local Rademacher-complexity bounds over $\mathcal{S}_k^\ell$ [35,36], that up to poly-logarithmic factors:

$$\tau_\ell^k(n) \lesssim r_\ell^k \sqrt{\frac{D_{\ell,k}^{\mathrm{eff}}(r_\ell^k)}{n}}, \tag{18}$$

where $D_{\ell,k}^{\mathrm{eff}}(r)$ is the *residual effective dimension* of features with scale r (see Def.1) within the current RKHS after removing the previously extracted features [37, 38], and $r_\ell^k \coloneqq \arg\max_r (r\sqrt{D_{\ell,k}^{\mathrm{eff}}(r)})$ is the dominant scale of fluctuations among candidate features. This can be estimated from the empirical spectrum of the kernel, making Eq.(17) a data-driven diagnostic for emergence of concepts.

We develop this formalization of emergence further in Appendix G. The criterion recovers known sample-complexity scales in solvable hierarchical models [39–43]. More importantly, on real data it predicts the sample scale at which learned concepts appear in different layers: in the CIFAR-10 experiment of Section 4, individual eigenvector overlaps rise near the predicted thresholds; see Fig. 4. This makes Neural LoFi a quantitative theory of layerwise concept emergence.

3.2 Why depth helps: low-degree compositionality

While approximation-theoretic works show that deep networks can represent certain compositional functions much more efficiently than shallow ones, especially in tree-like or hierarchical settings [44, 45], efficient representation does not by itself imply efficient learning. The previous discussion motivates the notion of *low-degree compositionality*: depth is useful to learn functions when each layer exposes a new low-degree signal that is visible in the representation constructed by the previous layers. This means that the target admits a sequence of intermediate representations

$$\boldsymbol{x} \;\longrightarrow\; \boldsymbol{z}_1(\boldsymbol{x}) \;\longrightarrow\; \boldsymbol{z}_2(\boldsymbol{x}) \;\longrightarrow\; \cdots \;\longrightarrow\; \boldsymbol{z}_L(\boldsymbol{x}),$$

such that, at each stage, the next useful representation is visible through a low-degree statistic of the current one. In the second-order Neural LoFi mechanism, this is precisely the condition that the population counterpart of Equation (12) is larger than the statistical noise floor. Thus, at layer ℓ, Neural LoFi asks whether the current representation contains a simple feature whose second-order statistics are predictive of the task. This gives a learnability criterion for compositional structure: A deep model does not benefit from *arbitrary compositions*; it benefits from *hierarchies* in which each intermediate step exposes a *statistically detectable low-degree signal*. Equivalently, the target should become progressively simpler when expressed in the learned coordinates.

A function may be high-degree, or otherwise complex, as a function of the original input, while still being learnable through a **sequence of low-degree problems***: Depth then converts one hard estimation problem in the input space into several easier estimation problems in adapted representation spaces.*

In Appendix A.1, we further explain how Neural LoFI connects low-degree compositionality to information exponents and coarse-graining.

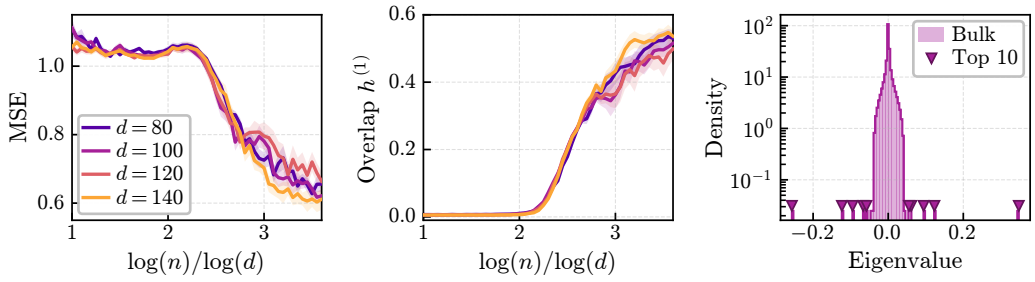


Figure 2: **Feature emergence in a solvable hierarchical model.** *Left:* Test mean-squared error versus sample exponent $\alpha = \log(n)/\log(d)$, for different input dimensions d. The error drops near the predicted recovery scale $n \propto d^{2.5}$. *Center:* Overlap of the recovered first-layer representation with the planted features. *Right:* Spectrum of the first-layer operator $\widehat{C}_1$, showing informative eigenvalues separated from the bulk. See Appendices E.5 and L.1 for the experimental setup and parameters.

4 Mechanistic illustrations: from a solvable model to real data

4.1 A solvable model

We first consider a toy model in which the full Neural LoFi mechanism can be inspected mathematically, that illustrates the salient feature discussed in the former section: *emergence* and *low-degree compositionality*. Following the hierarchical polynomial constructions of [40–43], let $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$ and generate labels through a planted compositional hierarchy $\boldsymbol{x} \longrightarrow \boldsymbol{h}^{(1)}(\boldsymbol{x}) \longrightarrow h^{(2)}(\boldsymbol{x}) \longrightarrow y$. Let $H_k(\boldsymbol{x})$ denote the degree-k Hermite feature vector of the input. We plant $d_1 = d^\varepsilon$ random directions $\boldsymbol{A}_1^{(1)}, \dots, \boldsymbol{A}_{d_1}^{(1)}$ in this degree-k feature space, together with a random symmetric matrix $\boldsymbol{A}^{(2)} \in \mathbb{R}^{d_1 \times d_1}$ acting on the first hidden layer. The hidden variables are

$$h_i^{(1)}(\boldsymbol{x}) = \left\langle \boldsymbol{A}_i^{(1)}, H_k(\boldsymbol{x}) \right\rangle, \qquad i = 1, \dots, d_1, \tag{19}$$

and the second representation is a quadratic function of these variables,

$$h^{(2)}(\boldsymbol{x}) = \left\langle \boldsymbol{A}^{(2)}, H_2(\boldsymbol{h}^{(1)}(\boldsymbol{x})) \right\rangle, \qquad y = g^\star(h^{(2)}(\boldsymbol{x})). \tag{20}$$

While the target is high-degree as a function of the input, it factors into low-degree transitions: $\boldsymbol{h}^{(1)}$ is degree-k in $\boldsymbol{x}$, and $h^{(2)}$ is quadratic in $\boldsymbol{h}^{(1)}$. This is why the model is useful for illustrating both the advantage of feature learning and the advantage of depth: A fixed kernel or random-feature method [46] sees only the composed map $\boldsymbol{x} \mapsto y$; if $g^\star$ contains a degree-r component, then for $k = 2$ this includes degree-$4r$ structure in the input. No learning whatsoever would arise before at least $n = O(d^4)$ data just to beat a random guess!

A two-layer feature learner can already improve on such a one-shot approach by recovering the first hidden representation $\boldsymbol{h}^{(1)}$, but the remaining target is still a nonlinear function of the $d_1 = d^\varepsilon$ latent variables, and thus we still face the curse of dimension (over d_1 instead of d^ε).

This is where the multi-layer approach solves the problem: A second feature-learning step provides the depth advantage: once $\boldsymbol{h}^{(1)}$ is recovered, the next hidden variable $h^{(2)}$ is only a quadratic spectral problem in dimension d_1, followed by a one-dimensional readout. The model realizes the separation identified in [40], but now with Neural LoFi itself: Each layer lowers the effective degree of the remaining task by making the next hidden variable visible through a low-degree statistic. The final target is therefore learned by Neural LoFi progressively, and efficiently, building the correct intermediate representation, rather than by fitting the full high-degree map $\boldsymbol{x} \mapsto y$ in one shot.

We refer to Appendix E for details. The function becomes learnable at $n \gg D_k d_1 = O(d^{k+\varepsilon})$, which, in the quadratic case used in the experiments, is $n \gg d^{2+\varepsilon}$. Around this scale, the Neural LoFi estimator simultaneously shows a drop in prediction error, a sharp increase in overlap with the planted representation, and the separation of the leading eigenvalues from the spectral bulk (see fig. 2 right as well as Fig. 5 in Appendix).

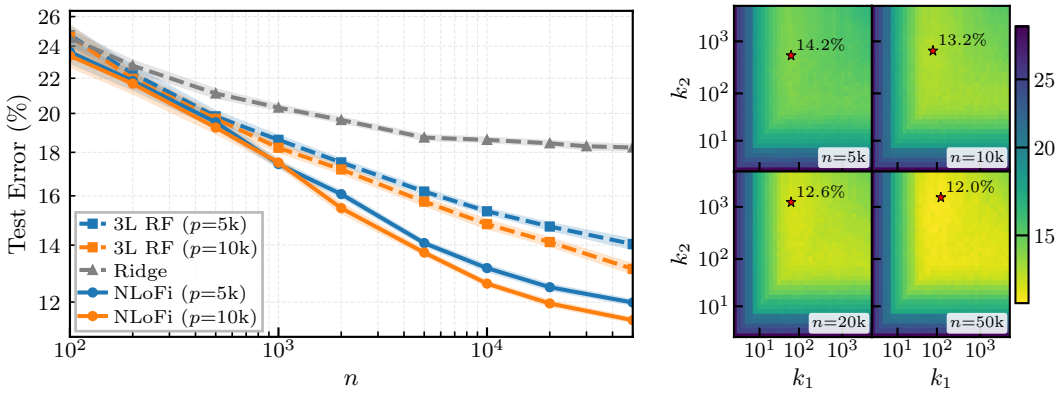


Figure 3: *Fully connected Neural LoFi on the CIFAR-10 animal-vs.-vehicle task. Left:* Test error vs number of training samples for ridge regression, three-layer random features, and Neural LoFi, with projection dimensions $p = 5\text{k}$ and $p = 10\text{k}$. *Right:* Test error over the number of retained features (k_1, k_2) in the first two LoFi layers, for different training-set sizes and fixed projection dimension $p = 5\text{k}$. Stars indicate the best point in each grid.

4.2 Fully connected networks (FCN)

We now focus on experiments on real data, intended as mechanistic illustrations of the theory, not as a claim that Neural LoFi is a competitive training algorithm. Fig.3 (FCN on CIFAR-10 [47]) illustrates two qualitative predictions: i) the label-weighted LoFi operator extracts useful directions beyond those available to fixed random features, leading to a consistent improvement over ridge and random-feature baselines. ii) feature selection is non-trivial: the best test error is not always achieved by keeping the largest number of features. As the sample size grows, the optimal number of retained features increases or remains stable, matching the signal-to-noise picture in which weaker directions become accessible with more data.

Neural LoFi matches GD trained for a few hundred steps. Fig. 1 (right) shows increasing overlap between GD-SVD and LoFi feature subspaces, with staircase-like gains under layerwise training (Appendices K.2–K.3).

Most directly, Fig. 4 tests the feature-emergence criterion of Eqs. (17) and (18). We compare each eigenvector learned from n samples with a large-sample reference eigenvector, and mark the predicted emergence thresholds. The overlap curves rise close to these thresholds, showing that the effective-dimension criterion predicts the sample scale at which individual task-relevant directions ("concepts") become learnable. Additional details on this experiment are given in App. K.5; further numerical evidence, including spectra of the $\widehat{C}$ operator and experiments with the infinite-width Neural LoFi kernel, is reported in App. K.

4.3 Convolutional networks (CNN)

Neural LoFi is not restricted to fully connected architectures: the moment operator can be built in the feature space exposed by the architecture. This makes it natural to incorporate structural inductive biases such as locality, weight sharing, equivariance, or graph structure, in the spirit of geometric deep learning [48]. Here, we consider convolutional networks. Let $\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \in \mathbb{R}^{s_\ell \times p_{\ell-1}}$ denote the input features to layer ℓ, where s_ℓ is the number of spatial locations and $p_{\ell-1}$ is the number of channels. We define the operator in channel space as:

$$\widehat{\boldsymbol{C}}^{(\ell)} := \frac{1}{n s_\ell} \sum_{\mu=1}^{n} \sum_{j=1}^{s_\ell} y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)_j \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)_j^\top . \tag{21}$$

The top eigenvectors of $\widehat{\boldsymbol{C}}^{(\ell)}$ define the task-relevant channel directions retained at layer ℓ; the resulting features are then passed through the same fixed random lifting used by Neural LoFi.

Recall from Fig. 1 that Neural LoFi matches GD trained for a few hundred steps, for the convolutional network as well. The convolutional experiment also gives the most direct visual evidence for low-degree compositionality. Applying the LoFi operator directly to pixels already recovers familiar

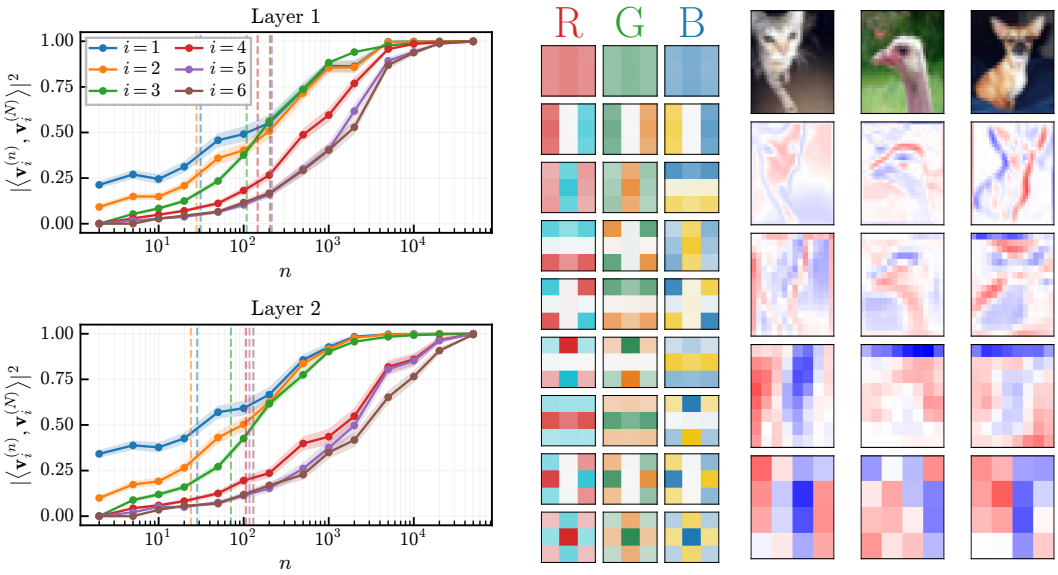


Figure 4: *Left:* **Predicting when individual features emerge on CIFAR-10.** For a three-layer fully connected Neural LoFi model on the CIFAR-10 animal-vs.-vehicle task, we track the squared overlap $|\langle v_i^{(n)}, v_i^{(N)} \rangle|^2$ between eigenvectors estimated from n samples and large-sample reference eigenvectors computed with $n = 60$k samples. Curves show mean $\pm$ SEM over 100 random subsamples at fixed random features. Dashed vertical lines indicate the predicted emergence thresholds n_ℓ^i from (17),(18) (for further details, see Eq. (117) in Appendix). The sharp rise of each overlap near its predicted threshold shows that the effective-dimension criterion predicts when individual task-relevant directions become learnable. *Right:* Neural LoFi with convolutional layers on the CIFAR-10 animal-vs.-vehicle task. Neural LoFi recovers the usual structures and edge detectors associated to GNN *without backpropagation or end-to-end optimization Filters:* Top first-layer filters obtained from the three RGB input channels. *Activations:* Activations of the sixth LoFi feature on test images at successive depths; top row shows the input images, lower rows the corresponding activation maps.

low-level visual features: a leading color-sensitive direction, followed by localized contrast and edge-like filters. Several later filters resemble finite-difference or Laplacian-like stencils, echoing classical edge-filter emergence in natural image models [49] and the early filters learned by CNNs trained with gradient descent [3, 50, 51]. In appendix, we further show in Fig. 10 the feature-emergence criterion of eqs. (17) and (18) for GNN, as we did for FCN in Fig.4.

Strikingly, these salient visual structures emerge *without backpropagation or end-to-end optimization*: they are obtained simply by diagonalizing the Neural LoFi label-weighted moment operator. This shows that part of early visual feature learning is already captured by the low-degree supervised correlations emphasized by our theory. Depth then turns these simple local filters into more selective features. In Figure 4, the same LoFi feature behaves like a generic edge or contrast detector at shallow depth, but becomes increasingly selective after successive LoFi layers, responding only to more structured spatial patterns. This is, again, the qualitative mechanism suggested by the theory: each layer extracts a low-degree task-correlated component from the current representation, and repeated extraction builds progressively more structured features. Additional experiments (including the spectrum of the operators, and different datasets) are presented in App. K.

The deliberate simplicity of Neural LoFi also delineates a clear research program. Theoretically, it remains to understand the long-time dynamics of the task-adaptive kernels generated by LoFi, their behavior beyond second-order filtering, and their extension to structured architectures such as attention. Algorithmically, the main challenge is to scale LoFi from a mechanistic surrogate into a useful training primitive for initialization, feature pretraining, pruning, or diagnostics. Incorporating data reuse, multi-pass dynamics, and backward feature correction would bring the framework closer to trained networks.

Acknowledgments

We would like to thank Bruno Loureiro and Lenka Zdeborová for discussions and encouragement. We acknowledge funding from the Swiss National Science Foundation grants OperaGOST (grant number 200021_200390) and DSGIANGO (grant number 225837). This work was supported by the Simons Collaboration on the Physics of Learning and Neural Computation via the Simons Foundation grant (#1257412). This work was also supported by the EPFL AI Center PhD Fellowship Program 2026.

References

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

[2] Terrence J Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020.

[3] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[4] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[5] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[6] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

[7] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[8] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

[9] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[10] Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

[11] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *stat*, 1050:22, 2018.

[12] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

[13] Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

[14] Karl Hajjar, Lénaïc Chizat, and Christophe Giraud. Training integrable parameterizations of deep neural networks in the infinite-width limit. *Journal of Machine Learning Research*, 25(196):1–130, 2024.

[15] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[16] Rosalba Pacelli, Sebastiano Ariosto, Mauro Pastore, Francesco Ginelli, Marco Gherardi, and Pietro Rotondo. A statistical mechanics framework for bayesian deep neural networks beyond the infinite-width limit. *Nature Machine Intelligence*, 5(12):1497–1507, 2023.

[17] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems*, 33:14820–14830, 2020.

[18] Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. *Advances in neural information processing systems*, 35:9768–9783, 2022.

[19] Alex Damian, Loucas Pillaud-Vivien, Jason Lee, and Joan Bruna. Computational-statistical gaps in gaussian single-index models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1262–1262. PMLR, 2024.

[20] Emanuele Troiani, Yatin Dandi, Leonardo Defilippis, Lenka Zdeborová, Bruno Loureiro, and Florent Krzakala. Fundamental computational limits of weak learnability in high-dimensional multi-index models. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[21] Jerome Garnier-Brun, Marc Mezard, Emanuele Moscato, and Luca Saglietti. How transformers learn structured data: Insights from hierarchical filtering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 18831–18847. PMLR, 13–19 Jul 2025.

[22] Francesco Cagnetta, Leonardo Petrini, Umberto M Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14(3):031001, 2024.

[23] Alessandro Favero, Francesco Cagnetta, and Matthieu Wyart. Locality defeats the curse of dimensionality in convolutional teacher-student scenarios. *Advances in Neural Information Processing Systems*, 34:9456–9467, 2021.

[24] Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. *arXiv preprint arXiv:2502.12089*, 2025.

[25] Yatin Dandi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The computational advantage of depth in learning high-dimensional hierarchical targets. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[26] Yunwei Ren, Yatin Dandi, Florent Krzakala, and Jason D. Lee. Provable learning of random hierarchy models and hierarchical shallow-to-deep chaining. *arXiv preprint arXiv:2601.19756*, 2026.

[27] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

[28] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[29] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[30] Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in neural information processing systems*, 36:14228–14246, 2023.

[31] Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

[32] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581, 2023.

[33] Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of probability*, 33(5):1643–1697, 2005.

[34] Samuel F Edwards and Raymund C Jones. The eigenvalue spectrum of a large symmetric random matrix. *Journal of Physics A: Mathematical and General*, 9(10):1595–1603, 1976.

[35] Shahar Mendelson. On the performance of kernel classes. *Journal of Machine Learning Research*, 4:759–771, 2003.

[36] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[37] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[38] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[39] Leonardo Defilippis, Florent Krzakala, Bruno Loureiro, and Antoine Maillard. Optimal scaling laws in learning hierarchical multi-index models. *arXiv preprint arXiv:2602.05846*, 2026.

[40] Hugo Tabanelli, Yatin Dandi, Luca Pesce, and Florent Krzakala. Deep learning of compositional targets with hierarchical spectral methods. *arXiv preprint arXiv:2602.10867*, 2026.

[41] Eshaan Nichani, Alex Damian, and Jason D. Lee. Provable guarantees for nonlinear feature learning in three-layer neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pages 10828–10875, 2023.

[42] Zihao Wang, Eshaan Nichani, and Jason D. Lee. Learning hierarchical polynomials with three-layer neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Hengyu Fu, Zihao Wang, Eshaan Nichani, and Jason D. Lee. Learning hierarchical polynomials of multiple nonlinear features. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, February 2017.

[45] Matus Telgarsky. Benefits of depth in neural networks. In *Proceedings of the 29th Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539. PMLR, June 2016.

[46] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022.

[47] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[48] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[49] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[50] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[51] 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.

[52] Alaa Saade, Florent Krzakala, and Lenka Zdeborová. Spectral clustering of graphs with the bethe hessian. *Advances in neural information processing systems*, 27, 2014.

[53] Valentina Ros, Gérard Ben Arous, Giulio Biroli, and Chiara Cammarota. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X*, 9(1):011003, 2019.

[54] Stefano Sarao Mannelli, Giulio Biroli, Chiara Cammarota, Florent Krzakala, and Lenka Zdeborová. Who is afraid of big bad minima? analysis of gradient-flow in spiked matrix-tensor models. *Advances in neural information processing systems*, 32, 2019.

[55] Stefano Sarao Mannelli, Giulio Biroli, Chiara Cammarota, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X*, 10(1):011057, 2020.

[56] Yue M Lu and Gen Li. Phase transitions of spectral initialization for high-dimensional non-convex estimation. *Information and Inference: A Journal of the IMA*, 9(3):507–541, 2020.

[57] Marco Mondelli and Andrea Montanari. Fundamental limits of weak recovery with applications to phase retrieval. In *Conference On Learning Theory*, pages 1445–1450. PMLR, 2018.

[58] Antoine Maillard, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Phase retrieval in high dimensions: Statistical and computational phase transitions. *Advances in Neural Information Processing Systems*, 33:11071–11082, 2020.

[59] Matteo Vilucchio, Yatin Dandi, Matéo Pirio Rossignol, Cédric Gerbelot, and Florent Krzakala. Asymptotics of non-convex generalized linear models in high-dimensions: A proof of the replica formula. *arXiv preprint arXiv:2502.20003*, 2025.

[60] Tony Bonnaire, Giulio Biroli, and Chiara Cammarota. The role of the time-dependent hessian in high-dimensional optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2025(8):083401, 2025.

[61] Bohan Zhang, Zihao Wang, Hengyu Fu, and Jason D. Lee. Neural networks learn generic multi-index models near information-theoretic limit. *arXiv preprint arXiv:2511.15120*, 2025.

[62] Andrea Montanari and Zihao Wang. Phase transitions for feature learning in neural networks. *arXiv preprint arXiv:2602.01434*, 2026.

[63] Gérard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research*, 22(106):1–51, 2021.

[64] Emmanuel Abbe, Enric Boix Adserà, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2552–2623. PMLR, 2023.

[65] Alex Damian, Jason D. Lee, and Joan Bruna. The generative leap: Tight sample complexity for efficiently learning gaussian multi-index models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[66] Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *Conference on Learning Theory*, pages 956–1006. PMLR, 2015.

[67] Thibault Lesieur, Léo Miolane, Marc Lelarge, Florent Krzakala, and Lenka Zdeborová. Statistical and computational phase transitions in spiked tensor estimation. In *2017 ieee international symposium on information theory (isit)*, pages 511–515. IEEE, 2017.

[68] Alexander S Wein, Ahmed El Alaoui, and Cristopher Moore. The kikuchi hierarchy and tensor pca. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1446–1468. IEEE, 2019.

[69] Gérard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Algorithmic thresholds for tensor pca. *The Annals of Probability*, 48(4):2052–2087, 2020.

[70] Yatin Dandi, Emanuele Troiani, Luca Arnaboldi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The benefits of reusing batches for gradient descent in two-layer networks: break-

ing the curse of information and leap exponents. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9991–10016, 2024.

[71] Emmanuel Abbe, Enric Boix-Adserà, Matthew S Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems*, 34:26989–27002, 2021.

[72] Gérard Ben Arous, Cédric Gerbelot, and Vanessa Piccolo. Stochastic gradient descent in high dimensions for multi-spiked tensor pca. *arXiv preprint arXiv:2410.18162*, 2024.

[73] Lorenzo Bardone, Sebastian Goldt, et al. Sliding down the stairs: how correlated latent variables accelerate learning with neural networks. In *International Conference on Machine Learning*, volume 235, pages 3024–3045, 2024.

[74] Hugo Tabanelli, Pierre Mergny, Lenka Zdeborová, and Florent Krzakala. Computational thresholds in multi-modal learning via the spiked matrix-tensor model. *arXiv preprint arXiv:2506.02664*, 2025.

[75] Daniel Kunin, Giovanni Luca Marchetti, Feng Chen, Dhruva Karkada, James B. Simon, Michael R. DeWeese, Surya Ganguli, and Nina Miolane. Alternating gradient flows: A theory of feature learning in two-layer neural networks. In *Advances in Neural Information Processing Systems*, 2025.

[76] Giovanni Luca Marchetti, Daniel Kunin, Adele Myers, Francisco Acosta, and Nina Miolane. Sequential group composition: A window into the mechanics of deep learning. In *International Conference on Machine Learning*, 2026.

[77] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

[78] Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20(133):1–34, 2019.

[79] Stefano Sarao Mannelli, Eric Vanden-Eijnden, and Lenka Zdeborová. Optimization and generalization of shallow neural networks with quadratic activation functions. *Advances in Neural Information Processing Systems*, 33:13445–13455, 2020.

[80] Yossi Arjevani, Joan Bruna, Joe Kileel, Elzbieta Polak, and Matthew Trager. Geometry and optimization of shallow polynomial networks. *SIAM Journal on Applied Algebra and Geometry*, 10(2):174–209, 2026.

[81] Vittorio Erba, Emanuele Troiani, Lenka Zdeborová, and Florent Krzakala. The nuclear route: Sharp asymptotics of erm in overparameterized quadratic networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[82] Leonardo Defilippis, Yizhou Xu, Julius Girardin, Vittorio Erba, Emanuele Troiani, Lenka Zdeborová, Bruno Loureiro, and Florent Krzakala. Scaling laws and spectra of shallow neural networks in the feature learning regime. In *The Fourteenth International Conference on Learning Representations*, 2026.

[83] Fabrizio Boncoraglio, Vittorio Erba, Emanuele Troiani, Florent Krzakala, and Lenka Zdeborová. Inductive bias and spectral properties of single-head attention in high dimensions. *arXiv preprint arXiv:2509.24914*, 2025.

[84] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[85] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, et al. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020.

[86] Tanguy Marchand, Misaki Ozawa, Giulio Biroli, and Stéphane Mallat. Multiscale data-driven energy estimation and generation. *Phys. Rev. X*, 13:041038, Nov 2023.

[87] Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *Journal of Machine Learning Research*, 25(350):1–59, 2024.

[88] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[89] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[90] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

[91] Aaditya 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.

[92] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in convolutional neural networks. *arXiv preprint arXiv:2309.00570*, 2023.

[93] Enric Boix-Adserà, Neil Rohit Mallinar, James B. Simon, and Mikhail Belkin. FACT: A first-principles alternative to the neural feature ansatz for how networks learn representations. In *The Fourteenth International Conference on Learning Representations*, 2026.

[94] Taehun Cha, Daniel Beaglehole, Adityanarayanan Radhakrishnan, and Donghun Lee. The weight gram matrix captures sequential feature linearization in deep networks. *arXiv preprint arXiv:2605.06258*, 2026.

[95] Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7:13276, 2016.

[96] Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[97] Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. In *Advances in Neural Information Processing Systems*, volume 33, pages 9346–9360, 2020.

[98] Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, pages 4839–4850, 2019.

[99] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of cnns. In *International Conference on Machine Learning*, pages 736–745, 2020.

[100] Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. In *Machine Learning and Knowledge Discovery in Databases*, pages 498–515. Springer, 2015.

[101] Bernd Illing, Wulfram Gerstner, and Johanni Brea. Biologically plausible deep learning—but how far can we go with shallow networks? *Neural Networks*, 118:90–101, 2019.

[102] Bernd Illing, Jean Ventura, Guillaume Bellec, and Wulfram Gerstner. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. In *Advances in Neural Information Processing Systems*, volume 34, pages 30365–30379, 2021.

[103] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[104] Terence D Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural networks*, 2(6):459–473, 1989.

[105] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

[106] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[107] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[108] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR, 13–18 Jul 2020.

[109] Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. *arXiv preprint arXiv:2602.07488*, 2026.

[110] Kenneth G Wilson. The renormalization group and critical phenomena. *Reviews of Modern Physics*, 55(3):583, 1983.

[111] Sebastian Goldt, Madhu Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborová. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in neural information processing systems*, 32, 2019.

[112] 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.

[113] Alberto Bietti, Joan Bruna, and Loucas Pillaud-Vivien. On learning gaussian multi-index models with gradient flow. *arXiv preprint arXiv:2310.19793*, 2023.

[114] Luca Arnaboldi, Ludovic Stephan, Florent Krzakala, and Bruno Loureiro. From high-dimensional & mean-field dynamics to dimensionless odes: A unifying approach to sgd in two-layers networks. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1199–1227. PMLR, 2023.

[115] Gerard Ben Arous, Murat A Erdogdu, Nuri Mert Vural, and Denny Wu. Learning quadratic neural networks in high dimensions: SGD dynamics and scaling laws. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[116] Francesco Cagnetta, Hyunmo Kang, and Matthieu Wyart. Learning curves theory for hierarchically compositional data with power-law distributed features. In *Forty-second International Conference on Machine Learning*, 2025.

[117] Garrett G. Wen, Hong Hu, Yue M. Lu, Zhou Fan, and Theodor Misiakiewicz. When does gaussian equivalence fail and how to fix it: Non-universal behavior of random features with quadratic scaling. *arXiv preprint arXiv:2512.03325*, 2025.

[118] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[119] James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A*, 209:415–446, 1909.

[120] Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[121] Daniel Beaglehole, Peter Súkeník, Marco Mondelli, and Mikhail Belkin. Average gradient outer product as a mechanism for deep neural collapse. *Advances in Neural Information Processing Systems*, 37:130764–130796, 2024.

[122] Hong Hu, Yue M Lu, and Theodor Misiakiewicz. Asymptotics of random feature regression beyond the linear scaling regime. *arXiv preprint arXiv:2403.08160*, 2024.

[123] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017.

[124] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[125] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[126] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[127] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[128] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A Further related works

Hessian and spectral learning —- Hessian-based spectral information has long played a central role in statistical physics and high-dimensional inference, for instance in community detection [52], spiked matrix–tensor models [53–55], and BBP/EJ-type selection mechanisms [33, 34]. Related label-weighted second-order operators also appear in single-index and multi-index estimation, where they yield spectral initializations and recovery procedures [56–59]. More recently, similar operators have been connected to the early dynamics of shallow neural networks through Hessian or Hessian-like interpretations [39, 60–62]. Neural LoFi extends this viewpoint beyond the first initial layer and to iterative multi-layer methods.

Beyond second order — While the second-order operator (3) is the first nontrivial term in the expansion, higher-order terms correspond to *tensor* correlations between the current representation and the label. In Gaussian single-index and multi-index models, such higher-degree components govern harder directions and later learning time scales, as captured by information, leap, and generative exponents [19, 20, 63–65]. Higher-order LoFi variants could replace $y\varphi^2$ by higher-degree statistics, but would lead to (difficult) tensor spectral problems, as tensor PCA can be notably harder than matrix PCA [66–69]. This limitation also points to a natural future direction: Multi-pass gradient descent can break the constraints imposed by information and leap exponents by repeatedly transforming the effective label and representation [70] as well as through staircase mechanisms [54, 71–74]. An interesting direction for future work is a multi-pass Neural LoFi, alternating low-degree spectral filtering with target/residual transformations and feature correction, that would connect the present mechanism to the more powerful generative-exponent/SQ-type learnability picture. We further discuss connections with this theoretical literature in App.C.

Recent alternating-gradient-flow analyses give a complementary dynamical account of feature learning from small initialization, in which training proceeds through plateaus and sharp drops as features activate sequentially [75]. Related work on sequential group composition studies how networks learn structured compositional tasks one representation-theoretic component at a time, and how depth can exploit associativity to improve scaling [76]. These works are closely aligned with the goal of understanding the order in which features emerge under gradient descent; Neural LoFi instead isolates a layerwise spectral primitive that predicts which low-degree directions become available at each representation.

Quadratic networks and spectral feature learning — A closely related line of work studies quadratic and polynomial networks, where the feature-learning component of the dynamics is especially transparent. Early works analyzed the optimization landscape and recovery properties of one-hidden-layer or polynomial networks through tensor and low-rank structure [77–80]. More recent high-dimensional analyses of overparameterized quadratic networks show that ERM can be mapped to matrix sensing with nuclear-norm regularization, so that capacity control emerges through low-rank learned feature maps [81]. Even closer to our perspective, [82] characterize the spectra of trained quadratic and diagonal networks in the feature-learning regime: informative directions appear as spectral outliers or spikes, while the bulk corresponds to learned noise or overfitting and should be pruned. A related bulk-plus-spikes picture was later observed in attention models [83]. This mirrors the Neural LoFi mechanism: the Hessian-like label-weighted operator extracts task-correlated spectral directions and discards the residual bulk. This viewpoint is also consistent with empirical observations that trained deep networks often exhibit structured, non-random spectra in their learned weights with eigenvalues popping out of the bulk [84].

Scattering, coarse-graining, and multiscale representations — Our construction is related in spirit to scattering transforms [85]. Both scattering and Neural LoFi build representations through a multilevel cascade of filtering and nonlinear lifting. The key difference is that scattering uses fixed analytic filters, typically wavelets, designed for invariance and stability, whereas Neural LoFi is supervised and task-adaptive: each layer selects directions through a label-weighted spectral operator on the current representation. This also resonates with coarse-graining and renormalization-inspired views of deep representations [86].

Rainbow networks — The rainbow analysis of [87] gives a post-training description of deep nets: trained layers behave like random feature maps with learned, often low-rank, weight covariances, yielding deterministic hierarchical kernels in the infinite-width limit. In this view, feature learning amounts to identifying low-dimensional covariance structure between random high-dimensional embeddings. Neural LoFi provides a possible mechanism for the emergence of this structure: its

label-weighted spectral operator selects the task-correlated directions to keep, while the subsequent random lift re-expands them into a new feature space. This is precisely the picture advocated in [87].

Mechanistic interpretability — Neural LoFi is complementary to mechanistic interpretability, which aims to reverse-engineer trained networks into interpretable features and circuits [88, 89]. Rather than starting from a trained model and asking which circuits implement a behavior, Neural LoFi asks why certain features are selected during training. Its basic objects are spectral directions in representation space, not necessarily individual neurons, aligning with the modern "features as directions" viewpoint underlying superposition [90].

Recursive Feature Machines — A recent line of work proposed the *average gradient outer product* (AGOP) as a mechanism for feature learning and uses it to define Recursive Feature Machines [51,91]. Closely related convolutional variants were shown to recover edge-detector-like filters and deep visual features through AGOP-based updates [92], and recent works further connect neural feature matrices, AGOP-type objects, and weight-Gram dynamics to gradient descent and convergence [93, 94]. This line of work is close in philosophy to Neural LoFi: both seek a simple iterative surrogate for representation learning. The constructions differ, however, in where the supervised spectral selection occurs. AGOP/RFM methods first fit a predictor and then update the feature metric using the gradient covariance of that predictor, whereas Neural LoFi applies a label-weighted spectral filter directly to the current representation before the nonlinear lift.

Appendix J explains the corresponding limitation of AGOP/RFM. For the hierarchical target, fixed-kernel KRR requires the degree-$2q$ scale $\Theta(d^{2q})$ before its prediction error improves. For RFM, feature recovery is measured by the fraction of AGOP-transformed feature energy assigned to the planted subspace. In the quadratic multi-index model this fraction vanishes below the degree-two dimension $D_2 = \Theta(d^2)$. The same mechanism suggests a d^{2q} bottleneck for Deep RFM on the hierarchical model. Neural LoFi, meanwhile, provably recovers the hidden representation at the scale $D_q d_1 = \Theta(d^{q+\epsilon})$ established in Appendix E.

Local and backpropagation-free learning — In spirit, Neural LoFi is also related to local alternatives to backpropagation, including feedback alignment and direct feedback alignment [95–97], local-loss and layerwise training methods [98, 99], target propagation [100], and biologically inspired plasticity rules [101, 102]. Although Neural LoFi is not proposed as a biologically faithful model, it is backpropagation-free and layerwise: its spectral operator can be approximated by label-modulated Hebbian/Oja-style updates [102–105].

Pruning — Neural LoFi also gives a feature-space perspective on why overparameterization and pruning are not contradictory. Classical pruning methods show that many weights or connections can be removed after training with little loss in performance [106, 107], while the lottery-ticket hypothesis suggests that large dense networks may contain smaller trainable subnetworks [108]. In Neural LoFi, width and rank play complementary roles: large width creates a rich feature space in which task-correlated directions can be discovered, while pruning retains only the directions that finite data can reliably support. Thus large networks are useful for discovery, but low-rank feature selection controls the effective dimension used for prediction.

A.1 Low-degree compositionality and information exponents

This perspective is consistent with a growing body of theoretical work on idealized models, where compositional structure leads to sequential feature recovery and sample-complexity gains from depth, see: [21–26,40,109]. Neural LoFi abstracts a common principle from these settings: a useful hierarchy is one whose next layer is statistically visible through low-degree correlations in the representation already learned. This viewpoint is in particular related to the *compositional information exponent* introduced for hierarchical Gaussian targets [25]. In that setting, one assumes access to a planted hierarchy of intermediate features $\boldsymbol{h}_\ell^\star(\boldsymbol{x})$ and asks for the smallest degree q such that

$$\left\| \mathbb{E}\left[(\boldsymbol{h}_\ell^\star(\boldsymbol{x}))^{\otimes q} f^\star(\boldsymbol{x}) \right] \right\|_F = \Theta(1). \tag{22}$$

A low compositional exponent means that the target has a strong low-degree dependence on the hidden intermediate representation. Instead of assuming access to $\boldsymbol{h}_\ell^\star$, Neural LoFi searches over the current learned feature space through Eq. (12), and acts as a data-adaptive compositional information test by asking whether the current representation contains simple, statistically visible features that are predictive of the task. In this sense, Neural LoFi implements a supervised coarse-graining

procedure: Each layer keeps a small number of task-relevant directions and discards much of the irrelevant high-dimensional variation, in a way reminiscent of physics renormalization-inspired views of learning across scales [110].

B Neural LoFi from Gradient Descent

In this section, our goal is to prove Proposition 1, whose formal version is stated below as Theorem 2, i.e. show that under layerwise training with vanishing initialization, the early-time dynamics of each layer produces spikes along the linear direction $\hat{\boldsymbol{u}}_\ell$ and aligns with the top eigenvectors of $\widehat{C}^{(\ell)}$, for a standard fully-connected architecture without skip connections.

Consider the standard L-layer network

$$\hat{f}(\boldsymbol{x}) = \langle a_L, \boldsymbol{z}_L(\boldsymbol{x})\rangle, \qquad \boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{x}, \qquad z_\ell(x) = \sigma\big(W_\ell \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\big), \quad \ell = 1, \dots, L, \tag{23}$$

with square loss $\mathcal{L} = \frac{1}{2n}\sum_\mu (y_\mu - \hat{f}(\boldsymbol{x}_\mu))^2$. The final readout $a_L \in \mathbb{R}^{p_L}$ is fixed throughout training; the weight matrices W_ℓ are updated layer-wise.

We write $w_{\ell,i}$ for the i-th row of W_ℓ and set

$$\alpha_m := \max_i \|\boldsymbol{w}_{m,i}(0)\|, \qquad m = 1, \dots, L, \tag{24}$$

with $\alpha_{L+1} := \|a_L\|$.

Assumption 1 (Layerwise scale separation)**.** *The dataset, architecture, and activation are fixed. The scales, including the final readout scale $\alpha_{L+1} = \|a_L\|$, form a strict hierarchy: for every ℓ and every $m \in \{\ell+1, \dots, L+1\}$,*

$$\alpha_m = o(\alpha_\ell). \tag{25}$$

Thus later hidden layers, and the final readout, vanish on a strictly smaller asymptotic scale than the layer currently being trained. The hierarchy is used below in a scale-counting sense: every replacement of a later-layer quantity by its leading initialization-scale term carries at least one additional factor from some α_m with $m > \ell$.

Assumption 2. *$\sigma : \mathbb{R} \to \mathbb{R}$ is three times continuously differentiable, $\sigma(0) = 0$, and $\|\sigma'\|_\infty, \|\sigma''\|_\infty, \|\sigma'''\|_\infty < \infty$.*

Define the *linear spike direction* and the *label-weighted covariance* at layer ℓ:

$$\hat{\boldsymbol{u}}_\ell := \frac{1}{n}\sum_{\mu=1}^n y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu), \qquad \widehat{C}^{(\ell)} := \frac{1}{n}\sum_{\mu=1}^n y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top. \tag{26}$$

Theorem 2 (Layerwise GD approximation)**.** *Under Assumptions 1–2, there exist constants $\tau, C > 0$ such that the layerwise gradient descent run admits horizons*

$$T_\ell := \left\lfloor \frac{\tau \alpha_\ell}{\eta} \right\rfloor, \qquad \ell = 1, \dots, L, \tag{27}$$

with the following property: for every layer ℓ, every neuron i, and every $0 \le t \le T_\ell$,

$$\|w_{\ell,i}(t)\| \le C\alpha_\ell.$$

Thus the layer receives an $O(\alpha_\ell)$ update while staying on its initialization scale. Define the leading effective readout

$$\bar{a}_{\ell,i} := c_0^{L-\ell}\Big[\Big(\prod_{m=\ell+1}^{L} W_m(0)\Big)^\top a_L\Big]_i \tag{28}$$

where $c_0 = \sigma'(0)$. Suppose that the effective readout for neuron i is non-degenerate i.e.:

$$|\bar{a}_{\ell,i}| \ge c_{\mathrm{rd}}\alpha_{L+1}\prod_{m=\ell+1}^{L}\alpha_m \tag{29}$$

for a fixed $c_{\mathrm{rd}} > 0$. For such neurons and $0 \le t < T_\ell$,

$$w_{\ell,i}(t+1) - w_{\ell,i}(t) = \eta\, c_0\, \bar{a}_{\ell,i}\, \hat{\boldsymbol{u}}_\ell + \eta\, \bar{a}_{\ell,i}\, c_1\, \widehat{C}^{(\ell)}\, w_{\ell,i}(t) + R_{\ell,i}(t), \tag{30}$$

where $c_1 = \sigma''(0)$. Equivalently, the two leading coefficients are $c_{0,\ell i} := c_0 \bar{a}_{\ell,i}$ and $c_{1,\ell i} := c_1 \bar{a}_{\ell,i}$. These initialization-dependent coefficients are independent across neurons i. For every $0 \le t \le T_\ell$ the accumulated residual obeys

$$\left\| \sum_{s=0}^{t-1} R_{\ell,i}(s) \right\| \le C\alpha_\ell^3. \tag{31}$$

Equivalently, the spike contributes $O(\alpha_\ell)$ over the horizon, the covariance-amplification term contributes $O(\alpha_\ell^2)$, and the discarded part is $O(\alpha_\ell^3)$. Thus the displayed approximation keeps the first two nontrivial orders in the layer-ℓ initialization scale.

Remark 1. *The three components in Equation* (30) *have the following roles:*

- ***Linear spike.*** *$\eta c_0 \bar{a}_{\ell,i} \hat{\boldsymbol{u}}_\ell$ is independent of the current weight; over the horizon T_ℓ it moves the neuron by $O(\alpha_\ell)$ toward the empirical label-feature correlation.*
- ***Covariance amplification.*** *$\eta \bar{a}_{\ell,i} c_1 \widehat{C}^{(\ell)} w_{\ell,i}(t)$ is linear in the current weight and amplifies components aligned with the leading eigenvectors of $\widehat{C}^{(\ell)}$.*
- ***Remainder.*** *The per-step remainder is second order in the current layer scale, hence it accumulates to $O(\alpha_\ell^3)$ over $T_\ell \asymp \alpha_\ell/\eta$ steps. Terms coming from later layers are smaller by the scale hierarchy in Assumption 1.*

The proof below shows that $\bar{a}_{\ell,i}$ is the leading, sample-independent backpropagation coefficient along the path from layer ℓ to the readout. The non-degeneracy condition only excludes neurons whose leading backpropagation coefficient is accidentally cancelled; such neurons are inactive at the displayed order.

B.1 Proof of Theorem 2

Proof. All constants below may depend on the fixed data and architecture bounds, L and the derivative bounds of σ. We write K for a finite constant of this kind, changing from line to line, and do not track separate data, width, or label constants.

For the standard architecture the gradient of the loss w.r.t. $w_{\ell,i}$ involves a single backpropagation path through all layers $\ell+1, \dots, L$. Define the layer-to-layer Jacobian

$$\boldsymbol{J}_{L\ell}(x) := \frac{\partial z_L(x)}{\partial z_\ell(x)} = \prod_{m=\ell+1}^{L} \operatorname{diag}\big(\sigma'(\boldsymbol{h}_m(x))\big) W_m, \quad \boldsymbol{h}_m(x) = W_m z_{m-1}(x), \tag{32}$$

and the sample-dependent effective readout

$$\bar{a}_{\ell,i}(x) := [\boldsymbol{J}_{L\ell}(x)^\top a_L]_i.$$

By the chain rule,

$$\nabla_{w_{\ell,i}} \mathcal{L} = -\frac{1}{n} \sum_{\mu=1}^{n} r_\mu \, \bar{a}_{\ell,i}(\boldsymbol{x}_\mu) \, \sigma'\big(u_{\ell,i}(\boldsymbol{x}_\mu)\big) \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu), \quad r_\mu := y_\mu - \hat{f}(\boldsymbol{x}_\mu), \tag{33}$$

where $u_{\ell,i}(x) = \langle w_{\ell,i}, \boldsymbol{z}_{\ell-1}(x) \rangle$.

The key observation is that, under Assumption 1, the activation derivatives along this path are well approximated by their value at zero, so $\bar{a}_{\ell,i}(x)$ is approximated by the initialization-dependent coefficient in (28). On the layerwise horizons considered below the representations are uniformly bounded and $\|W_m\|_{\mathrm{op}} \lesssim \alpha_m$. Hence $\|\boldsymbol{h}_m(x)\| \lesssim \alpha_m$, and Taylor expanding σ' around zero gives

$$\operatorname{diag}(\sigma'(\boldsymbol{h}_m(x))) = c_0 I + c_1 \operatorname{diag}(\boldsymbol{h}_m(x)) + O(\alpha_m^2), \quad c_0 = \sigma'(0), \; c_1 = \sigma''(0). \tag{34}$$

Let

$$\boldsymbol{J}_{L\ell}^{(0)} := c_0^{L-\ell} \prod_{m=\ell+1}^{L} W_m(0)$$

with the convention that the empty product is the identity. Expanding the product for $\boldsymbol{J}_{L\ell}(x)$ around this leading term gives the explicit scale bound

$$\|\boldsymbol{J}_{L\ell}(x) - \boldsymbol{J}_{L\ell}^{(0)}\|_{\mathrm{op}} \le K \left(\prod_{m=\ell+1}^{L} \alpha_m \right) \left(\sum_{m=\ell+1}^{L} \alpha_m \right).$$

Since each $\alpha_m = o(\alpha_\ell)$ for $m > \ell$, the right-hand side is $o(\alpha_\ell)$. Therefore

$$\|\boldsymbol{J}_{L\ell}(x) - \boldsymbol{J}_{L\ell}^{(0)}\|_{\mathrm{op}} = o(\alpha_\ell), \tag{35}$$

and when $\ell = L$ the left-hand side is zero. Multiplying by the final readout gives, with $\gamma_\ell = \alpha_{L+1} \prod_{m=\ell+1}^{L} \alpha_m$,

$$|\bar{a}_{\ell,i}(x) - \bar{a}_{\ell,i}| \le K\gamma_\ell \left(\sum_{m=\ell+1}^{L} \alpha_m \right),$$

while (29) gives the relative estimate

$$\frac{|\bar{a}_{\ell,i}(x) - \bar{a}_{\ell,i}|}{|\bar{a}_{\ell,i}|} \le C \sum_{m=\ell+1}^{L} \alpha_m = o(\alpha_\ell). \tag{36}$$

In particular, $\bar{a}_{\ell,i}(x) = \bar{a}_{\ell,i}(1 + \varepsilon_{\ell,i}(x))$ with $\sup_x |\varepsilon_{\ell,i}(x)| = o(\alpha_\ell)$. The error is zero when $\ell = L$.

We next translate the above bound to the required bound on the dynamics of W_ℓ. We start by stating uniform bounds required throughout. Since the sample, widths, and activation are fixed, and since we may restrict to scales $\max_m \alpha_m \le 1$, the row-norm condition $\|w_{j,i}\| \le 2\alpha_j$ implies recursively that

$$\max_{\mu,j} \|z_j(\boldsymbol{x}_\mu)\| \le K.$$

Hence, at the start of layer ℓ,

$$\|\hat{\boldsymbol{u}}_\ell\| \le \frac{1}{n} \sum_\mu |y_\mu| \, \|\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\| \le K,$$

and

$$\|\widehat{C}^{(\ell)}\|_{\mathrm{op}} \le \frac{1}{n} \sum_\mu |y_\mu| \, \|\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\|^2 \le K.$$

The same bounds give $\|\boldsymbol{J}_{L\ell}^{(0)}\|_{\mathrm{op}} \le K$ and hence uniformly bounded effective readouts. The residuals r_μ are uniformly bounded as well. Therefore $\|\nabla_{w_{\ell,i}} \mathcal{L}\| \le K$. Choose τ small enough that $\tau K \le 1$. Then, for every t with $\eta t \le \tau \alpha_\ell$,

$$\|w_{\ell,i}(t)\| \le \|w_{\ell,i}(0)\| + \eta t K \le 2\alpha_\ell.$$

On this horizon, $|u_{\ell,i}(\boldsymbol{x}_\mu)| \le K\alpha_\ell$. Taylor's theorem and Assumption 2 give

$$\sigma'(u) = c_0 + c_1\, u + \omega_\sigma(u), \quad |\omega_\sigma(u)| \le \tfrac{1}{2}\|\sigma'''\|_\infty u^2. \tag{37}$$

Moreover, by (36), $\bar{a}_{\ell,i}(\boldsymbol{x}_\mu) = \bar{a}_{\ell,i}(1 + \varepsilon_{\ell,i}(\boldsymbol{x}_\mu))$ with $\sup_\mu |\varepsilon_{\ell,i}(\boldsymbol{x}_\mu)| = o(\alpha_\ell)$. Thus the readout replacement error is smaller than the leading readout contribution itself; quantitatively it is $|\bar{a}_{\ell,i}|\, o(\alpha_\ell) = o(\alpha_\ell^2)$, because $|\bar{a}_{\ell,i}| = O(\gamma_\ell) = o(\alpha_\ell)$. By the hierarchy of scales, we further have $|\hat{f}(\boldsymbol{x}_\mu)| = o(\alpha_\ell^2)$, and $r_\mu = y_\mu + o(\alpha_\ell^2)$. Substituting these three estimates in (33) yields

$$\frac{1}{n} \sum_{\mu=1}^{n} r_\mu\, \bar{a}_{\ell,i}(\boldsymbol{x}_\mu)\, \sigma'\big(u_{\ell,i}(\boldsymbol{x}_\mu)\big)\, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) = \bar{a}_{\ell,i}\big[c_0 \hat{\boldsymbol{u}}_\ell + c_1 \widehat{C}^{(\ell)} w_{\ell,i}(t)\big] + \mathcal{E}_{\ell,i}(t),$$

so the two leading scalar coefficients in the main-text update (4) of Proposition 1 are $c_{0,\ell i} = c_0 \bar{a}_{\ell,i}$ and $c_{1,\ell i} = c_1 \bar{a}_{\ell,i}$. To see the independence, condition on $W_{\ell+2}(0), \ldots, W_L(0)$ and a_L. Then $\bar{a}_{\ell,i}$ is the inner product between the i-th column of $W_{\ell+1}(0)$ and a fixed downstream vector, up to the common scalar $c_0^{L-\ell}$. The columns of $W_{\ell+1}(0)$ are independent at initialization, so the coefficients are independent across i at leading order with the per-step error bound

$$\|\mathcal{E}_{\ell,i}(t)\| \le K\Big(|\bar{a}_{\ell,i}| \sup_\mu |\varepsilon_{\ell,i}(\boldsymbol{x}_\mu)| + |\bar{a}_{\ell,i}|\alpha_\ell^2 + o(\alpha_\ell^2) \Big) \le K\alpha_\ell^2.$$

Finally, $w_{\ell,i}(t+1) - w_{\ell,i}(t) = -\eta \nabla_{w_{\ell,i}} \mathcal{L}$, so setting $R_{\ell,i}(t) := \eta \mathcal{E}_{\ell,i}(t)$ proves (30). Summing over $0 \le s < t \le T_\ell$ gives

$$\left\| \sum_{s=0}^{t-1} R_{\ell,i}(s) \right\| \le \eta t\, K\alpha_\ell^2 \le K\alpha_\ell^3,$$

which is (31). □

C Two-layer networks

C.1 Training a two-layer network on multi-index models

In this section, we instantiate the general framework on a canonical setting that has driven much of the recent theory of feature learning: *training a two-layer network on a Gaussian multi-index model* (e.g. [18, 19, 64, 65, 70–72, 111–114].)

The goal of this subsection is to make explicit how the second-order correlation criterion that drives Neural LoFi in the main text reduces, in this setting, to the classical *information exponent* of [72], and how the regime $\mathrm{IE} \le 2$ is precisely the regime in which Neural LoFi alone (i.e., a single low-degree filtering step) suffices for weak recovery of the hidden subspace. This is connected to a recent line of work on the Hessian of the loss in, e.g. [39,60–62]. We then describe how, when $\mathrm{IE} = 2$, the resulting dynamics of the two-layer network under gradient descent take the form of a *saddle-to-saddle* cascade through the top directions of the population correlation matrix.

Setting We consider inputs $\boldsymbol{x} \in \mathbb{R}^d$ with $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$ and a teacher of *multi-index* form

$$y \;=\; f^\star(\boldsymbol{x}) \;=\; g^\star\big(\langle \boldsymbol{u}_1^\star, \boldsymbol{x}\rangle, \dots, \langle \boldsymbol{u}_r^\star, \boldsymbol{x}\rangle\big) + \xi, \tag{38}$$

where $U^\star = [\boldsymbol{u}_1^\star, \dots, \boldsymbol{u}_r^\star] \in \mathbb{R}^{d\times r}$ has orthonormal columns spanning a hidden subspace $V^\star \subset \mathbb{R}^d$ of dimension $r = \mathcal{O}(1)$, $g^\star : \mathbb{R}^r \to \mathbb{R}$ is a fixed link function with $\mathbb{E}[g^\star(Z)^2] < \infty$ for $Z \sim \mathcal{N}(0, I_r)$, and ξ is independent sub-Gaussian noise. The student is the two-layer network

$$\hat{f}(\boldsymbol{x}) \;=\; \frac{1}{\sqrt{p}} \sum_{i=1}^{p} a_i\, \sigma\big(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle\big), \qquad \boldsymbol{w}_i \in \mathbb{R}^d,\; a_i \in \mathbb{R}, \tag{39}$$

trained by (online or layerwise) gradient descent on the squared loss with small initialization $\boldsymbol{w}_i(0) \sim \mathcal{N}(0, d^{-1} I_d)$ and second-layer weights a_i either fixed at random signs or trained on a fresh batch. This is the standard setup of e.g. [64, 70, 72, 111, 113, 114].

From the LoFi correlation criterion to the information exponent Specializing the first-layer ($\ell = 1$) Neural LoFi criterion of equation (12) to this setting, the input representation is $\boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{x}$, and the population version of the second-order correlation maximized by Neural LoFi at layer 1 becomes

$$\rho(\varphi) \;=\; \Big|\, \mathbb{E}\Big[y\, \varphi(\boldsymbol{x})^2\Big] \Big|, \qquad \varphi \in \mathcal{H}_0,\; \|\varphi\|_{\mathcal{H}_0} = 1, \tag{40}$$

where $\mathcal{H}_0 = L^2(\gamma_d)$ is the Gaussian RKHS associated with the identity representation reducing to the constraint $\|\boldsymbol{w}\| = 1$. . Expanding $f^\star$ in the Hermite basis, $f^\star(\boldsymbol{x}) = \sum_{k\ge 0} \langle f^\star, H_k\rangle\, H_k(U^{\star\top}\boldsymbol{x})$, and any test feature φ in the same basis, the criterion (40) retains only the components of φ^2 that overlap with the lowest non-zero Hermite component of $f^\star$. Following [72], define the *information exponent* of $f^\star$ as

$$\mathrm{IE}(f^\star) \;\coloneqq\; \min\big\{\, k \ge 1 \,:\, \mathbb{E}[f^\star(\boldsymbol{x})\, H_k(\langle \boldsymbol{u}, \boldsymbol{x}\rangle)] \not\equiv 0 \text{ for some } \boldsymbol{u} \in V^\star \,\big\}. \tag{41}$$

Our second-order correlation $\rho(\varphi)$ probes φ^2, so it is sensitive to the first two Hermite components. In particular,

- if $\mathrm{IE}(f^\star) = 1$ (linear teacher direction), the maximizer of (40) corresponds to φ linear in $\boldsymbol{x}$ and aligned with the leading direction of $\mathbb{E}[y\boldsymbol{x}]$;
- if $\mathrm{IE}(f^\star) = 2$, the maximizer is a quadratic form in $\boldsymbol{x}$ and the criterion reduces to the top-eigenvector problem for the population correlation matrix $C^\star \coloneqq \mathbb{E}[y\, \boldsymbol{x}\boldsymbol{x}^\top] - \mathbb{E}[y]\, I_d$;
- if $\mathrm{IE}(f^\star) \ge 3$, the second-order correlation is degenerate at the population level on the hidden subspace, and a single LoFi step recovers no signal.

This is precisely the well-known threshold separating "easy" from "hard" multi-index problems for online SGD [18, 19, 71, 72].

Neural LoFi suffices for weak recovery when $\mathrm{IE} \le 2$. Let $\hat{C}^{(1)} = \frac{1}{n}\sum_{\mu=1}^{n} y_\mu\, \boldsymbol{x}_\mu \boldsymbol{x}_\mu^\top - \bar{y}\, I_d$ be the empirical correlation used by Neural LoFi at layer 1, and let $\widehat{V}_1$ be the top-k_1 eigenspace of $\hat{C}^{(1)}$. The following statement, which is a direct consequence of matrix concentration results applied to $\hat{C}^{(1)}$, makes the link with weak recovery quantitative.

Proposition 2 (Neural LoFi recovers $V^\star$ when $\mathrm{IE} \le 2$)**.** *Assume the multi-index model* (38) *with* $\mathrm{IE}(f^\star) \le 2$ *and bounded link function. There exist constants* $c, C, \delta > 0$ *depending only on* $g^\star$ *such that, with sample size* $n \ge C\, d$*, the top-*r *eigenspace* $\widehat{V}_1$ *of* $\hat{C}^{(1)}$ *satisfies*

$$\left\| P_{\widehat{V}_1} - P_{V^\star} \right\|_{\mathrm{op}} \;\le\; 1 - \delta, \qquad \textit{i.e. weak recovery of } V^\star, \tag{42}$$

with probability $1 - d^{-c}$.

In other words, when $\mathrm{IE} \le 2$, a single Neural LoFi layer is statistically sufficient: the same low-degree filtering step that, in the main text, defines the Neural LoFi feature already captures the hidden subspace at the optimal $n = \Theta(d)$ sample complexity, in agreement with the upper bounds of [64, 72]. When $\mathrm{IE} \ge 3$, Proposition 2 fails by construction, and recovery of $V^\star$ requires either non-Gaussian preprocessing of y (e.g. polynomial reweightings, as in [19, 65]) or genuinely deeper compositionality, which is the regime studied in Appendix B and the main text.

Saddle-to-saddle dynamics in the Neural LoFi regime with $\mathrm{IE} = 2$**.** We now describe how, under the same multi-index model with $\mathrm{IE}(f^\star) = 2$, the actual gradient-descent dynamics of the two-layer network (39) traverse the top directions of $C^\star$ in a saddle-to-saddle cascade, recovering the same ordered features as Neural LoFi.

Order the eigenvalues of $C^\star$ as $\lambda_1 > \lambda_2 > \cdots > \lambda_r > 0$ on the hidden subspace and $\lambda_{r+1} = \cdots = \lambda_d = 0$ off it, with associated eigenvectors $\boldsymbol{v}_1^\star, \ldots, \boldsymbol{v}_r^\star$. With small initialization $\|\boldsymbol{w}_i(0)\| \asymp d^{-1/2}$ and step-size η, the leading-order continuous-time dynamics of the (rescaled) neuron weights $\widetilde{\boldsymbol{w}}_i(t)$ satisfy, after averaging over the second-layer signs, a power-iteration–type ODE driven by $C^\star$:

$$\dot{\widetilde{\boldsymbol{w}}}_i(t) \;=\; \big(C^\star - \widetilde{\boldsymbol{w}}_i^\top C^\star \widetilde{\boldsymbol{w}}_i\, I_d\big)\, \widetilde{\boldsymbol{w}}_i(t) \;+\; o(1), \tag{43}$$

Equation (43) is the gradient flow of $-\widetilde{\boldsymbol{w}}^\top C^\star \widetilde{\boldsymbol{w}}$ on the sphere, whose only attractors are the top eigenvectors $\pm \boldsymbol{v}_1^\star$ and whose other critical points are strict saddles indexed by the remaining $\boldsymbol{v}_j^\star$.

When several eigenvalues λ_j are well separated, this gives rise to the *saddle-to-saddle* picture: starting from a small isotropic initialization, the trajectory spends a time $T_j \asymp \lambda_j^{-1} \log(d)$ in a neighborhood of the saddle associated with $\boldsymbol{v}_j^\star$ before escaping along the next leading direction [115]. The neurons therefore learn the directions $\boldsymbol{v}_1^\star, \boldsymbol{v}_2^\star, \ldots, \boldsymbol{v}_r^\star$ *sequentially*, in order of decreasing population correlation λ_j, with sharp transitions between successive plateaus.

The eigenbasis traversed by GD in such saddle-to-saddle dynamics is exactly the basis selected by the Neural LoFi criterion (40). In both cases, the relevant operator is $C^\star = \mathbb{E}[y\, \boldsymbol{x}\boldsymbol{x}^\top]$ (up to centering), and the ordering by λ_j is the ordering of low-degree correlations with the label. Neural LoFi can thus be viewed as the *static abstraction* of the saddle-to-saddle GD dynamics in the $\mathrm{IE} = 2$ regime, replacing the dynamics through saddles by a single eigendecomposition of $\hat{C}^{(1)}$.

Finally, we note that with the use of a different loss other than the squared loss or with data reuse or label transformations [19, 65], the criterion in Equation 40 is modified to:

$$\rho_g(\varphi) \;=\; \Big|\, \mathbb{E}\Big[g(y)\, \varphi(\boldsymbol{x})^2\Big] \Big|, \qquad \varphi \in \mathcal{H}_0,\ \|\varphi\|_{\mathcal{H}_0} = 1, \tag{44}$$

for a transformation $g : \mathbb{R} \to \mathbb{R}$. The condition $\rho_g(\varphi) \neq 0$ then corresponds to generative exponent [63, 65] ≥ 2 instead of the information exponent [72].

D Vector labels

In the main text we wrote Neural LoFi for scalar labels. For vector labels $\boldsymbol{y} \in \mathbb{R}^c$, the natural object is not a signed scalar correlation but the label-space vector of correlations. Given the layer representation $\boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \in \mathbb{R}^{p_{\ell-1}}$ and a scalar feature $\psi : \mathcal{X} \to \mathbb{R}$, define

$$\mathbf{c}_{\ell,n}[\psi] := \widehat{\mathbb{E}}_n[\boldsymbol{y}\,\psi(\boldsymbol{x})] = \big(\widehat{\mathbb{E}}_n[y_a\psi(\boldsymbol{x})]\big)_{a=1}^{c} \in \mathbb{R}^c. \tag{45}$$

For a fixed unit label direction $\boldsymbol{a}$, the corresponding scalar LoFi correlation is

$$\widehat{\mathbb{E}}_n[\langle \boldsymbol{a}, \boldsymbol{y}\rangle\,\psi(\boldsymbol{x})] = \langle \boldsymbol{a}, \mathbf{c}_{\ell,n}[\psi]\rangle.$$

Optimizing this projected scalar correlation over $\boldsymbol{a}$ gives

$$\big\|\mathbf{c}_{\ell,n}[\psi]\big\|_2^2 = \max_{\|\boldsymbol{a}\|_2=1} \Big|\widehat{\mathbb{E}}_n[\langle \boldsymbol{a}, \boldsymbol{y}\rangle\,\psi(\boldsymbol{x})]\Big|^2.$$

Thus the vector-label first-order LoFi criterion is

$$\psi_{\ell,j}^{(1)} \in \arg \max_{\substack{\|\psi\|_{\mathcal{H}_{\ell-1}}=1 \\ \psi\perp\psi_{\ell,1}^{(1)},\ldots,\psi_{\ell,j-1}^{(1)}}} \Big\|\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\psi(\boldsymbol{x})]\Big\|_2^2. \tag{46}$$

The vector-label second-order LoFi criterion is

$$\varphi_{\ell,j}^{(2)} \in \arg \max_{\substack{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1 \\ \varphi\perp\varphi_{\ell,1}^{(2)},\ldots,\varphi_{\ell,j-1}^{(2)}}} \Big\|\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\varphi(\boldsymbol{x})^2]\Big\|_2^2. \tag{47}$$

D.1 Parametric implementation

We now explain how the feature-selection criteria (46)–(47) translate into concrete spectral computations when the candidate features are linear functions of the current representation. The first-order criterion becomes an SVD of a label-feature matrix, while the second-order criterion becomes an alternating Rayleigh maximization over a label direction and a feature direction.

First-order term: SVD. For a linear feature $\psi_{\boldsymbol{v}}(\boldsymbol{x}) = \langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\rangle$, define

$$\widehat{M}_\ell := \widehat{\mathbb{E}}_n[\boldsymbol{y}\,\boldsymbol{z}_{\ell-1}^\top] \in \mathbb{R}^{c\times p_{\ell-1}}.$$

Then $\mathbf{c}_{\ell,n}[\psi_{\boldsymbol{v}}] = \widehat{M}_\ell \boldsymbol{v}$, so (46) becomes

$$\max_{\|\boldsymbol{v}\|_2=1} \|\widehat{M}_\ell \boldsymbol{v}\|_2^2.$$

The maximizing feature directions are the right singular vectors of $\widehat{M}_\ell$, while the associated label directions are the left singular vectors. In the first layer this is exactly the SVD of $\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\boldsymbol{x}^\top]$.

Second-order term: alternating Rayleigh maximization. For a linear second-order feature $\varphi_{\boldsymbol{v}}(\boldsymbol{x}) = \langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\rangle$, define

$$\widehat{\boldsymbol{r}}_\ell(\boldsymbol{v}) := \widehat{\mathbb{E}}_n[\langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}\rangle^2 \boldsymbol{y}] \in \mathbb{R}^c.$$

Then (47) is

$$\max_{\|\boldsymbol{v}\|_2=1} \|\widehat{\boldsymbol{r}}_\ell(\boldsymbol{v})\|_2^2 = \max_{\|\boldsymbol{v}\|_2=1,\ \|\boldsymbol{a}\|_2=1} \Big|\boldsymbol{v}^\top \widehat{C}_\ell(\boldsymbol{a})\boldsymbol{v}\Big|^2,$$

where

$$\widehat{C}_\ell(\boldsymbol{a}) := \widehat{\mathbb{E}}_n[\langle \boldsymbol{a}, \boldsymbol{y}\rangle\,\boldsymbol{z}_{\ell-1}\boldsymbol{z}_{\ell-1}^\top] \in \mathbb{R}^{p_{\ell-1}\times p_{\ell-1}}.$$

For fixed $\boldsymbol{a}$, the best $\boldsymbol{v}$ is the signed leading eigenvector of $\widehat{C}_\ell(\boldsymbol{a})$, ordered by $|\lambda|$. For fixed $\boldsymbol{v}$, the best label direction is

$$\boldsymbol{a} = \widehat{\boldsymbol{r}}_\ell(\boldsymbol{v})/\|\widehat{\boldsymbol{r}}_\ell(\boldsymbol{v})\|_2.$$

This gives the alternating maximization used in the vector-label algorithm.

Combining the two spectral steps gives the vector-label Neural LoFi procedure: at each layer we keep the leading right singular directions from the first-order matrix, add the directions found by the

second-order alternating Rayleigh step, orthogonalize the resulting feature subspace, and then apply the same random feature lift as in scalar Neural LoFi.

Algorithm 2 Neural LoFi for vector labels

Require: Dataset $\{(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu)\}_{\mu=1}^n$ with $\boldsymbol{y}_\mu \in \mathbb{R}^c$, depth L, first-order ranks $\{q_\ell\}_{\ell=1}^L$, second-order ranks $\{k_\ell\}_{\ell=1}^L$, widths $\{p_\ell\}_{\ell=1}^L$, inner iterations T, tolerance ε.

Initialize $\boldsymbol{z}_0(\boldsymbol{x}) \leftarrow \boldsymbol{x}$ and $p_0 = d$.
for $\ell = 1, \dots, L$ **do**
  Form data matrices $Y = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_n]^\top$ and $Z_{\ell-1} = [\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_1), \dots, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_n)]^\top$.
  Compute the first-order SVD

$$\widehat{M}_\ell = \frac{1}{n} Y^\top Z_{\ell-1} = \widehat{U}_\ell \widehat{\Sigma}_\ell (\widehat{V}_\ell^{(1)})^\top,$$

  and retain the top q_ℓ right singular vectors in $\widehat{V}_{\ell,q}^{(1)}$.
  Initialize $\boldsymbol{a}^{(0)} \in \mathbb{S}^{c-1}$, e.g. the leading left singular vector in $\widehat{U}_\ell$.
  for $t = 0, \dots, T-1$ **do**
    Form the projected-label covariance

$$\widehat{C}_{\boldsymbol{a}^{(t)}}^{(\ell)} \leftarrow \frac{1}{n} \sum_{\mu=1}^n \langle \boldsymbol{a}^{(t)}, \boldsymbol{y}_\mu \rangle \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top.$$

    Compute the signed leading eigenpair

$$(\widehat{\lambda}^{(t)}, \widehat{\boldsymbol{v}}^{(t)}) \in \arg\max_{\lambda, \, \|\boldsymbol{v}\|_2 = 1} |\lambda| \quad \text{s.t.} \quad \widehat{C}_{\boldsymbol{a}^{(t)}}^{(\ell)} \boldsymbol{v} = \lambda \boldsymbol{v}.$$

    Compute the label-correlation vector

$$\widehat{\boldsymbol{r}}^{(t)} \leftarrow \frac{1}{n} \sum_{\mu=1}^n \langle \widehat{\boldsymbol{v}}^{(t)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \rangle^2 \boldsymbol{y}_\mu.$$

    Update $\boldsymbol{a}^{(t+1)} \leftarrow \widehat{\boldsymbol{r}}^{(t)} / \|\widehat{\boldsymbol{r}}^{(t)}\|_2$.
    if $t > 0$ and $|\|\widehat{\boldsymbol{r}}^{(t)}\|_2 - \|\widehat{\boldsymbol{r}}^{(t-1)}\|_2| < \varepsilon$ **then break**.
  Set $\boldsymbol{a}_\ell^\star \leftarrow \boldsymbol{a}^{(t)}$. Form $\widehat{C}_{\boldsymbol{a}_\ell^\star}^{(\ell)}$ and extract k_ℓ eigenvectors by decreasing $|\widehat{\lambda}|$:

$$\widehat{V}_\ell^{(2)} = [\widehat{\boldsymbol{v}}_1^{(\ell)}, \dots, \widehat{\boldsymbol{v}}_{k_\ell}^{(\ell)}].$$

  Combine first- and second-order directions

$$\widehat{V}_\ell \leftarrow \mathrm{orth}\big([\widehat{V}_{\ell,q}^{(1)}, \widehat{V}_\ell^{(2)}]\big).$$

  Project:

$$\boldsymbol{g}_\ell(\boldsymbol{x}) \leftarrow \widehat{V}_\ell^\top \boldsymbol{z}_{\ell-1}(\boldsymbol{x}).$$

  Lift:

$$\boldsymbol{z}_\ell(\boldsymbol{x}) \leftarrow p_\ell^{-1/2} \sigma(R_\ell \boldsymbol{g}_\ell(\boldsymbol{x})), \qquad R_\ell \in \mathbb{R}^{p_\ell \times (q_\ell + k_\ell)}.$$

Fit a final linear or logistic readout $A \in \mathbb{R}^{c \times p_L}$ on $\boldsymbol{z}_L(\boldsymbol{x})$.
return Final predictor $\widehat{\boldsymbol{y}}(\boldsymbol{x}) = A \boldsymbol{z}_L(\boldsymbol{x})$.

D.2 Connection with gradient descent

The preceding subsection gives a static algorithmic rule. We now explain why the same spectral objects arise from the early gradient-descent dynamics of a vector-output network. We start with the two-layer setting, where the updates of the readout and hidden weights form a coupled power iteration on the label-feature matrix, and then indicate why the same mechanism is local in depth.

Consider a two-layer vector-output network

$$\widehat{\boldsymbol{y}}(\boldsymbol{x}) = A\sigma(W\boldsymbol{z}(\boldsymbol{x})) = \sum_{i=1}^p \boldsymbol{a}_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{z}(\boldsymbol{x}) \rangle), \qquad A = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_p] \in \mathbb{R}^{c \times p}, \quad W = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \vdots \\ \boldsymbol{w}_p^\top \end{bmatrix},$$

trained with squared loss and residual $\boldsymbol{e}_\mu = \boldsymbol{y}_\mu - \widehat{\boldsymbol{y}}(\boldsymbol{x}_\mu)$. Let

$$\widehat{M} := \widehat{\mathbb{E}}_n[\boldsymbol{y}\,\boldsymbol{z}^\top] \in \mathbb{R}^{c\times p_{\ell-1}}.$$

At small initialization scale, $\boldsymbol{e} = \boldsymbol{y} + o(1)$, while $\sigma(W\boldsymbol{z}) = c_0 W\boldsymbol{z}$ + higher-order terms and $\sigma'(\langle \boldsymbol{w}_i, \boldsymbol{z}\rangle) = c_0$ + higher-order terms. Hence the readout update is

$$\frac{dA}{dt} = \widehat{\mathbb{E}}_n[\boldsymbol{e}\,\sigma(W\boldsymbol{z})^\top] = c_0\,\widehat{M}W^\top + \text{higher-order terms}.$$

Thus A is driven by the right projection of the label-feature matrix $\widehat{M}$ through the current hidden directions W: column i obeys $\frac{d\boldsymbol{a}_i}{dt} \simeq c_0\widehat{M}\boldsymbol{w}_i$. Conversely, the hidden weights obey

$$\frac{d\boldsymbol{w}_i}{dt} = \widehat{\mathbb{E}}_n[\langle \boldsymbol{e}, \boldsymbol{a}_i\rangle\,\sigma'(\langle \boldsymbol{w}_i, \boldsymbol{z}\rangle)\boldsymbol{z}]. \tag{48}$$

The same expansion gives

$$\frac{d\boldsymbol{w}_i}{dt} = c_0\,\widehat{M}^\top \boldsymbol{a}_i + \text{higher-order terms},$$

$$\text{or equivalently} \qquad \frac{dW}{dt} = c_0\,A^\top \widehat{M} + \text{higher-order terms}.$$

Thus W is driven by the left projection of $\widehat{M}$ through the current readout directions A. Ignoring the higher-order terms, the leading dynamics are the coupled power iteration

$$\frac{dA}{dt} \simeq c_0\widehat{M}W^\top, \qquad \frac{dW^\top}{dt} \simeq c_0\widehat{M}^\top A.$$

After normalization and deflation, the coupled iteration aligns the readout directions with the left singular vectors of $\widehat{M}$ and the hidden directions with the corresponding right singular vectors. This is the gradient-descent mechanism behind the first-order vector LoFi SVD of $\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\boldsymbol{z}^\top]$.

The same argument is local in depth. For a general layer ℓ, view $\boldsymbol{z}_{\ell-1}$ as the input to the block $\boldsymbol{z}_\ell = \sigma(W_\ell \boldsymbol{z}_{\ell-1})$, and view the next-layer weights $W_{\ell+1}$, together with the backpropagated residual at layer $\ell+1$, as the vector readout. Then the update of $W_{\ell+1}$ plays the role of the update of A above, while the update of W_ℓ plays the role of the update of W. Thus the same left/right projection mechanism produces the layerwise SVD picture.

Keeping also the term linear in $\boldsymbol{w}_i$ in the hidden-weight expansion gives

$$\frac{d\boldsymbol{w}_i}{dt} = c_0\,\widehat{\mathbb{E}}_n[\langle \boldsymbol{y}, \boldsymbol{a}_i\rangle\boldsymbol{z}] + c_1\,\widehat{C}(\boldsymbol{a}_i)\boldsymbol{w}_i + \text{higher-order terms}. \tag{49}$$

Thus each readout direction $\boldsymbol{a}_i$ induces the scalar projected label $\langle \boldsymbol{y}, \boldsymbol{a}_i\rangle$, and hence the projected-label covariance $\widehat{C}(\boldsymbol{a}_i)$ appearing in Algorithm 2.

This already marks an important difference from the scalar-label case. When $\boldsymbol{y} \in \mathbb{R}^c$, the linear label-feature matrix $\widehat{M} = \widehat{\mathbb{E}}_n[\boldsymbol{y}\boldsymbol{z}^\top]$ has up to c nonzero left singular directions, and its right singular vectors can therefore expose multiple feature directions from the first-order signal alone. Thus, for vector labels, the SVD step on $\widehat{M}$ may already be sufficient for feature learning before one invokes the projected second-order covariance $\widehat{C}(\boldsymbol{a})$.

E A Solvable Theoretical Setting

The agnostic and recursive nature of Neural LoFi calls for a theoretical setting that contains a compositional structure, while not revealing the relevant intermediate variables to the learner. In this appendix, we follow the hierarchical spectral construction of [40], building on the fundamental earlier works [41–43]. This allows us to define a controlled high-dimensional model and to use it to study how depth turns a globally hard learning problem into a sequence of simpler spectral recoveries.

A natural way to isolate the role of depth is to consider teacher-student models where the target is not merely a low-dimensional function of the input, but is built through a hierarchy of intermediate representations. Such models have appeared in several recent works on compositional learning and the computational advantage of depth, including random hierarchy models, hierarchical Gaussian targets, and polynomial teacher-student constructions (e.g. [21, 22, 25, 26, 40–43, 109, 116]). They share the same guiding principle: a target may look high-dimensional or high-degree as a function of the input, while becoming low-degree after the right intermediate representation has been found. This is precisely the situation in which depth should help, since learning can proceed by a sequence of simpler feature-recovery problems rather than by solving the full high-degree task at once.

E.1 Setting

We focus on the high-dimensional Gaussian teacher-student model of [40], which provides a particularly tractable instance of this principle. The input is Gaussian, $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$, and the target is generated by a two-level compositional hierarchy

$$\boldsymbol{x} \in \mathbb{R}^d \quad \longrightarrow \quad h^{(1)}(\boldsymbol{x}) \in \mathbb{R}^{d_1} \quad \longrightarrow \quad h^{(2)}(\boldsymbol{x}) \in \mathbb{R} \quad \longrightarrow \quad y. \tag{50}$$

For $q \geq 1$, we denote by $H_q(\cdot)$ the normalized Hermite polynomial of order q, either in its scalar or tensor-valued form.[2] We write $\langle \cdot, \cdot \rangle$ for the Frobenius product between tensors of the same order.[3] We write F_q for the symmetric flattening map from order-q symmetric tensors to $\mathbb{R}^{D_q}$, with $D_q = \binom{d+q-1}{q}$, chosen so that the Frobenius product is preserved. Below, we freely identify a symmetric tensor with its flattened representation whenever no ambiguity arises. The teacher parameters are given by symmetric tensors

$$A_i^{(1)} \in (\mathbb{R}^d)^{\otimes q}_{\mathrm{sym}}, \qquad i = 1, \dots, d_1, \qquad d_1 = d^\epsilon, \tag{51}$$

normalized so that the first-layer features have order-one variance, and by a symmetric matrix

$$A^{(2)} \in \mathbb{R}^{d_1 \times d_1}. \tag{52}$$

The latent variables and the label are then defined as

$$h_i^{(1)}(\boldsymbol{x}) = \left\langle A_i^{(1)}, H_q(\boldsymbol{x}) \right\rangle, \qquad i = 1, \dots, d_1, \tag{53}$$

$$h^{(2)}(\boldsymbol{x}) = \left\langle A^{(2)}, H_2(h^{(1)}(\boldsymbol{x})) \right\rangle, \tag{54}$$

$$y = g^\star(h^{(2)}(\boldsymbol{x})). \tag{55}$$

Thus the first layer selects only $d_1 = d^\epsilon$ directions inside the ambient degree-q Hermite space of dimension D_q. When $d_1 \ll D_q$, the relevant information is sparse in this low-degree feature space: the target depends on $\boldsymbol{x}$ only through a small hidden subspace of Hermite features. Learning the first layer therefore amounts to recovering this subspace, so that the rest of the hierarchy can be expressed as a low-degree problem in the variables $h^{(1)}$.

The analysis of [40] shows that this sparse compositional structure can be exploited by a hierarchical spectral estimator. Rather than learning the full composed function in one step, the procedure first recovers the hidden degree-q subspace defining $h^{(1)}$, and then uses this recovered representation to make the next component of the hierarchy accessible. In this sense, depth turns the learning problem into a sequence of spectral recovery tasks, each one exposing the variables needed by the next layer.

[2] For a vector $\boldsymbol{x} \in \mathbb{R}^m$, the tensor Hermite polynomial $H_q(\boldsymbol{x}) \in (\mathbb{R}^m)^{\otimes q}_{\mathrm{sym}}$ is defined through the tensorial Rodrigues formula $\sqrt{q!}\, H_q(\boldsymbol{x}) = (-1)^q e^{\|\boldsymbol{x}\|^2/2} \nabla_{\boldsymbol{x}}^{\otimes q} \left(e^{-\|\boldsymbol{x}\|^2/2} \right)$, where $\nabla_{\boldsymbol{x}}^{\otimes q}$ denotes the q-fold symmetric tensor of derivatives. For $m = 1$, this reduces to the normalized scalar Hermite polynomial $\sqrt{q!}\, H_q(z) = (-1)^q e^{z^2/2} \frac{d^q}{dz^q} \left(e^{-z^2/2} \right)$.

[3] For example, if $\boldsymbol{x} \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times m}$ is symmetric, then $\langle A, H_2(\boldsymbol{x}) \rangle = \frac{1}{\sqrt{2}} \left(\boldsymbol{x}^\top A \boldsymbol{x} - \mathrm{Tr}(A) \right)$.

This gives a clean explanation for the advantage of depth in this model. A one-shot method that works directly on the input must resolve the full high-degree dependence of y on x. By contrast, the hierarchical spectral procedure only needs to reveal the first representation and then reuse it to make the next layer visible. In the regime $d_1 = d^\epsilon$, the first stage requires on the order of $d^{q+\epsilon}$ samples, while the second stage requires only the sample complexity of a quadratic problem in dimension d_1. The dominant cost is therefore the recovery of the first hidden representation, rather than the degree of the full composed polynomial.

The price to pay is that the corresponding spectral estimators are still partially co-designed with the Gaussian-Hermite structure of the teacher. Indeed, the first stage is built in an explicit degree-k Hermite feature space, while the second stage uses a prescribed second-order Hermite structure in the recovered variables. A natural way to relax this feature-design aspect is to replace the explicit Hermite construction by nonlinear random features. Rather than specifying the polynomial coordinates in advance, one lets a random feature map generate a generic nonlinear representation and applies the same layer-wise spectral selection in that space. In the hierarchical teacher-student setting above, this random-feature extension of the hierarchical spectral estimator matches precisely the Neural LoFi algorithm.

E.2 Random-feature hierarchical estimator

In this setting, Neural LoFi takes the following concrete form. Let $W_1 = (w_{1,a})_{a=1}^{p_1}$ with $w_{1,a} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, and define the first random-feature representation

$$\phi_\mu^{(1)} = \frac{1}{\sqrt{p_1}} \sigma_1(W_1 x_\mu) \in \mathbb{R}^{p_1}. \tag{56}$$

The first spectral operator is

$$\widehat{C}_1 = \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \phi_\mu^{(1)} \phi_\mu^{(1)\top}. \tag{57}$$

Let $\widehat{V}_1 \in \mathbb{R}^{p_1 \times d_1}$ contain the eigenvectors of $\widehat{C}_1$ associated with the largest eigenvalues in absolute value. Here we keep d_1 directions for simplicity; this can be replaced by a standard rank-selection step. The recovered first-layer coordinates are

$$\widehat{h}_\mu^{(1)} = \widehat{V}_1^\top \phi_\mu^{(1)} \in \mathbb{R}^{d_1}. \tag{58}$$

We then draw $W_2 = (w_{2,a})_{a=1}^{p_2}$ with $w_{2,a} \sim \mathrm{Unif}(\mathbb{S}^{d_1-1})$, and define

$$\phi_\mu^{(2)} = \frac{1}{\sqrt{p_2}} \sigma_2(W_2 \widehat{h}_\mu^{(1)}) \in \mathbb{R}^{p_2}. \tag{59}$$

At the second layer, since the teacher contains only one hidden direction, we do not construct a matrix-valued spectral estimator. Instead, we directly form the first-order moment vector

$$\widehat{v}_2 = \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \phi_\mu^{(2)}. \tag{60}$$

The associated second-layer coordinate is then obtained by projection:

$$\widehat{h}_\mu^{(2)} = \widehat{v}_2^\top \phi_\mu^{(2)}. \tag{61}$$

Finally, the readout is fitted on the one-dimensional representation $\{(\widehat{h}_\mu^{(2)}, y_\mu)\}_{\mu=1}^n$, for instance by ridge regression in a polynomial feature space,

$$\widehat{g} \in \arg\min_{g \in \mathcal{G}} \frac{1}{n} \sum_{\mu=1}^{n} \Big(y_\mu - g(\widehat{h}_\mu^{(2)}) \Big)^2 + \lambda \|g\|_{\mathcal{G}}^2. \tag{62}$$

The choice of keeping exactly d_1 eigenvectors at the first layer is not meant to define a different procedure, but simply to make explicit the outcome of the usual rank-selection step within Neural LoFi in the present setting. More generally, one could keep an arbitrary number k of directions and optimize over k, exactly as in the rest of the pipeline. In the model considered here, this optimization would select d_1 directions at the first layer and a single direction at the second layer. In that sense, fixing d_1 in the first layer and using the linear estimator $\widehat{v}_2$ in the second layer is fully equivalent to the standard Neural LoFi selection rule.

E.3 Main Results

The main prediction of this tractable model is that replacing the explicit Hermite structure of the estimator by random features preserves the emergence transition. In particular, for a degree-q first layer with $d_1 = d^\epsilon$ hidden variables, we expect the first representation $h^{(1)}$ to become recoverable at the sample scale

$$n \gg d^{q+\epsilon}. \tag{63}$$

In the quadratic setting used in our experiments, $q = 2$, and the predicted first-stage transition is therefore $n \gg d^{2+\epsilon}$.

This prediction is the concrete instance of the emergence criterion in Eq. (18) in the main text. Indeed, Eq. (18) states that the empirical fluctuation level is governed by the effective dimension of the current residual feature class. Section G.2 shows that, before the first hidden variables have been recovered, the relevant effective dimension is the size of the degree-q Hermite block, namely $D_q = O(d^q)$. The additional factor $d_1 = d^\epsilon$ comes from separating and aligning with the d_1 planted directions in this block, giving the scale $D_q d_1 = O(d^{q+\epsilon})$.

The role of the next subsection is to explain why the random-feature estimator contains the same signal-bearing Hermite spectral object as the explicit construction, now embedded in the random-feature representation. Figure 2 illustrates the resulting transition numerically: in the quadratic case, the drop in MSE, the growth of the overlap with $h^{(1)}$, and the separation of the leading eigenvalues all occur at the predicted first-layer emergence scale.

E.4 Mathematical Justification

We now analyze the signal structure of the random-feature estimator introduced above. The key point is that, although the estimator is built from generic nonlinear random features, its population signal behaves as a well-defined Hermite estimator aligned with the teacher hierarchy. This will allow us to connect the first step of the Neural LoFi algorithm to the spectral transition established in the Hermite model of [40].

Let us expand the first-layer activation in Hermite as

$$\sigma_1(z) = \sum_{q \geq 2} c_q H_q(z), \qquad c_q = \mathbb{E}_{z \sim \mathcal{N}(0,1)} \left[\sigma_1(z) H_q(z) \right], \tag{64}$$

where H_q denotes the normalized scalar Hermite polynomial of degree q. For each row $\boldsymbol{w}_{1,a}$ of W_1, we use the standard Hermite tensor identity

$$H_q(\langle \boldsymbol{w}_{1,a}, \boldsymbol{x} \rangle) = \left\langle \boldsymbol{w}_{1,a}^{\otimes q}, H_q(\boldsymbol{x}) \right\rangle. \tag{65}$$

Here and below, $\boldsymbol{w}_{1,a}^{\otimes q}$ denotes the degree-q Hermite coefficient vector, with the multi-index normalization chosen so that the above identity holds. Collecting these coefficient tensors over all rows of W_1, and flattening them with the Frobenius-preserving map $F[\cdot]$, we define:

$$\frac{1}{\sqrt{p_1}} H_q(W_1 \boldsymbol{x}) = P_q H_q(\boldsymbol{x}), \quad \text{with} \quad P_q := \frac{1}{\sqrt{p_1}} \begin{bmatrix} F[\boldsymbol{w}_{1,1}^{\otimes q}]^\top \\ \vdots \\ F[\boldsymbol{w}_{1,p_1}^{\otimes q}]^\top \end{bmatrix} \in \mathbb{R}^{p_1 \times D_q}. \tag{66}$$

where $H_q(W_1 \boldsymbol{x}) \in \mathbb{R}^{p_1}$ is understood entrywise on the left-hand side and $H_q(\boldsymbol{x}) \in \mathbb{R}^{D_q}$ is the flattened Hermite tensor. Hence the first random-feature representation decomposes as

$$\phi_\mu^{(1)} = \frac{1}{\sqrt{p_1}} \sigma_1(W_1 \boldsymbol{x}_\mu) = \sum_{q \geq 2} c_q P_q H_q(\boldsymbol{x}_\mu). \tag{67}$$

We denote by

$$\widehat{C}_H^{(q)} := \frac{1}{n} \sum_{\mu=1}^{n} y_\mu H_q(\boldsymbol{x}_\mu) H_q(\boldsymbol{x}_\mu)^\top \tag{68}$$

the degree-q Hermite moment matrix appearing in the middle. If one uses the centered second-order Hermite convention of [40], this matrix should be replaced by its centered version; the difference is an empirical isotropic term, controlled by the same estimates and absorbed in the rates below. With this notation, the degree-q contribution to the first Neural LoFi operator is

$$\widehat{C}_{\mathrm{RF}}^{(q)} = c_q^2 P_q \widehat{C}_H^{(q)} P_q^\top. \tag{69}$$

Thus the random-feature operator contains the Hermite estimator conjugated by the random Hermite embedding P_q, up to the scalar factor c_q^2.

We now focus on the quadratic case $q = 2$, which is the case controlled explicitly by the quadratic Hermite decomposition of [117]. Define the normalized RF Gram

$$G_2 := D_2 P_2^\top P_2. \tag{70}$$

The important point is that G_2 is not close to the identity on the full degree-2 Hermite space. The trace direction produces a deterministic contraction spike. More precisely, by Lemma F.6 and Corollary F.7 of [117], one has the decomposition

$$G_2 = I_{D_2} + K_2 + R_2, \qquad K_2 = \theta_d e e^\top, \qquad |\theta_d| \le \widetilde{O}(d), \tag{71}$$

where $e \in \mathbb{R}^{D_2}$ is the normalized trace direction in the degree-2 Hermite block. The term K_2 is the explicit non-trivial contraction term coming from the trace component of the quadratic Hermite features, while R_2 is the remaining centered fluctuation of the random-feature Gram, with

$$\|R_2\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right), \tag{72}$$

where the $\widetilde{O}(\cdot)$ hides logarithmic factors in d. This conservative full-column concentration bound is not expected to be optimal, but it is sufficient for the projected comparison below.

Finally, let $A^{(1)} \in \mathbb{R}^{d_1 \times D_2}$ be the matrix of planted first-layer Hermite directions, and define its random-feature image by

$$A_{\mathrm{RF}}^{(1)} := \sqrt{D_2}\, A^{(1)} P_2^\top \in \mathbb{R}^{d_1 \times p_1}. \tag{73}$$

We can now state the projected RF analogue of Theorem 3.1 in [40].

Theorem 3 (Projected RF Hermite recovery, quadratic case)**.** *Assume the setting and normalization of Theorem 3.1 in [40], with $d_1 = d^\varepsilon$ and $\varepsilon < 1/2$. With G_2, K_2, R_2 as defined above, the following holds with high probability:*

$$\begin{aligned}
&\sqrt{d_1}\left\| \frac{D_2}{c_2^2} A_{\mathrm{RF}}^{(1)} \widehat{C}_{\mathrm{RF}}^{(2)} A_{\mathrm{RF}}^{(1)\top} - \nu_1 A^{(2)} \right\|_{\mathrm{op}} \\
&\quad \le \widetilde{O}\left(\sqrt{\frac{d^2 d_1}{n}}\right) + \widetilde{O}\left(\frac{d_1}{\sqrt{d}}\right) + \widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right) \\
&\qquad + \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1^2}{d^2} + \sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\left(1 + \sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\right).
\end{aligned} \tag{74}$$

In particular, if

$$n \gg d^2 d_1, \qquad p_1 \gg \sqrt{d_1} d^2,$$

then the additional RF bridge error is $o(1)$. Therefore the projected RF estimator has the same limiting signal as the Hermite estimator, up to the deterministic scalar factor c_2^2/D_2.

Proof. The proof is a direct comparison with the Hermite estimator. By the definitions of $A_{\mathrm{RF}}^{(1)}$, $\widehat{C}_{\mathrm{RF}}^{(2)}$, and G_2, we have the exact identity

$$\frac{D_2}{c_2^2} A_{\mathrm{RF}}^{(1)} \widehat{C}_{\mathrm{RF}}^{(2)} A_{\mathrm{RF}}^{(1)\top} = A^{(1)} G_2 \widehat{C}_H^{(2)} G_2 A^{(1)\top}. \tag{75}$$

Thus it is enough to compare $A^{(1)}G_2\widehat{C}_H^{(2)}G_2A^{(1)\top}$ with $A^{(1)}\widehat{C}_H^{(2)}A^{(1)\top}$.

We use the RF Gram decomposition stated above,

$$G_2 = I_{D_2} + K_2 + R_2, \quad K_2 = \theta_d e e^\top, \quad |\theta_d| \le \widetilde{O}(d), \quad \|R_2\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right).$$

Here K_2 is the trace-contraction spike and R_2 is the centered RF Gram remainder.

Let $C := \widehat{C}_H^{(2)}$. We use the following standard high-probability bounds:

$$\|A^{(1)}e\|_2 \le \widetilde{O}\left(\frac{\sqrt{d_1}}{d}\right), \tag{76}$$

$$\|C\|_{\mathrm{op}} \le \widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right), \tag{77}$$

$$\|e^\top C A^{(1)\top}\|_2 \le \widetilde{O}\left(\sqrt{\frac{d_1}{n}} + \frac{1}{d}\right), \tag{78}$$

$$|e^\top C e| \le \widetilde{O}\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d_1}}{d^2}\right). \tag{79}$$

The first estimate is the overlap of a random d_1-dimensional Gaussian subspace with the fixed trace direction. The remaining estimates are obtained by the same truncation, matrix Bernstein, and Gaussian-equivalence arguments used in Appendix A.4 of [40], applied either with one test tensor equal to the trace direction e, or with both test tensors equal to e. The population contributions are smaller because the trace overlap satisfies $\|A^{(1)}e\|_2 = \widetilde{O}(\sqrt{d_1}/d)$.

Expanding $G_2 = I + K_2 + R_2$ gives

$$G_2 C G_2 - C = K_2C + CK_2 + K_2CK_2 + \text{terms containing } R_2.$$

For the first trace term,

$$A^{(1)}K_2CA^{(1)\top} = \theta_d(A^{(1)}e)(e^\top CA^{(1)\top}),$$

and therefore, using (76)–(78),

$$\sqrt{d_1}\|A^{(1)}K_2CA^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1}{d}\right).$$

The transpose term $A^{(1)}CK_2A^{(1)\top}$ is bounded identically. For the quadratic trace term,

$$A^{(1)}K_2CK_2A^{(1)\top} = \theta_d^2(A^{(1)}e)(e^\top Ce)(A^{(1)}e)^\top,$$

so by (76) and (79),

$$\sqrt{d_1}\|A^{(1)}K_2CK_2A^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1^2}{d^2}\right).$$

Finally, all terms containing R_2 are controlled by $\|R_2\|_{\mathrm{op}}$ and (77). The worst mixed terms are K_2CR_2 and R_2CK_2, and they satisfy

$$\sqrt{d_1}\|A^{(1)}K_2CR_2A^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\right),$$

with the same bound for the transpose. The R_2CR_2 term contributes at most

$$\widetilde{O}\left(\sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)^2\right).$$

Combining these estimates gives the bridge bound

$$\sqrt{d_1}\left\|A^{(1)}\left(G_2CG_2-C\right)A^{(1)\top}\right\|_{\mathrm{op}} \le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}}+\frac{d_1}{d}+\frac{d_1^2}{d^2}\right) + \widetilde{O}\left(\sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}}+\frac{D_2}{p_1}\right)\left(1+\sqrt{\frac{D_2}{p_1}}+\frac{D_2}{p_1}\right)\right). \quad (80)$$

Theorem 3.1 of [40] gives

$$\sqrt{d_1}\left\|A^{(1)}\widehat{C}_H^{(2)}A^{(1)\top}-\nu_1 A^{(2)}\right\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{\frac{d^2d_1}{n}}\right)+\widetilde{O}\left(\frac{d_1}{\sqrt{d}}\right)+\widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right).$$

Combining this with (80) proves (74). □

We stated the theorem for the quadratic case $q = 2$ because the contraction correction is completely explicit. For any fixed degree q, the same argument applies by replacing the trace direction by the finite list of lower contraction sectors given by the Gegenbauer decomposition; see Lemma F.8 of [117]. These contraction terms are subleading on the random planted subspace by the same vanishing-contraction estimates used in Appendix A.1–A.4 of [40]. Consequently, the Hermite recovery theorem transfers to the degree-q RF estimator with the same sample scaling. Together with the RF Gram concentration requirement, the sufficient scaling for recovery is

$$n \gg D_q d_1, \qquad p_1 \gg D_q, \quad (81)$$

up to logarithmic factors. Under these conditions, the degree-q RF estimator has the same projected signal limit as the Hermite estimator, up to the scalar factor c_q^2/D_q.

The projected comparison above shows that the first LoFi step has the same signal scaling as the explicit Hermite estimator. Therefore, the bottleneck remains the Hermite recovery scale of the first layer, together with the RF width needed to realize the corresponding degree-q block. The same reasoning applies recursively to later LoFi steps, with the ambient dimension replaced by the dimension of the representation entering that layer.

E.5 Numerical experiments

We begin by defining the two observables reported in the random-feature experiments. Given a fresh test set $\{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^{n_{\mathrm{test}}}$, we measure the prediction error through the test mean-squared error

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{n_{\mathrm{test}}}\sum_{\mu=1}^{n_{\mathrm{test}}}\left(\hat{y}_\mu - y_\mu\right)^2, \quad (82)$$

and we quantify the recovery of the first hidden layer through the feature overlap

$$\mathrm{Overlap}(h^{(1)}, \widehat{h}^{(1)}) = \left\|H^{(1)}\hat{H}^{(1)\top}\right\|_F^2, \quad H^{(1)} = \left(h_\mu^{(1)}\right)_{\mu\le n_{\mathrm{test}}}, \quad \hat{H}^{(1)} = \left(\hat{h}_\mu^{(1)}\right)_{\mu\le n_{\mathrm{test}}}. \quad (83)$$

All curves are averaged over 10 seeds, and error bars indicate one standard deviation. Concerning the precise setting of the experiments, we specialize to the quadratic first-layer setting $q = 2$. Unless otherwise stated, we fix

$$d_1 = \lfloor d^\epsilon \rfloor, \qquad \epsilon = \frac{1}{2}, \qquad n = \lfloor d^\alpha \rfloor, \quad (84)$$

and generate labels according to

$$y = g^\star(h^{(2)}(\boldsymbol{x})), \qquad g^\star(t) = \tanh(t). \quad (85)$$

The purpose of these experiments is to test whether the random-feature pipeline exhibits the same first-layer emergence transition as the explicit Hermite spectral estimator, while replacing the structured Hermite features by generic nonlinear random features.

The random-feature maps used by the learner are chosen independently of the teacher. The rows of the random matrices are sampled uniformly on the sphere,

$$\boldsymbol{w}_{1,a} \sim \mathrm{Unif}(\mathbb{S}^{d-1}), \qquad \boldsymbol{w}_{2,a} \sim \mathrm{Unif}(\mathbb{S}^{d_1-1}), \quad (86)$$

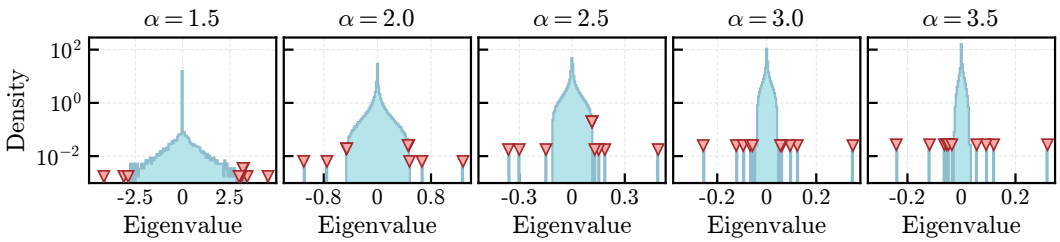


Figure 5: **Spectral emergence in the Neural LoFi estimator.** Spectrum of the first random-feature spectral operator $\widehat{C}_1$ for the hierarchical solvable model of section 4.1, shown at increasing sample exponents $\alpha = \log(n)/\log(d)$. Blue histograms display the bulk eigenvalue density, while red triangles indicate the leading d_1 eigenvalues in absolute value. As α increases, the leading eigenvalues progressively separate from the bulk, marking the *emergence* of concepts in the first layer, as predicted by the sample-complexity scale $n \gg d^{2+\epsilon}$ (from Eq. (17)) in the quadratic setting $k = 2$.

as in the estimator of Section E.2. For both layers, we use a ReLU activation with its degree-zero and degree-one Hermite components removed:

$$\sigma_{\perp\{0,1\}}(z) = \mathrm{ReLU}(z) - c_0 H_0(z) - c_1 H_1(z), \qquad c_r = \mathbb{E}_{G\sim\mathcal{N}(0,1)}\left[\mathrm{ReLU}(G)H_r(G)\right]. \quad (87)$$

The random-feature widths are chosen in the overcomplete regime relative to the quadratic Hermite block,

$$p_1 \gtrsim D_2, \qquad D_2 = \binom{d+1}{2}, \quad (88)$$

in agreement with the finite-width requirement appearing in Theorem 3. In the second layer, the width p_2 is chosen large enough with an equivalent regime as for the first layer, $p_2 \geq d_1^2$, with the same activation $\sigma_{\perp\{0,1\}}$ at both layers.

For each value of d and α, we generate a training set of size $n = \lfloor d^\alpha \rfloor$ and an independent test set. We follow the Neural LoFi learning strategy detailed in Sec. E.2, adding only batch normalization to smooth the anisotropy. The readout $\widehat{g}$ is fitted by ridge regression on degree-5 polynomial features of $\widehat{h}^{(2)}$, with regularization parameter obtained by cross-validation. We additionally apply the same output calibration procedure throughout all experiments.

Figure 2 shows that the random-feature estimator undergoes a clear transition near the predicted first-layer scale. For small α, the test MSE remains close to its baseline value and the overlap with $h^{(1)}$ is essentially zero, indicating that the first hidden representation is not yet spectrally accessible. Around the predicted scale $n \gg d^{2+\epsilon}$, the MSE drops and the representation overlap grows sharply. This confirms that the improvement in prediction is tied to the recovery of the first hidden representation, rather than merely to the final one-dimensional regression step.

Figure 5 gives a more direct spectral view of the same phenomenon. At small sample sizes, the leading eigenvalues of $\widehat{C}_1$ remain buried in the random-feature bulk. As α increases, the leading d_1 eigenvalues separate from the bulk and become stable outliers. This outlier formation is the spectral signature of the first-layer signal emerging in the random-feature representation. Together with the overlap curve, it shows that Neural LoFi recovers the hidden degree-2 representation at the same sample scale predicted by the Hermite theory, while avoiding the explicit construction of the Hermite feature map.

F Neural LoFi Kernel

F.1 Preliminaries: RKHS, spectral theorem and kernel integral operators

This subsection collects the standard background on Reproducing Kernel Hilbert Spaces (RKHS) and the spectral decomposition of kernel integral operators that we use in the rest of the appendix. We refer to [27, 28] for a more detailed exposition. Throughout, $(\mathcal{X}, \mu)$ denotes a measurable input space equipped with a finite Borel measure μ (e.g. the data distribution), and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a symmetric kernel.

Reproducing Kernel Hilbert Space A Hilbert space $\mathcal{H}$ of real-valued functions on $\mathcal{X}$ is a *Reproducing Kernel Hilbert Space* (RKHS) if, for every $\boldsymbol{x} \in \mathcal{X}$, the evaluation functional $\mathrm{ev}_{\boldsymbol{x}} : f \mapsto f(\boldsymbol{x})$ is bounded (continuous) on $\mathcal{H}$. By the Riesz representation theorem, there exists a unique element $K_{\boldsymbol{x}} \in \mathcal{H}$ such that

$$f(\boldsymbol{x}) = \langle f, K_{\boldsymbol{x}} \rangle_{\mathcal{H}} \qquad \text{for all } f \in \mathcal{H}. \tag{89}$$

Defining $K(\boldsymbol{x}, \boldsymbol{x}') := \langle K_{\boldsymbol{x}}, K_{\boldsymbol{x}'} \rangle_{\mathcal{H}} = K_{\boldsymbol{x}'}(\boldsymbol{x})$ yields a symmetric positive semi-definite kernel, and (89) is called the *reproducing property* because the kernel section $K_{\boldsymbol{x}} = K(\boldsymbol{x}, \cdot)$ literally reproduces the value of f at $\boldsymbol{x}$ via an inner product. Conversely, by the Moore–Aronszajn theorem [118], every symmetric positive semi-definite kernel K uniquely determines an RKHS $\mathcal{H}_K$ in which (89) holds; concretely, $\mathcal{H}_K$ is obtained by completing $\mathrm{span}\{K(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathcal{X}\}$ under the inner product $\langle K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot) \rangle_{\mathcal{H}_K} = K(\boldsymbol{x}, \boldsymbol{x}')$.

Spectral theorem for compact self-adjoint operators Let $\mathcal{H}$ be a separable Hilbert space and $T : \mathcal{H} \to \mathcal{H}$ a bounded linear operator. Recall that T is *self-adjoint* if $\langle Tf, g \rangle = \langle f, Tg \rangle$, and *compact* if it maps bounded sets to relatively compact sets (equivalently, it is a norm-limit of finite-rank operators). The spectral theorem states:

Theorem 4 (Hilbert–Schmidt spectral theorem)**.** *If T is compact and self-adjoint on a separable Hilbert space $\mathcal{H}$, then there exist a (finite or countable) sequence of real eigenvalues $\{\lambda_i\}_{i \geq 1}$ with $|\lambda_1| \geq |\lambda_2| \geq \cdots \to 0$ and an orthonormal system $\{e_i\}_{i \geq 1} \subset \mathcal{H}$ of eigenvectors $Te_i = \lambda_i e_i$ such that*

$$Tf \;=\; \sum_{i \geq 1} \lambda_i \, \langle f, e_i \rangle \, e_i \qquad \textit{for all } f \in \mathcal{H}. \tag{90}$$

Trace-class and Hilbert–Schmidt operators are special cases of compact operators on which the spectrum is, respectively, absolutely summable ($\sum_i |\lambda_i| < \infty$) and square-summable ($\sum_i \lambda_i^2 < \infty$).

Kernel integral operator Given a kernel K and the measure μ, the associated *integral operator* $T_K : L^2(\mathcal{X}, \mu) \to L^2(\mathcal{X}, \mu)$ is defined by

$$(T_K f)(\boldsymbol{x}) \;:=\; \int_{\mathcal{X}} K(\boldsymbol{x}, \boldsymbol{x}') \, f(\boldsymbol{x}') \, d\mu(\boldsymbol{x}'). \tag{91}$$

Under the standard assumptions that K is symmetric, continuous (or merely measurable) and square-integrable in the sense that $\int_{\mathcal{X} \times \mathcal{X}} K(\boldsymbol{x}, \boldsymbol{x}')^2 \, d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}') < \infty$, the operator T_K is self-adjoint and Hilbert–Schmidt (in particular compact) on $L^2(\mathcal{X}, \mu)$.

Applying Theorem 4 to T_K yields eigenpairs $\{(\lambda_i, e_i)\}_{i \geq 1}$ with $\lambda_i \geq 0$, $\lambda_i \downarrow 0$, and $\{e_i\}$ orthonormal in $L^2(\mathcal{X}, \mu)$. Mercer's theorem [119] then states that, under mild continuity assumptions on $\mathcal{X}$ and K (e.g. $\mathcal{X}$ compact and K continuous), the kernel itself admits the absolutely and uniformly convergent expansion

$$K(\boldsymbol{x}, \boldsymbol{x}') \;=\; \sum_{i \geq 1} \lambda_i \, e_i(\boldsymbol{x}) \, e_i(\boldsymbol{x}'). \tag{92}$$

Relation between the integral operator and the RKHS The Mercer decomposition (92) provides an explicit isometric description of the RKHS $\mathcal{H}_K$ in terms of the spectral data of T_K. Restricting to indices with $\lambda_i > 0$,

$$\mathcal{H}_K \;=\; \Big\{ f = \sum_{i : \lambda_i > 0} a_i \, e_i \;:\; \|f\|_{\mathcal{H}_K}^2 \;:=\; \sum_{i : \lambda_i > 0} \frac{a_i^2}{\lambda_i} < \infty \Big\}, \tag{93}$$

with inner product $\langle f, g \rangle_{\mathcal{H}_K} = \sum_i a_i b_i / \lambda_i$. Equivalently, $\mathcal{H}_K$ is the image of $L^2(\mathcal{X}, \mu)$ under the square root $T_K^{1/2}$, with norm $\|T_K^{1/2} g\|_{\mathcal{H}_K} = \|g\|_{L^2(\mathcal{X}, \mu)}$ (modulo $\ker T_K$); equivalently, $T_K^{1/2} : L^2(\mathcal{X}, \mu) \to \mathcal{H}_K$ is a partial isometry. Two consequences will be used repeatedly below:

- The eigenfunctions $\{\sqrt{\lambda_i}\, e_i\}_{i:\lambda_i>0}$ form an orthonormal basis of $\mathcal{H}_K$, while $\{e_i\}$ form an orthonormal basis of the closure of $\mathrm{range}(T_K)$ in $L^2(\mathcal{X}, \mu)$.
- Smoothness in $\mathcal{H}_K$ corresponds to spectral concentration: $f \in \mathcal{H}_K$ iff its L^2 coefficients $a_i = \langle f, e_i \rangle_{L^2}$ decay fast enough that $\sum_i a_i^2/\lambda_i < \infty$. In particular, functions in the top-k eigenspace of T_K are exactly the "low-degree" or "low-frequency" functions used throughout the main text.

This dictionary between the kernel K, the integral operator T_K and the RKHS $\mathcal{H}_K$ is what allows us, in the rest of this appendix, to translate between function-space statements (norms, projections, low-degree truncation in $\mathcal{H}_K$) and operator-spectral statements (eigenvalues and eigenfunctions of T_K).

F.2 Representer Property

We begin by establishing a fundamental property: the optimal features lie in the finite-dimensional span of training features, which enables efficient computation via the kernel trick.

Lemma 1 (Representer property of LoFi features)**.** *Let $\{\hat{\boldsymbol{v}}_j^{(\ell)}\}_{j=1}^{k_\ell}$ denote any minimizers of the empirical objective in* (8)*. Then the corresponding weight vectors $\{\hat{\boldsymbol{v}}_j^{(\ell)}\}_{j=1}^{k_\ell}$ lie in the span of the previous-layer features evaluated at the training inputs, i.e.*

$$\hat{\boldsymbol{v}}_j^{(\ell)} \in \mathrm{span}\big\{\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_1), \ldots, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_n)\big\}, \qquad j = 1, \ldots, k_\ell. \tag{94}$$

Proof of Lemma 1. Fix any index $j \in \{1, \ldots, k_\ell\}$. The empirical objective in (8) depends on $\hat{\boldsymbol{v}}_j^{(\ell)}$ only through the inner products $\langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle$, $\mu = 1, \ldots, n$, and is optimized subject to the norm constraint $\|\hat{\boldsymbol{v}}_j^{(\ell)}\|_2 = 1$ (together with the deflation/orthogonality constraints to previously selected directions, which are also expressible solely through such inner products). Decompose $\hat{\boldsymbol{v}}_j^{(\ell)} = \boldsymbol{v}_j^{\parallel} + \boldsymbol{v}_j^{\perp}$, where $\boldsymbol{v}_j^{\parallel}$ is the orthogonal projection onto the finite-dimensional subspace $\mathcal{S} := \mathrm{span}\{\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\}_{\mu=1}^n \subset \mathbb{R}^{p_{\ell-1}}$ and $\boldsymbol{v}_j^{\perp} \in \mathcal{S}^{\perp}$. By orthogonality, $\langle \boldsymbol{v}_j^{\perp}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle = 0$ for every training input, so $\langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle = \langle \boldsymbol{v}_j^{\parallel}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle$, i.e. the data-dependent objective depends only on $\boldsymbol{v}_j^{\parallel}$. Pythagoras gives $1 = \|\hat{\boldsymbol{v}}_j^{(\ell)}\|_2^2 = \|\boldsymbol{v}_j^{\parallel}\|_2^2 + \|\boldsymbol{v}_j^{\perp}\|_2^2$, so any nonzero $\boldsymbol{v}_j^{\perp}$ forces $\|\boldsymbol{v}_j^{\parallel}\|_2 < 1$. The rescaled vector $\tilde{\boldsymbol{v}}_j := \boldsymbol{v}_j^{\parallel}/\|\boldsymbol{v}_j^{\parallel}\|_2 \in \mathcal{S}$ is then feasible (it has unit norm and inherits the orthogonality constraints, since these involve only inner products with vectors in $\mathcal{S}$) and yields a strictly larger objective by a factor $\|\boldsymbol{v}_j^{\parallel}\|_2^{-2} > 1$, contradicting optimality of $\hat{\boldsymbol{v}}_j^{(\ell)}$. Hence every minimizer satisfies $\boldsymbol{v}_j^{\perp} = 0$, i.e. $\hat{\boldsymbol{v}}_j^{(\ell)} \in \mathcal{S}$, which is (94). □

F.3 RKHS-Euclidean Equivalence

We now establish a fundamental result that underlies all subsequent proofs in this section. By Lemma 1, we know features lie in the span of training features. This allows us to identify RKHS constraints with Euclidean constraints on weight vectors.

Lemma 2 (RKHS-Euclidean Norm Equivalence)**.** *Let $K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{z}_{\ell-1}(\boldsymbol{x}), \boldsymbol{z}_{\ell-1}(\boldsymbol{x}')\rangle$ and let $\mathcal{H}_{\ell-1}$ be its induced RKHS. Then the RKHS norm of a linear feature coincides with the Euclidean norm of its weight vector: for any $\boldsymbol{u} \in \mathbb{R}^{p_{\ell-1}}$, the linear feature*

$$\varphi_{\boldsymbol{u}}(\boldsymbol{x}) := \langle \boldsymbol{u}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\rangle.$$

satisfies

$$\|\varphi_{\boldsymbol{u}}\|_{\mathcal{H}_{\ell-1}} = \|\boldsymbol{u}\|_2.$$

Consequently, the RKHS unit ball $\{\varphi : \|\varphi\|_{\mathcal{H}_{\ell-1}} \le 1\}$ and the Euclidean unit sphere $\{\boldsymbol{u} : \|\boldsymbol{u}\|_2 = 1\}$ are in bijection through $\boldsymbol{u} \mapsto \varphi_{\boldsymbol{u}}$, establishing a primal–dual equivalence between RKHS constraints and Euclidean constraints.

F.4 Proof of Theorem 1

We prove each part of Theorem 1 in turn, using Lemma 2 to convert between the RKHS and Euclidean formulations.

Proof of part (i). By Lemma 2, the objective equals

$$\left|\widehat{\mathbb{E}}_n\big[y\,\psi(\boldsymbol{x})\big]\right| = \left|\frac{1}{n}\sum_{\mu=1}^{n} y_\mu \langle \boldsymbol{u}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle\right| = \left|\Big\langle \boldsymbol{u},\ \hat{\boldsymbol{u}}^\ell \Big\rangle\right|,$$

where $\hat{\boldsymbol{u}}^\ell = \frac{1}{n}\sum_{\mu=1}^{n} y_\mu \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)$ is the empirical first-order moment computed in Algorithm 1. By the Cauchy–Schwarz inequality this is maximized at $\boldsymbol{u} = \hat{\boldsymbol{u}}^\ell/\|\hat{\boldsymbol{u}}^\ell\|_2$, giving $\psi^\ell(\boldsymbol{x}) = \langle \hat{\boldsymbol{u}}^\ell, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\rangle$ as the unique maximizer (up to sign). □

Proof of part (ii). By Lemma 2, the second-order objective equals

$$\left|\widehat{\mathbb{E}}_n\big[y\,\varphi(\boldsymbol{x})^2\big]\right| = \left|\frac{1}{n}\sum_{\mu=1}^{n} y_\mu \langle \boldsymbol{u}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle^2\right| = \left|\boldsymbol{u}^\top \widehat{\boldsymbol{C}}^{(\ell)} \boldsymbol{u}\right|,$$

where $\widehat{\boldsymbol{C}}^{(\ell)} = \frac{1}{n}\sum_{\mu=1}^{n} y_\mu\, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top$ is the empirical second-order moment operator. By the Courant–Fischer minimax theorem, the unit-norm vector maximizing $|\boldsymbol{u}^\top \widehat{\boldsymbol{C}}^{(\ell)}\boldsymbol{u}|$ is the leading eigenvector $\hat{\boldsymbol{v}}_1^{(\ell)}$ of $\widehat{\boldsymbol{C}}^{(\ell)}$ (in absolute eigenvalue). Successive orthogonal maximizers are the subsequent eigenvectors $\hat{\boldsymbol{v}}_2^{(\ell)}, \ldots, \hat{\boldsymbol{v}}_{k_\ell}^{(\ell)}$. Via $\boldsymbol{u} \mapsto \varphi_{\boldsymbol{u}}$, these correspond exactly to $\varphi_1, \ldots, \varphi_{k_\ell}$ satisfying the stated variational recursion. □

Dual (kernel-space) form By Lemma 1, any optimal feature φ_j lies in the span of the n training feature vectors:

$$\varphi_j(\boldsymbol{x}) = \sum_{\mu=1}^{n} \alpha_{j,\mu} K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}_\mu). \tag{95}$$

Define $\boldsymbol{\alpha}_j = (\alpha_{j,1}, \ldots, \alpha_{j,n})^\top$ as the coefficient vector for feature j. By Lemma 2, the RKHS norm of this feature is

$$\|\varphi_j\|^2_{\mathcal{H}_{\ell-1}} = \boldsymbol{\alpha}_j^\top G_{\ell-1} \boldsymbol{\alpha}_j, \tag{96}$$

where $G_{\ell-1} = Z_{\ell-1} Z_{\ell-1}^\top \in \mathbb{R}^{n\times n}$ is the Gram matrix with entries $G_{\ell-1}(\mu,\nu) = K_{\ell-1}(\boldsymbol{x}_\mu, \boldsymbol{x}_\nu)$.

The empirical objective from Theorem 1 (ii) becomes

$$\max_{\boldsymbol{\alpha}:\boldsymbol{\alpha}^\top G_{\ell-1}\boldsymbol{\alpha}=1} \left|\frac{1}{n}\boldsymbol{\alpha}^\top G_{\ell-1} Y G_{\ell-1}\boldsymbol{\alpha}\right|, \tag{97}$$

where $Y = \mathrm{diag}(y_1, \ldots, y_n)$ and we used that the vector of feature evaluations on the training data is $G_{\ell-1}\boldsymbol{\alpha}$.

Hence, we obtain the following proposition.

Proposition 3 (Generalized Eigenvector Problem)**.** *The optimal dual coefficients $\{\hat{\boldsymbol{\alpha}}_j\}_{j=1}^{k_\ell}$ that sequentially maximize the above constrained objective can be chosen on the range of $G_{\ell-1}$ to satisfy*

$$\frac{1}{n} G_{\ell-1} Y G_{\ell-1} \hat{\boldsymbol{\alpha}}_j = \lambda_j G_{\ell-1} \hat{\boldsymbol{\alpha}}_j, \qquad j = 1, \ldots, k_\ell, \tag{98}$$

with $\hat{\boldsymbol{\alpha}}_i^\top G_{\ell-1} \hat{\boldsymbol{\alpha}}_j = \delta_{ij}$, where the generalized eigenvalues λ_j are ordered by magnitude. These coefficients define the second-order features without requiring explicit access to the feature vectors.

Proof of part (iii). At every layer, assume that the limiting eigenvalue (or eigenvalue cluster) being selected is separated from the rest of the spectrum. For a single feature this means a positive eigengap; for a cluster the statements below hold for the spectral projector, with an arbitrary orthonormal basis chosen inside the limiting eigenspace.

Induction statement. Let $\rho_{\boldsymbol{x}}$ be the data distribution. After layer r, let

$$\boldsymbol{g}_r^{(p)}(\boldsymbol{x}) = \big(\psi_r^{(p)}(\boldsymbol{x}), \varphi_{r,1}^{(p)}(\boldsymbol{x}), \ldots, \varphi_{r,k_r}^{(p)}(\boldsymbol{x})\big)$$

denote the projected features selected by Neural LoFi at finite width, and let $K_r^{(p)}$ be the kernel produced by the following random lift. The induction claim I_r is that there exist deterministic functions $\boldsymbol{g}_r^\infty$ and a deterministic kernel K_r^∞ such that

$$\|\boldsymbol{g}_r^{(p)} - \boldsymbol{g}_r^\infty\|_{L^2(\rho_{\boldsymbol{x}})} \xrightarrow[p\to\infty]{P} 0 \tag{99}$$

and, for the training inputs,

$$\|\boldsymbol{G}_r^{(p)} - \boldsymbol{G}_r^{\infty}\|_{\mathrm{op}} \xrightarrow[p\to\infty]{P} 0, \qquad \sum_{\mu=1}^{n} \|K_r^{(p)}(\cdot, \boldsymbol{x}_\mu) - K_r^{\infty}(\cdot, \boldsymbol{x}_\mu)\|_{L^2(\rho_{\boldsymbol{x}})}^2 \xrightarrow[p\to\infty]{P} 0, \tag{100}$$

where $(\boldsymbol{G}_r^{(p)})_{\mu\nu} = K_r^{(p)}(\boldsymbol{x}_\mu, \boldsymbol{x}_\nu)$. The base case is deterministic: $\boldsymbol{g}_0(\boldsymbol{x}) = \boldsymbol{x}$ and $K_0(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle$.

We first record the kernel-propagation step used in the induction. Suppose that, for some layer r, the projected features already satisfy $\boldsymbol{g}_r^{(p)} \to \boldsymbol{g}_r^{\infty}$ in $L^2(\rho_{\boldsymbol{x}})$. For a deterministic feature map $\boldsymbol{g}$, write

$$\mathcal{K}_r[\boldsymbol{g}](\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{a}\sim\pi_r}\big[\sigma(\boldsymbol{a}^\top \boldsymbol{g}(\boldsymbol{x}))\sigma(\boldsymbol{a}^\top \boldsymbol{g}(\boldsymbol{x}'))\big].$$

The limiting next-layer kernel is $K_r^{\infty} = \mathcal{K}_r[\boldsymbol{g}_r^{\infty}]$. Since $\boldsymbol{g}_r^{(p)} \to \boldsymbol{g}_r^{\infty}$ in L^2 and σ is pseudo-Lipschitz with the required moment bounds, dominated convergence gives convergence of the deterministic kernels $\mathcal{K}_r[\boldsymbol{g}_r^{(p)}] \to \mathcal{K}_r[\boldsymbol{g}_r^{\infty}]$ on the fixed Gram matrix and in the L^2 section norm in (100). Conditional on $\boldsymbol{g}_r^{(p)}$, the finite-width kernel is an average of iid random features:

$$K_r^{(p)}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{p_r}\sum_{i=1}^{p_r} \sigma(\boldsymbol{a}_i^\top \boldsymbol{g}_r^{(p)}(\boldsymbol{x}))\sigma(\boldsymbol{a}_i^\top \boldsymbol{g}_r^{(p)}(\boldsymbol{x}')).$$

For each fixed anchor $\boldsymbol{x}_\mu$,

$$\mathbb{E}_{\boldsymbol{a}}\big\|K_r^{(p)}(\cdot, \boldsymbol{x}_\mu) - \mathcal{K}_r[\boldsymbol{g}_r^{(p)}](\cdot, \boldsymbol{x}_\mu)\big\|_{L^2(\rho_{\boldsymbol{x}})}^2 \le \frac{C_\mu}{p_r}, \tag{101}$$

and the same iid law of large numbers applies to the finitely many Gram entries. Thus (100) follows. This is the formal sense in which, once the features entering a layer have a deterministic limit, the next layer sees a fixed deterministic distribution.

It remains to show that this deterministic kernel convergence propagates through the spectral filtering step. Assume $\mathsf{I}_{\ell-1}$ and abbreviate $K_p = K_{\ell-1}^{(p)}$, $K_\infty = K_{\ell-1}^{\infty}$, and $\boldsymbol{G}_p, \boldsymbol{G}_\infty$ for their Gram matrices. By (100),

$$\|\boldsymbol{G}_p - \boldsymbol{G}_\infty\|_{\mathrm{op}} \xrightarrow[p\to\infty]{P} 0, \qquad \sum_{\mu=1}^{n} \|K_p(\cdot, \boldsymbol{x}_\mu) - K_\infty(\cdot, \boldsymbol{x}_\mu)\|_{L^2(\rho_{\boldsymbol{x}})}^2 \xrightarrow[p\to\infty]{P} 0. \tag{102}$$

The explicit $O_P(n/\sqrt{p})$ Gram-matrix rate in the non-adaptive, fixed-$\boldsymbol{g}$ case is the usual random-feature concentration bound; for the section convergence, (101) gives the sharper Hilbert-space law of large numbers needed for out-of-sample features.

By Lemma 1, every selected feature has the form

$$\varphi(\boldsymbol{x}) = \sum_{\mu=1}^{n} \alpha_\mu K_p(\boldsymbol{x}, \boldsymbol{x}_\mu).$$

Its RKHS norm and empirical second-order objective are

$$\|\varphi\|_{\mathcal{H}_p}^2 = \boldsymbol{\alpha}^\top \boldsymbol{G}_p \boldsymbol{\alpha}, \qquad \widehat{\mathbb{E}}_n[y\varphi(\boldsymbol{x})^2] = \frac{1}{n}\boldsymbol{\alpha}^\top \boldsymbol{G}_p Y \boldsymbol{G}_p \boldsymbol{\alpha}, \qquad Y = \mathrm{diag}(y_1, \dots, y_n).$$

Therefore, on the range of $\boldsymbol{G}_p$, the generalized eigenproblem in (98) is equivalently the symmetric eigenproblem

$$\boldsymbol{B}_p \boldsymbol{\beta} = \lambda \boldsymbol{\beta}, \qquad \boldsymbol{B}_p = \frac{1}{n}\boldsymbol{G}_p^{1/2} Y \boldsymbol{G}_p^{1/2}, \qquad \boldsymbol{\beta} = \boldsymbol{G}_p^{1/2}\boldsymbol{\alpha}. \tag{103}$$

The square-root map is continuous on positive semidefinite matrices, hence $\|\boldsymbol{B}_p - \boldsymbol{B}_\infty\|_{\mathrm{op}} \to 0$ in probability. If the kth limiting eigenvalue is isolated with gap $\Delta_k > 0$, the Davis–Kahan $\sin\Theta$ theorem [120] implies convergence of the corresponding projectors at rate $O_P(\|\boldsymbol{B}_p - \boldsymbol{B}_\infty\|_{\mathrm{op}}/\Delta_k)$. For a simple eigenvalue, after fixing the sign,

$$\boldsymbol{\beta}_{p,k} \xrightarrow[p\to\infty]{P} \boldsymbol{\beta}_{\infty,k}. \tag{104}$$

We next convert the convergence of $\boldsymbol{\beta}_{p,k}$ into convergence of the actual feature functions. Let

$$\boldsymbol{k}_p(\boldsymbol{x}) = \big(K_p(\boldsymbol{x}, \boldsymbol{x}_1), \dots, K_p(\boldsymbol{x}, \boldsymbol{x}_n)\big)^\top, \qquad \boldsymbol{\alpha}_{p,k} = \boldsymbol{G}_p^{\dagger/2} \boldsymbol{\beta}_{p,k},$$

where $\dagger$ denotes the Moore–Penrose inverse on the stable range of the limiting Gram matrix. If $\boldsymbol{G}_\infty$ is nonsingular, this is the ordinary inverse square root. Continuity of the pseudo-inverse operator and (104) give $\boldsymbol{\alpha}_{p,k} \to \boldsymbol{\alpha}_{\infty,k}$ in probability. Hence, with

$$\widehat{\varphi}_{p,k}(\boldsymbol{x}) = \boldsymbol{k}_p(\boldsymbol{x})^\top \boldsymbol{\alpha}_{p,k}, \qquad \phi_k^\infty(\boldsymbol{x}) = \boldsymbol{k}_\infty(\boldsymbol{x})^\top \boldsymbol{\alpha}_{\infty,k},$$

we have

$$\begin{aligned} \|\widehat{\varphi}_{p,k} - \phi_k^\infty\|_{L^2(\rho_{\boldsymbol{x}})} \le{}& \|\boldsymbol{k}_p - \boldsymbol{k}_\infty\|_{L^2(\rho_{\boldsymbol{x}};\mathbb{R}^n)} \, \|\boldsymbol{\alpha}_{p,k}\|_2 \\ &+ \|\boldsymbol{k}_\infty\|_{L^2(\rho_{\boldsymbol{x}};\mathbb{R}^n)} \, \|\boldsymbol{\alpha}_{p,k} - \boldsymbol{\alpha}_{\infty,k}\|_2 \xrightarrow[p\to\infty]{P} 0. \end{aligned} \tag{105}$$

The linear feature satisfies the same conclusion directly, since

$$\psi_p^\ell(\boldsymbol{x}) = \frac{1}{n} \sum_{\mu=1}^n y_\mu K_p(\boldsymbol{x}, \boldsymbol{x}_\mu) \to \frac{1}{n} \sum_{\mu=1}^n y_\mu K_\infty(\boldsymbol{x}, \boldsymbol{x}_\mu) = \psi_\infty^\ell(\boldsymbol{x})$$

in $L^2(\rho_{\boldsymbol{x}})$. Thus the whole projected vector $\boldsymbol{g}_\ell^{(p)} = (\psi_p^\ell, \widehat{\varphi}_{p,1}, \dots, \widehat{\varphi}_{p,k_\ell})$ converges to the deterministic vector $\boldsymbol{g}_\ell^\infty$ in L^2. Applying the first part of the induction to this deterministic limit gives convergence of the next random feature kernel $K_\ell^{(p)}$ to

$$K_\ell^\infty(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{a}\sim\pi_\ell}\left[\sigma(\boldsymbol{a}^\top \boldsymbol{g}_\ell^\infty(\boldsymbol{x}))\sigma(\boldsymbol{a}^\top \boldsymbol{g}_\ell^\infty(\boldsymbol{x}'))\right],$$

which proves I_ℓ. By induction, all finite collections of learned features and the induced kernels converge layer by layer to deterministic infinite-width limits.

□

G Effective dimension and sample complexity of emergence

As discussed in the main text, training dynamics is found in practice to often display long plateaus followed by abrupt transitions, with new directions in representation space *emerging* sequentially [29–32]. We return in this section to the emergence of learned features at each layer, and in particular, we now make the noise level $\tau_\ell^k(n)$ in (17) explicit. Throughout the section our results apply conditioned on a fixed kernel $K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}')$ defined by the features $\boldsymbol{z}_{\ell-1}$ of the previous layers.

Recall the residual candidate class

$$\mathcal{S}_k^\ell = \left\{ \varphi \in \mathcal{H}_{\ell-1} : \|\varphi\|_{\mathcal{H}_{\ell-1}} = 1,\ \varphi \perp \varphi_1, \cdots, \varphi_{k-1} \right\}.$$

Let $T_{\ell-1} : L^2(P_x) \to L^2(P_x)$ be the integral operator of the current kernel,

$$(T_{\ell-1} f)(\boldsymbol{x}) = \int K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}')\, dP_x(\boldsymbol{x}'),$$

and let $\Pi_k^\perp$ denote the projection orthogonal to the selected features $\varphi_1, \ldots, \varphi_{k-1}$. The residual covariance operator is

$$T_{\ell,k} := \Pi_k^\perp T_{\ell-1} \Pi_k^\perp.$$

For a resolution parameter $r > 0$, define the *residual effective dimension*:

Definition 1 (Residual effective dimension)**.** *Let $\{\lambda_j^{\ell,k}\}_{j\geq 1}$ denote the eigenvalues of $T_{\ell,k}$. The* effective residual dimension *at scale r is defined as*

$$D_k^\ell(r) := \mathrm{Tr}\Big[T_{\ell,k}(T_{\ell,k} + rI)^{-1} \Big] = \sum_{j\geq 1} \frac{\lambda_j^{\ell,k}}{\lambda_j^{\ell,k} + r}, \tag{106}$$

The above notion of effective dimension is standard in kernel regression [37,38], and similar quantities appear in local Rademacher analyses of kernel classes [35,36]. $D_k^\ell(r)$ can be interpreted as counting the number of residual directions whose variance is above the resolution r.

Assumption 3 (Residual eigenfunction hypercontractivity)**.** *Let $\mathcal{E}_{\ell,k}$ denote the $L^2(P_x)$-closed span of the residual eigenfunctions $\{\phi_j^{\ell,k}\}_{j\geq 1}$. There exists $H_\ell < \infty$ such that every $g \in \mathcal{E}_{\ell,k}$ satisfies*

$$\|g\|_{L^4(P_x)} \leq H_\ell \|g\|_{L^2(P_x)}.$$

Equivalently, the same inequality holds for every L^2-convergent expansion $g = \sum_j c_j \phi_j^{\ell,k}$ with $\sum_j c_j^2 < \infty$.

Such an assumption holds, in particular, for the polynomial eigenbases that appear in random-feature kernels such as Hermite polynomials and spherical harmonics ([46]).

Theorem 5 (Emergence sample complexity)**.** *Suppose that Assumption 3 holds and the kernel $K_{\ell-1}$ and labels y are uniformly bounded. Let*

$$r_\ell^{k,\star} := \operatorname*{arg\,max}_{r:\, r \leq \lambda_1^{\ell,k}} r\sqrt{D_k^\ell(r)}.$$

Here $\lambda_1^{\ell,k}$ is the top residual eigenvalue, hence the largest possible variance scale of a unit-RKHS candidate in $\mathcal{S}_k^\ell$. Then, for any $\delta > 0$, there is a constant $\widetilde{C}_{\ell,H} > 0$ such that with probability at least $1 - \delta$,

$$\tau_\ell^k(n) := \sup_{\varphi \in \mathcal{S}_k^\ell} \left| \widehat{c}_{\ell,n}(\varphi) - c_\ell(\varphi) \right| \leq \widetilde{C}_{\ell,H}\, r_\ell^{k,\star} \sqrt{\frac{D_k^\ell(r_\ell^{k,\star})}{n}}. \tag{107}$$

This gives, up to polylogarithmic factors, the sample scale

$$n_k \gtrsim \frac{(r_\ell^{(k)})^2 D_k^\ell(r_\ell^{(k)})}{(\rho_\ell^{(k)})^2},$$

for recovery of φ_k conditional on the recovery of previous features $\varphi_1, \cdots, \varphi_{k-1}$.

Remark 2. *The order of maximizers of the noise scale $r\sqrt{D_k^\ell(r)}$ may not perfectly match the order of maximizers of the population correlation $\left|\mathbb{E}[y\varphi(\boldsymbol{x})^2]\right|$. Hence, the next emergent feature after recovering $\varphi_1, \cdots, \varphi_{k-1}$ may not be φ_k. For simplicity, we neglect the correction in sample complexity due to such order switches.*

Proof. For $r > 0$, let

$$\mathcal{S}_k^\ell(r) := \{\varphi \in \mathcal{S}_k^\ell : \mathbb{E}[\varphi(\boldsymbol{X})^2] \le r\},$$

We proceed by decomposing the fluctuations into contributions from $S_k^\ell(r)$ for different variance scales r. Define

$$\tau_\ell^k(r,n) := \sup_{\varphi \in \mathcal{S}_k^\ell(r)} \left|\hat{\mathbb{E}}[y\varphi^2] - \mathbb{E}[y\varphi^2]\right|. \tag{108}$$

We will show that:

$$\tau_\ell^k(r,n) \lesssim \widetilde{C}_{\ell,H}\, r\sqrt{\frac{D_k^\ell(r)}{n}} \tag{109}$$

The above bound follows through an extension of Theorem 2.1 in [35] which bounds the rademacher complexity of $S_k^\ell(r)$. Concretely, Theorem 2.1 in [35] provides that for a unit ball of an RKHS of a bounded kernel with eigenvalues $\{\lambda_j\}_{j\ge1}$:

$$\mathbb{E}R_n\left\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \le 1,\, \mathbb{E}f(\boldsymbol{X})^2 \le r\right\} \lesssim \left(\frac{1}{n}\sum_{j\ge1}\min\{r,\lambda_j\}\right)^{1/2}.$$

Since for every $r > 0$ and every eigenvalue λ_j,

$$\frac{1}{2}\min\{r,\lambda_j\} \le \frac{r\lambda_j}{r+\lambda_j} \le \min\{r,\lambda_j\},$$

the same bound holds for $D_k^\ell(r)$ defined in Definition 1.

We now adapt the proof of Theorem 2.1 of [35] to the quadratic process. By the boundedness of the labels y, $\tau_\ell^k(r,n)$ can be bounded up to constants by the rademacher complexity of φ^2 over $S_k^\ell(r)$, defined as:

$$\mathbb{E}_\epsilon[\sup_{\varphi\in\mathcal{S}_k^\ell(r)}\left|\sum_{i=1}^n \epsilon_i\varphi(\boldsymbol{X}_i)^2\right|], \tag{110}$$

where $\epsilon \in \{\pm1\}$ denote standard rademacher random variables.

Let $\{\phi_j^{\ell,k}\}_{j\ge1}$ denote the $L^2(P_x)$-orthonormal eigenfunctions of $T_{\ell,k}$ Any φ with $\|\varphi\|_{\mathcal{H}} = 1$ can be expressed in the basis w.r.t $\{\phi_j^{\ell,k}\}_{j\ge1}$ as

$$\varphi_\beta(\boldsymbol{x}) = \sum_{j\ge1}\beta_j\sqrt{\lambda_j^{\ell,k}}\,\phi_j^{\ell,k}(\boldsymbol{x}), \qquad \sum_j \beta_j^2 \le 1.$$

The constraint $\mathbb{E}[\varphi(\boldsymbol{X})^2] \le r$ then translates to $\sum_j \lambda_j^{\ell,k}\beta_j^2 \le r$. Hence, the set of coefficients for $\varphi \in S_k^\ell(r)$ is given by:

$$I_r := \left\{\beta : \sum_j \beta_j^2 \le 1, \quad \sum_j \lambda_j^{\ell,k}\beta_j^2 \le r\right\}.$$

The proof of Theorem 2.1 in [35] relates the set I_r to an ellipsoid B_r through the following inclusions:

$$B_r \subset I_r \subset \sqrt{2}\,B_r, \qquad B_r := \left\{\beta : \sum_j \mu_j(r)\beta_j^2 \le 1\right\}, \qquad \mu_j(r) = \left(\min\{1, r/\lambda_j^{\ell,k}\}\right)^{-1}.$$

Thus the supremum over I_r is bounded, up to an absolute constant, by the supremum over B_r. Setting $\theta_j = \sqrt{\mu_j(r)}\,\beta_j$, B_r is re-parameterized as the Euclidean unit ball $\|\theta\|_2 \le 1$. Next, define

$$a_j(r) \coloneqq \frac{\lambda_j^{\ell,k}}{\mu_j(r)} = \min\{\lambda_j^{\ell,k}, r\}$$

The decomposition $\varphi_\beta(\boldsymbol{x}) = \sum_{j\ge1} \beta_j \sqrt{\lambda_j^{\ell,k}}\,\phi_j^{\ell,k}(\boldsymbol{x})$ can then be re-expressed as:

$$\begin{aligned}
&\sum_{j\ge1} \beta_j \sqrt{\lambda_j^{\ell,k}}\,\phi_j^{\ell,k}(\boldsymbol{x}) \\
&= \sum_j \sqrt{\mu_j(r)}\,\beta_j \frac{1}{\sqrt{\mu_j(r)}} \phi_j^{\ell,k}(\boldsymbol{x}) \\
&= \sum_j \theta_j \sqrt{a_j(r)} \phi_j^{\ell,k}(\boldsymbol{x})
\end{aligned}$$

Hence,

$$\sup_{\beta\in I_r} \left| \sum_{i=1}^n \epsilon_i \varphi_\beta(\boldsymbol{X}_i)^2 \right| \lesssim \sup_{\|\theta\|_2\le1} \left| \sum_{i=1}^n \epsilon_i \left(\sum_j \theta_j \sqrt{a_j(r)} \phi_j^{\ell,k}(\boldsymbol{X}_i) \right)^2 \right|. \tag{111}$$

Define:

$$v_{\ell,k,r}(\boldsymbol{x}) \coloneqq \left(\sqrt{a_j(r)}\,\phi_j^{\ell,k}(\boldsymbol{x}) \right)_{j\ge1},$$

then the RHS in Equation 111 can be expressed as:

$$\left\| \sum_{i=1}^n \epsilon_i v_{\ell,k,r}(\boldsymbol{X}_i) \otimes v_{\ell,k,r}(\boldsymbol{X}_i) \right\|_{\mathrm{op}}. \tag{112}$$

Let

$$\psi_{\ell,k}(r)^2 \coloneqq \sum_j a_j(r) \asymp r D_k^\ell(r).$$

We now bound Equation 112 through a matrix concentration argument which requires bounding the operator norm of the following fourth-moment matrix:

$$M_{\ell,k}(r) \coloneqq \mathbb{E}\Big[\big\| v_{\ell,k,r}(\boldsymbol{X}) \big\|_2^2\, v_{\ell,k,r}(\boldsymbol{X}) \otimes v_{\ell,k,r}(\boldsymbol{X}) \Big].$$

We bound $\big\|M_{\ell,k}(r)\big\|$ through Assumption 3. For every $\|u\|_2 = 1$, write $g_u(\boldsymbol{x}) = \langle u, v_{\ell,k,r}(\boldsymbol{x})\rangle$. Then

$$\mathbb{E}[g_u(\boldsymbol{X})^2] = \sum_j u_j^2 a_j(r) \le r,$$

since $a_j(r) \le r$. Subsequently by Cauchy–Schwarz and hypercontractivity (Assumption 3), we obtain:

$$\begin{aligned}
\big\langle u, M_{\ell,k}(r) u \big\rangle &= \sum_j a_j(r)\, \mathbb{E}\Big[\big(\phi_j^{\ell,k}(\boldsymbol{X})\big)^2 g_u(\boldsymbol{X})^2 \Big] \\
&\le \sum_j a_j(r) \left\| \phi_j^{\ell,k} \right\|_{L^4(P_x)}^2 \|g_u\|_{L^4(P_x)}^2 \\
&\le H_\ell^4 \psi_{\ell,k}(r)^2 \mathbb{E}[g_u(\boldsymbol{X})^2] \le H_\ell^4 r\, \psi_{\ell,k}(r)^2.
\end{aligned}$$

Therefore

$$\big\|M_{\ell,k}(r)\big\|_2 \lesssim H_\ell^4 r\, \psi_{\ell,k}(r)^2 \asymp H_\ell^4 r^2 D_k^\ell(r).$$

Matrix Bernstein then gives, up to constants and polylogarithmic factors:

$$\mathbb{E}_\epsilon \sup_{\varphi\in\mathcal{S}_k^\ell(r)} \left| \sum_{i=1}^n \epsilon_i \varphi(\boldsymbol{X}_i)^2 \right| \lesssim H_\ell^2 \sqrt{nr}\, \psi_{\ell,k}(r) \asymp H_\ell^2 nr \sqrt{\frac{D_k^\ell(r)}{n}}.$$

By symmetrization and the boundedness of labels, this yields (109). Taking the maximum over shells gives the stated bound with $r_\ell^{k,\star}$. □

G.1 Estimating $\tau_\ell^k(n)$ in Feature space

We discuss here how $\tau_\ell^k(n)$ can be estimated directly in the feature space (rather than in function space, as we did so far).

Define the true label, weighted covariance:

$$\boldsymbol{C}^{(\ell)} \coloneqq \mathbb{E}[y\, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\boldsymbol{z}_{\ell-1}(\boldsymbol{x})^\top], \tag{113}$$

and the unweighted covariance:

$$\boldsymbol{\Sigma}^{(\ell)} \coloneqq \mathbb{E}[\boldsymbol{z}_{\ell-1}(\boldsymbol{x})\boldsymbol{z}_{\ell-1}(\boldsymbol{x})^\top]. \tag{114}$$

To estimate the cutoff n_ℓ^k for the number of samples required for the recovery of ϕ_k, we proceed as follows:

(i) Let $\boldsymbol{v}_1, \dots \boldsymbol{v}_{k-1}$ denote the top $k-1$ eigenvectors of $\boldsymbol{C}^{(\ell)}$. Compute the residual covariance:

$$\boldsymbol{\Sigma}_k^{(\ell)} \coloneqq (I - \boldsymbol{v}_{k-1}\boldsymbol{v}_{k-1}^\top)\boldsymbol{\Sigma}_{k-1}^{(\ell)}(I - \boldsymbol{v}_{k-1}\boldsymbol{v}_{k-1}^\top). \tag{115}$$

(ii) Let $\{\lambda_j^{\ell,k}\}_{j=1}^\infty$ denote the eigenvalues of $\boldsymbol{\Sigma}_k^{(\ell)}$. Then, for any $r \in \mathbb{R}$, define:

$$D_\ell^k(r) \coloneqq \sum_{j\geq 1} \frac{\lambda_j^{\ell,k}}{\lambda_j^{\ell,k} + r}. \tag{116}$$

(iii) Set $r_\ell^k = \arg\max_{r \leq \lambda_1^{\ell,k}} r\sqrt{D_\ell^k(r)}$. We predict the emergence of $\boldsymbol{v}_k$ with samples n_ℓ^k satisfying:

$$n_\ell^k \sim \left(\frac{r_\ell^k}{\rho_\ell^{(k)}}\right)^2 D_\ell^k, \tag{117}$$

where $\rho_\ell^{(k)}$ is the k_{th} eigenvalue of $\boldsymbol{C}^{(\ell)}$.

G.2 Effective dimension in the Hermite toy model

The abstract quantity $D_k^\ell(r)$ becomes concrete in the toy model Appendix E. In this paragraph, k denotes the Hermite degree used to define the first hidden representation $h^{(1)}$, not the index of the sequentially selected feature. Write the flattened degree-k Hermite tensor as

$$H_k(\boldsymbol{x}) = \{H_\alpha(\boldsymbol{x}) : |\alpha| = k\} \in \mathbb{R}^{D_k}, \qquad D_k = \binom{d+k-1}{k}.$$

For Gaussian inputs, these coordinates are orthonormal in $L^2(P_x)$. Hence, for any two coefficient vectors $a, b \in \mathbb{R}^{D_k}$,

$$\mathbb{E}\big[\langle a, H_k(\boldsymbol{X})\rangle\langle b, H_k(\boldsymbol{X})\rangle\big] = \langle a, b\rangle_{\mathbb{R}^{D_k}}.$$

Thus a degree-k candidate feature $\varphi_a(\boldsymbol{x}) = \langle a, H_k(\boldsymbol{x})\rangle$ is literally a vector in a D_k-dimensional Euclidean feature space as far as $L^2(P_x)$ norms and projections are concerned.

Equivalently, the degree-k Hermite kernel

$$K_{H,k}(\boldsymbol{x}, \boldsymbol{x}') = \langle H_k(\boldsymbol{x}), H_k(\boldsymbol{x}')\rangle = \sum_{|\alpha|=k} H_\alpha(\boldsymbol{x})H_\alpha(\boldsymbol{x}')$$

has integral operator

$$T_{H,k}f = \sum_{|\alpha|=k} \langle f, H_\alpha\rangle_{L^2(P_x)} H_\alpha.$$

This is the orthogonal projector onto the degree-k Hermite chaos. Its only nonzero eigenvalue is 1, with multiplicity D_k. If the same block is reached through a random-feature activation, as in the construction of Subsection E.4, the eigenvalue is multiplied by the squared Hermite coefficient and by normalization constants, but the multiplicity is still at most D_k.

Therefore, effective dimension relevant for the recovery of $h^{(1)}$ is bounded by this Hermite block rank. For a block eigenvalue $a_k > 0$,

$$D^{(k)}_{\mathrm{toy},1}(r) = \sum_{j=1}^{D_k} \frac{a_k}{a_k + r} = \frac{a_k}{a_k + r} D_k \le D_k = \binom{d+k-1}{k} = O(d^k),$$

and for resolutions $\lambda \lesssim a_k$ this is $\asymp D_k$. Hence, during first-level feature recovery, the statistical fluctuation is governed by the ambient degree-k Hermite space.

The planted first-level variables in Appendix E,

$$h^{(1)}_i(\boldsymbol{x}) = \langle A^{(1)}_i, H_k(\boldsymbol{x}) \rangle, \qquad i = 1, \dots, d_1,$$

form a $d_1 = d^\epsilon$ dimensional signal subspace inside this $D_k = O(d^k)$ dimensional Hermite block. The residual effective dimension controls the size of the empirical noise over the full candidate block, while the planted subspace controls the rank and strength of the population signal. Concretely, by the definition of y, we have that:

$$\mathbb{E}[y(h^{(1)}_i(\boldsymbol{x}))^2] \sim \frac{1}{\sqrt{d_1}} \tag{118}$$

Substituting the above scaling into the sample complexity prediction prescribed by Theorem 5 gives the $d^{k+\epsilon}$ first-stage sample scale stated in Appendix E.

G.3 Effective dimension in the multi-index models

We now discuss how the sample complexity prediction in Proposition 5 recovers the rates in the setting of multi-index models with power-law dependence on the label.

We follow here [39, 82] and consider hierarchical multi-index target $y = f^*(\langle \boldsymbol{w}^*_1, \boldsymbol{x} \rangle, \dots, \langle \boldsymbol{w}^*_r, \boldsymbol{x} \rangle) + \xi$ with isotropic Gaussian inputs $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and orthonormal teacher directions $\{\boldsymbol{w}^*_i\}_{i=1}^r$. Because the input covariance is the identity $D^\ell \asymp d$, at the slice-wise scale used in Proposition 5. The features at layer ℓ along each candidate direction have unit variance under the Gaussian input, so the variance scale is order one, $\tau = r^{(k)}_\ell = O(1)$. Finally, the power-law dependence of the label on the teacher coefficients translates into a power-law decay of the population correlations across the planted directions: the i-th spike satisfies $\rho^{(i)}_\ell = O(i^{-\gamma})$, where $\gamma > 0$ is the exponent governing the label spectrum. Plugging $D^\ell_k \asymp d$, $\tau = O(1)$ and $\rho^{(i)} = O(i^{-\gamma})$ into the recovery condition $n \gg (r^{(k)}_\ell)^2 D^\ell_k / (\rho^{(k)}_\ell)^2$ of Proposition 5 yields the sample-complexity threshold $n_i \gtrsim d\, i^{2\gamma}$ for resolving the i-th spike, which matches the optimal scaling laws derived in [39, 82].

H Partial whitening and a covariance correction

Theorem 2 replaces the residual $y - \hat{f}$ by y, so a candidate feature ϕ is selected by maximizing its correlation with the label. To retain the leading effect of $\hat{f}$, one can approximate it by ϕ and instead minimize the empirical squared loss

$$\frac{1}{2}\widehat{\mathbb{E}}_n[(y-\phi)^2] = \frac{1}{2}\widehat{\mathbb{E}}_n[y^2] - \widehat{\mathbb{E}}_n[y\phi] + \frac{1}{2}\widehat{\mathbb{E}}_n[\phi^2].$$

Relative to correlation maximization, this adds the variance penalty $\widehat{\mathbb{E}}_n[\phi^2]$. For $\phi_{\boldsymbol{w}} = \langle \boldsymbol{w}, \boldsymbol{z}_{\ell-1}\rangle$, $\widehat{\mathbb{E}}_n[\phi_{\boldsymbol{w}}^2] = \boldsymbol{w}^\top \widehat{\Sigma}^{(\ell)} \boldsymbol{w}$, with $\widehat{\Sigma}^{(\ell)} := \widehat{\mathbb{E}}_n[\boldsymbol{z}_{\ell-1}(\boldsymbol{x})\boldsymbol{z}_{\ell-1}(\boldsymbol{x})^\top]$. A tractable way to incorporate this covariance is regularized whitening on its leading empirical eigenspace. Let $\widehat{\Sigma}^{(\ell)} = \sum_{j=1}^{p_{\ell-1}} s_j \boldsymbol{v}_j \boldsymbol{v}_j^\top$ with $s_1 \geq \cdots \geq s_{p_{\ell-1}}$, and define the regularized top-m whitening operator

$$P_{\lambda,m}^{(\ell)} := \sum_{j=1}^{m} (s_j + \lambda)^{-1/2} \boldsymbol{v}_j \boldsymbol{v}_j^\top, \qquad \widetilde{\boldsymbol{z}}_{\ell-1} = P_{\lambda,m}^{(\ell)} \boldsymbol{z}_{\ell-1}.$$

The empirical first-order correlation vector and second-order label-weighted operator computed from $\widetilde{\boldsymbol{z}}_{\ell-1}$ are, respectively,

$$\widetilde{\boldsymbol{u}}_\ell = P_{\lambda,m}^{(\ell)} \hat{\boldsymbol{u}}_\ell, \qquad \widetilde{C}^{(\ell)} = P_{\lambda,m}^{(\ell)} \widehat{C}^{(\ell)} P_{\lambda,m}^{(\ell)}. \tag{119}$$

For the first-order feature, mapping the correlation direction back to the original coordinates gives

$$\boldsymbol{w}_{\lambda,m}^{(1)} = \big(P_{\lambda,m}^{(\ell)}\big)^2 \hat{\boldsymbol{u}}_\ell = \sum_{j=1}^{m} \frac{\langle \boldsymbol{v}_j, \hat{\boldsymbol{u}}_\ell \rangle}{s_j + \lambda} \boldsymbol{v}_j.$$

This is the ridge coefficient restricted to the leading empirical covariance eigenspace. If $m = p_{\ell-1}$, it is exactly

$$\arg\min_{\boldsymbol{w}} \left\{ \widehat{\mathbb{E}}_n[(y - \langle \boldsymbol{w}, \boldsymbol{z}_{\ell-1}\rangle)^2] + \lambda \|\boldsymbol{w}\|_2^2 \right\} = (\widehat{\Sigma}^{(\ell)} + \lambda I)^{-1} \hat{\boldsymbol{u}}_\ell.$$

For the second-order feature, the ordinary eigendecomposition of $\widetilde{C}^{(\ell)}$ in (119) replaces the eigendecomposition of $\widehat{C}^{(\ell)}$. Under full whitening, a mapped-back direction $\boldsymbol{w}$ therefore solves the generalized eigenvalue problem

$$\widehat{C}^{(\ell)} \boldsymbol{w} = \gamma (\widehat{\Sigma}^{(\ell)} + \lambda I) \boldsymbol{w};$$

top-m whitening gives the corresponding problem restricted to $\operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$.

The covariance correction also appears directly by retaining the next term in the squared-loss dynamics. Write $t = \langle \boldsymbol{w}, \boldsymbol{z}_{\ell-1}\rangle$ and use the local quadratic approximation

$$\phi_{\boldsymbol{w}}(\boldsymbol{x}) = c_0 t + \frac{c_1}{2} t^2, \qquad \mathcal{L}_{\mathrm{loc}}(\boldsymbol{w}) = \frac{1}{2}\widehat{\mathbb{E}}_n[(y - \phi_{\boldsymbol{w}}(\boldsymbol{x}))^2].$$

Then

$$\begin{aligned} -\nabla_{\boldsymbol{w}} \mathcal{L}_{\mathrm{loc}}(\boldsymbol{w}) &= \widehat{\mathbb{E}}_n[(y - \phi_{\boldsymbol{w}})(c_0 + c_1 t)\boldsymbol{z}_{\ell-1}] \\ &= c_0 \hat{\boldsymbol{u}}_\ell + \big(c_1 \widehat{C}^{(\ell)} - c_0^2 \widehat{\Sigma}^{(\ell)}\big)\boldsymbol{w} - \frac{3c_0 c_1}{2}\widehat{\mathbb{E}}_n[t^2 \boldsymbol{z}_{\ell-1}] - \frac{c_1^2}{2}\widehat{\mathbb{E}}_n[t^3 \boldsymbol{z}_{\ell-1}] \\ &= c_0 \hat{\boldsymbol{u}}_\ell + \big(c_1 \widehat{C}^{(\ell)} - c_0^2 \widehat{\Sigma}^{(\ell)}\big)\boldsymbol{w} + O(\|\boldsymbol{w}\|_2^2). \end{aligned} \tag{120}$$

The term $-c_0^2 \widehat{\Sigma}^{(\ell)} \boldsymbol{w}$ is therefore the first correction to the correlation-based dynamics in Theorem 2. After treating the first-order forcing separately, the corresponding second-order iteration is

$$\boldsymbol{w}^{(t+1)} \leftarrow \mathrm{normalize}\Big(\boldsymbol{w}^{(t)} + \eta (c_1 \widehat{C}^{(\ell)} - c_0^2 \widehat{\Sigma}^{(\ell)}) \boldsymbol{w}^{(t)}\Big).$$

I Higher-order Decay Ansatz

I.1 A Motivation: averaging in the gradient

In the main text, Proposition 1 motivates Neural LoFi through the label-weighted moment operator, and Appendix B derives this operator from a layer-wise small-initialization expansion. That argument orders terms by their initialization scale. The present section gives a different, and complementary, justification for suppressing higher-order terms that does not rely on layer-wise scale separation, and is therefore more natural from the point of view of machine learning practices. The point is instead that the gradient is an average over data: even if a higher-order interaction is present and can matter on individual samples, its contribution to the averaged gradient is controlled by its correlation with the corresponding label-weighted degree component after lower-order correlations have been removed.

Fix a layer ℓ and write $Z = \boldsymbol{z}_{\ell-1}(X)$. For a neuron initialized at direction w_0, the exact layerwise GD update direction from (33) is, up to the global sign convention,

$$\widehat{G}^{\mathrm{GD}}_{\ell,i}(w_0) = \widehat{\mathbb{E}}_n\left[r\,\bar{a}_{\ell,i}(X)\,\sigma'(\langle w_0, Z\rangle)Z\right], \qquad r := y - \hat{f}(X),$$

where $\bar{a}_{\ell,i}(X)$ is the sample-dependent effective readout coming from the later layers. For this averaging argument, we treat this readout as constant in expectation and assume its fluctuations vanish after averaging over data. We also ignore the self-interaction from $\hat{f}(X)$ in $r = y - \hat{f}(X)$, replacing r by y at leading order. After absorbing the constant readout into the neuron-dependent scalar coefficient, the gradient direction relevant for the averaging argument is therefore

$$\widehat{G}_\ell(w_0) = \widehat{\mathbb{E}}_n\left[y\,\sigma'(\langle w_0, Z\rangle)Z\right].$$

To see which feature directions this gradient can reveal, test it against a reference direction v:

$$\langle \widehat{G}_\ell(w_0), v\rangle = \widehat{\mathbb{E}}_n\left[y\,\sigma'(\langle w_0, Z\rangle)\langle Z, v\rangle\right].$$

At population level, let P_ℓ be the law of Z, and choose an orthonormal polynomial basis $\{\psi_{\ell,k,a}\}_{k,a}$ for the polynomial span in $L^2(P_\ell)$, chosen so that for each k, $\{\psi_{\ell,k,a}\}_a$ is an orthonormal basis of the degree-k subspace obtained after removing all correlations with degrees $< k$. This basis lets us decompose the tested projection $\langle G_\ell(w_0), v\rangle$ into contributions from different degree interactions. The first two contributions are exactly the Neural LoFi terms: the degree-zero interaction gives the linear part $\langle u_\ell, v\rangle$, with $u_\ell := \mathbb{E}[yZ]$, while the degree-one interaction gives the second-order part $\langle v, C_\ell w_0\rangle$, with $C_\ell := \mathbb{E}[yZZ^\top]$; empirically this is $\langle v, \widehat{C}^{(\ell)} w_0\rangle$. The higher-degree interactions are the terms controlled by the ansatz below. If

$$a_{\ell,v}(Z) := \mathbb{E}[y\langle Z, v\rangle \mid Z], \qquad s_{\ell,w_0}(Z) := \sigma'(\langle w_0, Z\rangle),$$

then both functions can be expanded in this basis. Writing the coefficients as

$$\alpha_{\ell,k,a}(v) := \langle a_{\ell,v}, \psi_{\ell,k,a}\rangle_{L^2(P_\ell)}, \qquad \beta_{\ell,k,a}(w_0) := \langle s_{\ell,w_0}, \psi_{\ell,k,a}\rangle_{L^2(P_\ell)},$$

the population analogue G_ℓ of $\widehat{G}_\ell$ decomposes as

$$\begin{aligned}\langle G_\ell(w_0), v\rangle &= \mathbb{E}[a_{\ell,v}(Z)s_{\ell,w_0}(Z)] \\ &= \sum_{k\geq 0}\sum_a \alpha_{\ell,k,a}(v)\beta_{\ell,k,a}(w_0).\end{aligned}$$

Equivalently, writing $\Pi_{\ell,k}$ for the projection onto this degree-k subspace, defined below,

$$\langle G_\ell(w_0), v\rangle = \sum_{k\geq 0}\left\langle \Pi_{\ell,k}a_{\ell,v}, \Pi_{\ell,k}s_{\ell,w_0}\right\rangle_{L^2(P_\ell)}.$$

Thus the k-th term in the expansion is a correlation between two degree-k polynomials, each orthogonal to all lower-degree terms. For a fixed data distribution and reference direction v, the first polynomial is fixed; since w_0 is random, the second polynomial behaves like a randomly chosen degree-k direction. Heuristically, this is the same dimensional calculation as the correlation between two generic degree-k polynomials. If Z has m_ℓ degrees of freedom, the space of degree-k polynomials after removing lower-order terms has dimension of order m_ℓ^k for fixed k. A random normalized vector in an m_ℓ^k-dimensional space has squared projection of order m_ℓ^{-k} on any fixed normalized direction.

Neural Manifold Ansatz. Fix k and a layer ℓ, and write $Z = z_{\ell-1}(X)$. There exists a scale m_ℓ such that, for any fixed normalized degree-k polynomial $p_k(Z)$ orthogonal to lower-degree polynomials in $L^2(P_\ell)$, any fixed normalized degree-k univariate polynomial q_k orthogonal to lower degrees, and $w_0 \sim \mathrm{Unif}(\mathbb{S}^{p_{\ell-1}-1})$,

$$\left|\mathbb{E}_{Z\sim P_\ell}\left[p_k(Z)q_k(\langle w_0, Z\rangle)\right]\right|^2_{\mathrm{typ}} \asymp_k m_\ell^{-k}.$$

We call m_ℓ the effective number of degrees of freedom in the layer representation. Equivalently, the unsquared overlap is typically of size $\asymp_k m_\ell^{-k/2}$.

Equivalent tensor form. The above ansatz can equivalently be cast in terms of the scaling of random projections on higher-order moment tensors. For scalar labels, define the raw order-k label tensor

$$T^{\mathrm{raw}}_{\ell,k} := \mathbb{E}[y\, Z^{\otimes k}].$$

To remove lower-order correlations, associate a symmetric tensor T with the polynomial $Z \mapsto \langle T, Z^{\otimes k}\rangle$, project this polynomial onto the degree-k component orthogonal to all lower degrees, and write $T^{\mathrm{res}}_{\ell,k}$ for the corresponding residualized tensor obtained from $T^{\mathrm{raw}}_{\ell,k}$. In these coordinates, only the fixed label tensor is residualized; the random initialization enters through the raw rank-one tensor $w_0^{\otimes k}$. Thus the ansatz can also be read as

$$\left|\left\langle \frac{T^{\mathrm{res}}_{\ell,k}}{\|T^{\mathrm{res}}_{\ell,k}\|}, \frac{w_0^{\otimes k}}{\|w_0^{\otimes k}\|}\right\rangle\right|^2_{\mathrm{typ}} \asymp_k m_\ell^{-k},$$

For Gaussian or Hermite coordinates, $w_0^{\otimes k}$ is the usual degree-k random rank-one tensor.

I.2 Why this scaling is natural

The reason for m_ℓ^{-k} is dimension counting. If the current representation has m_ℓ independent coordinates, then the number of degree-k monomials after lower degrees have been removed is

$$\binom{m_\ell + k - 1}{k} \asymp_k m_\ell^k.$$

Thus the degree-k component lives in a space with about m_ℓ^k orthogonal directions. A random unit vector in an N-dimensional space has squared projection $1/N$ on a fixed unit vector; taking $N \asymp_k m_\ell^k$ gives the ansatz. The assumption is therefore not that higher-order terms are absent on individual samples. It is that, after averaging over data, a generic random degree-k component spreads its mass over the full degree-k space rather than aligning with the particular label-weighted component.

For Gaussian $Z \sim N(0, I_m)$, this count is exact at the level needed here. The orthogonal degree-k basis is the Hermite basis and has dimension $\binom{m+k-1}{k}$. If q_k were uniform on this unit sphere, then $\mathbb{E}|\langle p_k, q_k\rangle|^2 = 1/\binom{m+k-1}{k} \asymp_k m^{-k}$. The random ridge feature $\rho(\langle w_0, Z\rangle)$ has degree-k Hermite component represented by $w_0^{\otimes k}$. Rotational invariance forces the covariance of this random component to be a scalar multiple of the identity on the degree-k Hermite space, and the trace normalization gives the same m^{-k} squared-overlap scaling.

The same picture applies to an m-dimensional product distribution. If $U = (U_1, \ldots, U_m)$ has product law and $\varphi_{j,r}$ are the one-dimensional orthonormal polynomials, then the degree-k products $\prod_j \varphi_{j,\alpha_j}(U_j)$, $|\alpha| = k$, form $\binom{m+k-1}{k} \asymp_k m^k$ orthogonal directions. For data lying near an m-dimensional manifold, the same substitution $d \mapsto m$ is expected when the degree-k restrictions to the manifold remain nondegenerate, so that the order-k polynomial features span a space of dimension comparable to m^k. This is the effective-dimension content of the ansatz.

I.3 Consequence for the layerwise Neural LoFi approximation.

The population analogues of the first- and second-order Neural LoFi operators are

$$u_\ell := \mathbb{E}[yZ], \qquad C_\ell := \mathbb{E}[yZZ^\top]$$

In the layerwise GD approximation discussed above, the full neuron update is replaced by

$$G_\ell(w_0) = \mathbb{E}[y\sigma'(\langle w_0, Z\rangle)Z].$$

For a reference direction v, set

$$a_{\ell,v}(Z) := \mathbb{E}[y\langle Z, v\rangle \mid Z], \qquad s_{\ell,w_0}(Z) := \sigma'(\langle w_0, Z\rangle).$$

Then

$$\langle G_\ell(w_0), v\rangle = \langle a_{\ell,v}, s_{\ell,w_0}\rangle_{L^2(P_\ell)}.$$

Now expand s_{ℓ,w_0} in an orthonormal polynomial basis for the law P_ℓ, grouped by degree after lower-degree correlations have been removed. Writing $\Pi_{\ell,r}$ for the degree-r projection, the first two components have the form

$$\Pi_{\ell,0}s_{\ell,w_0} = c_0, \qquad \Pi_{\ell,1}s_{\ell,w_0} = c_1\langle w_0, Z\rangle,$$

where c_0, c_1 are scalar expansion coefficients. For $r \geq 2$, write

$$\Pi_{\ell,r}s_{\ell,w_0} = c_r q_{\ell,r,w_0}, \qquad \|q_{\ell,r,w_0}\|_{L^2(P_\ell)} = 1.$$

These coefficients come from the orthonormal-polynomial expansion of $\sigma'(\langle w_0, Z\rangle)$, not from a Taylor expansion of σ'. By orthogonality across residual degrees,

$$\langle G_\ell(w_0), v\rangle = c_0\langle u_\ell, v\rangle + c_1\langle C_\ell w_0, v\rangle + \sum_{r\geq 2} c_r R_{\ell,r}(w_0, v),$$

where

$$R_{\ell,r}(w_0, v) := \left\langle \Pi_{\ell,r}a_{\ell,v}, q_{\ell,r,w_0}\right\rangle_{L^2(P_\ell)}.$$

The averaging ansatz says that for each fixed v, the residual degree-r projection created by a random initialization direction satisfies

$$\boxed{|R_{\ell,r}(w_0, v)|_{\mathrm{typ}} \lesssim_r m_\ell^{-r}.} \tag{121}$$

Consequently,

$$\langle G_\ell(w_0), v\rangle = c_0\langle u_\ell, v\rangle + c_1\langle C_\ell w_0, v\rangle + O_{\mathrm{typ}}(m_\ell^{-2}),$$

with higher corrections of order m_ℓ^{-r} at degree r. This is the averaging mechanism behind the Neural LoFi approximation: the first two terms are exactly the population first- and second-order LoFi operators, while the higher-order ridge contributions are suppressed after averaging over data.

I.4 CIFAR test of the ansatz

Figure 6 tests the ansatz on real data by measuring the same random-overlap effect without explicitly constructing the full degree-k polynomial space. Each random initialization direction gives a one-dimensional ridge slice of the current representation. For random directions $w_{\ell,i}(0)$, set

$$a_{\mu,i} = \langle w_{\ell,i}(0), Z_\mu\rangle.$$

For each i, orthonormalize $1, a_{\mu,i}, a_{\mu,i}^2, \ldots, a_{\mu,i}^{k_{\max}}$ by QR on the reference samples, and let $\phi_{\ell,k,i}$ be the resulting degree-k feature. Thus $\phi_{\ell,k,i}$ is the empirical version of the degree-k component of a random ridge feature after the lower-degree components along the same slice have been removed. It satisfies

$$\widehat{\mathbb{E}}_N[\phi_{\ell,j,i}(X)\phi_{\ell,k,i}(X)] = \delta_{jk}.$$

We then plot

$$\widehat{B}_{\ell,k} = \mathrm{median}_i \left|\widehat{\mathbb{E}}_N[y\,\phi_{\ell,k,i}(X)]\right|^2$$

after subtracting the fitted $1/n$ finite-sample floor. The quantity inside the median is the squared empirical correlation between the label and a normalized residual degree-k random ridge component. Taking the median over i estimates the typical random-initialization overlap rather than a rare aligned direction. Thus $\widehat{B}_{\ell,k}$ is the empirical analogue of the squared random-overlap effect underlying (121); the observed decay with k is consistent with a degree-k polynomial space whose dimension grows like m_ℓ^k.

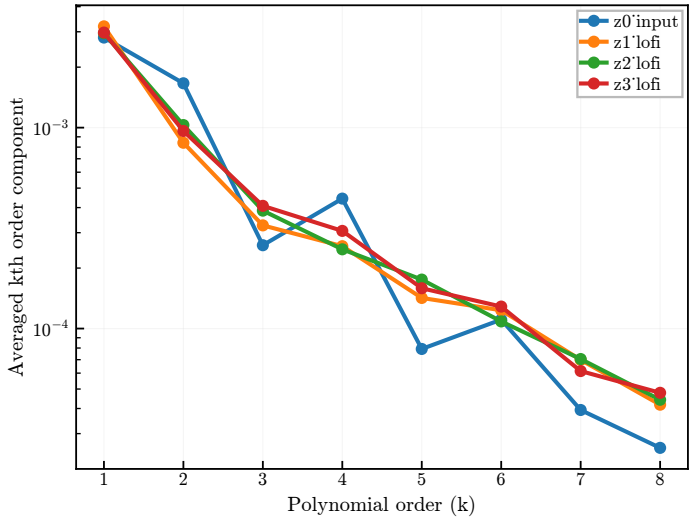


Figure 6: **Layer-wise higher-order decay on CIFAR-10.** For a three-layer fully connected Neural LoFi model trained on binary CIFAR-10 animals-versus-vehicles, we plot the averaged squared degree-k residual random-projection component as a function of polynomial order k for the input representation and the three successive LoFi layer representations.

J Comparison with AGOP/RFM

This appendix compares Neural LoFi with AGOP/RFM-type feature updates. In Recursive Feature Machines, one fits a predictor and uses its gradient with respect to the input, or to the current representation, to identify directions on which the fitted function is sensitive. Deep RFM [121] combines this gradient-based metric update with a nonlinear feature map and repeats the procedure across layers. AGOP and Recursive Feature Machines therefore provide an adaptive alternative to fixed kernels, and have been shown to recover meaningful features, including edge detectors, in neural and kernel models [51, 91, 92]. The distinction we emphasize is not that RFM is non-adaptive, but that its adaptation is mediated by a predictor that has already been fit.

More precisely, RFM first forms a kernel-ridge-regression predictor $\hat{f}$ and then updates the feature metric using its empirical average gradient outer product

$$\widehat{G}_{\mathrm{AGOP}}(\hat{f}) := \frac{1}{n}\sum_{\mu=1}^{n} \nabla\hat{f}(\boldsymbol{x}_\mu)\nabla\hat{f}(\boldsymbol{x}_\mu)^\top.$$

The resulting metric can reveal a planted feature only through the dependence already present in $\hat{f}$. Consequently, the sample size needed by an AGOP/RFM update is tied to the prediction threshold of the preceding KRR fit. This threshold is model dependent and can differ from the recovery threshold of Neural LoFi, which estimates the label-weighted feature moment directly rather than first fitting the target as a function. We first analyze a quadratic Gaussian multi-index target, for which the two thresholds differ, and then discuss the corresponding prediction threshold for Deep RFM on the hierarchical model of Appendix E.

J.1 Quadratic multi-index model

The quadratic model is the simplest case in which the target depends on a fixed low-dimensional subspace but its coefficient belongs to a polynomial space whose dimension grows as d^2. It lets us compare the full AGOP of a sub-threshold KRR predictor with the low-rank spectral truncation used by Neural LoFi.

Let

$$\boldsymbol{x} \sim \mathcal{N}(0, I_d), \qquad y = \langle A_\star, H_2(\boldsymbol{x})\rangle, \qquad A_\star = V^\star B_\star V^{\star\top},$$

where $V^\star \in \mathbb{R}^{d\times r}$ has orthonormal columns, $B_\star = B_\star^\top$ is full rank, $r = O(1)$, and $H_2(\boldsymbol{x}) = (\boldsymbol{x}\boldsymbol{x}^\top - I_d)/\sqrt{2}$. The target AGOP is

$$G(y) := \mathbb{E}[\nabla y(\boldsymbol{x})\nabla y(\boldsymbol{x})^\top] = 2A_\star^2,$$

so its range is exactly $\mathrm{span}(V^\star)$.

Since the target is purely quadratic, we isolate its relevant component by considering the degree-two kernel

$$K_2(\boldsymbol{x}, \boldsymbol{x}') = \frac{\kappa_2}{D_2}\langle H_2(\boldsymbol{x}), H_2(\boldsymbol{x}')\rangle_F, \qquad D_2 = \frac{d(d+1)}{2},$$

with fixed $\kappa_2 > 0$. This restriction removes higher-order contributions to the AGOP and makes the relation between KRR and its quadratic coefficient explicit. Every KRR predictor for this kernel has the form

$$\hat{f}_n(\boldsymbol{x}) = \langle \widehat{Q}_n, H_2(\boldsymbol{x})\rangle, \qquad \widehat{Q}_n = \widehat{Q}_n^\top.$$

It follows immediately that

$$G(\hat{f}_n) := \mathbb{E}_{\boldsymbol{x}}[\nabla \hat{f}_n(\boldsymbol{x}) \nabla \hat{f}_n(\boldsymbol{x})^\top] = 2\widehat{Q}_n^2. \tag{122}$$

Let $P_\star = V^\star V^{\star\top}$. Define the normalized trace of the AGOP projected onto the planted subspace by

$$\mathcal{E}_\star(\hat{f}_n) := \frac{\mathrm{Tr}(P_\star G(\hat{f}_n))}{\mathrm{Tr}(G(\hat{f}_n))} = \frac{\|\widehat{Q}_n P_\star\|_F^2}{\|\widehat{Q}_n\|_F^2}. \tag{123}$$

Proposition 4 (Quadratic AGOP bottleneck)**.** *Assume $\|A_\star\|_F = \Theta(1)$, fix $\delta > 0$, and consider KRR with kernel K_2, penalty $\lambda_n \geq 0$, and $n \to \infty$ satisfying $n = O(d^{2-\delta})$. Then*

$$\mathcal{E}_\star(\hat{f}_n) = o_{\mathbb{P}}(1).$$

Thus, before the d^2 prediction threshold, $\mathrm{Tr}(P_\star G(\hat{f}_n))/\,\mathrm{Tr}(G(\hat{f}_n))$ vanishes in probability.

Proof. We begin with the exact relation between the KRR predictor and its degree-two Hermite coefficient. Identify the degree-two Hermite space with the D_2-dimensional space of symmetric matrices equipped with the Frobenius inner product, and choose an orthonormal basis $\{E_a\}_{a=1}^{D_2}$ of this space. Define

$$h(\boldsymbol{x})_a := \langle E_a, H_2(\boldsymbol{x})\rangle, \qquad \Phi_{\mu a} := h(\boldsymbol{x}_\mu)_a,$$

so that $\Phi \in \mathbb{R}^{n \times D_2}$ and $K_n = (\kappa_2/D_2)\Phi\Phi^\top$ is the training kernel matrix. For the objective

$$\frac{1}{n}\sum_{\mu=1}^{n}(y_\mu - f(\boldsymbol{x}_\mu))^2 + \lambda_n \|f\|_{\mathcal{H}_{K_2}}^2,$$

write $y = (y_1, \ldots, y_n)^\top$. The dual coefficients and the fitted quadratic coefficient are exactly

$$\alpha = (K_n + n\lambda_n I_n)^{-1} y, \qquad \widehat{Q}_n = \frac{\kappa_2}{D_2}\sum_{\mu=1}^{n}\alpha_\mu H_2(\boldsymbol{x}_\mu). \tag{124}$$

Gaussian Hermite orthogonality gives $\mathbb{E}[h(\boldsymbol{x})h(\boldsymbol{x})^\top] = I_{D_2}$. The degree-two kernel-matrix concentration used in the polynomial-regime KRR results [17, 46, 122] states that, for $n = O(d^{2-\delta})$,

$$\Delta_n := \frac{1}{D_2}\Phi\Phi^\top - I_n, \qquad \|\Delta_n\|_{\mathrm{op}} = o_{\mathbb{P}}(1). \tag{125}$$

Substituting $K_n = \kappa_2(I_n + \Delta_n)$ into (124) and expanding the inverse gives

$$\alpha = \frac{1}{\kappa_2 + n\lambda_n}(I_n + E_n)y, \qquad \|E_n\|_{\mathrm{op}} = o_{\mathbb{P}}(1).$$

Now define the empirical degree-two label coefficient

$$\overline{Q}_n := \frac{1}{n}\sum_{\mu=1}^{n} y_\mu H_2(\boldsymbol{x}_\mu).$$

Equation (124) therefore yields

$$\widehat{Q}_n = a_{n,d}\overline{Q}_n + R_n, \qquad a_{n,d} := \frac{\kappa_2 n}{D_2(\kappa_2 + n\lambda_n)}, \tag{126}$$

where, using $\|\Phi\|_{\mathrm{op}}^2 = D_2(1+o_{\mathbb{P}}(1))$ and $\|y\|_2 = O_{\mathbb{P}}(\sqrt{n})$,

$$\|R_n\|_F = o_{\mathbb{P}}\left(a_{n,d}\sqrt{\frac{D_2}{n}}\right). \tag{127}$$

It remains to compare $\|\overline{Q}_n P_\star\|_F$ with $\|\overline{Q}_n\|_F$. Hermite orthogonality gives $\mathbb{E}\overline{Q}_n = A_\star$. Decompose $\boldsymbol{x} = V^\star z + V_\perp^\star u$, where $z \sim \mathcal{N}(0, I_r)$, $u \sim \mathcal{N}(0, I_{d-r})$, and z, u are independent. The label depends only on the fixed-dimensional vector z. Since $H_2(\boldsymbol{x})$ has $D_2 = \Theta(d^2)$ orthonormal coordinates, while $H_2(\boldsymbol{x})P_\star$ contains only $O(d)$ coordinates, independence across samples and Gaussian-chaos concentration give

$$\|\overline{Q}_n\|_F^2 = \Theta_{\mathbb{P}}\left(\frac{D_2}{n}\right), \qquad \|\overline{Q}_n P_\star\|_F^2 = O_{\mathbb{P}}\left(1+\frac{d}{n}\right). \tag{128}$$

For completeness, these orders follow by expanding the centered second moments:

$$\mathbb{E}\|\overline{Q}_n - A_\star\|_F^2 = \frac{1}{n}\left(\mathbb{E}[y^2\|H_2(\boldsymbol{x})\|_F^2] - \|A_\star\|_F^2\right) = \Theta\left(\frac{D_2}{n}\right),$$
$$\mathbb{E}\|(\overline{Q}_n - A_\star)P_\star\|_F^2 = \frac{1}{n}\left(\mathbb{E}[y^2\|H_2(\boldsymbol{x})P_\star\|_F^2] - \|A_\star\|_F^2\right) = O\left(\frac{d}{n}\right).$$

The first expectation contains $\Theta(d^2)$ orthonormal quadratic coordinates, whereas the projection onto the fixed-rank space $P_\star$ retains only $O(d)$ of them.

Combining (128) gives

$$\frac{\|\overline{Q}_n P_\star\|_F^2}{\|\overline{Q}_n\|_F^2} = O_{\mathbb{P}}\left(\frac{n}{D_2} + \frac{d}{D_2}\right) = o_{\mathbb{P}}(1).$$

The remainder bound (127) is negligible relative to $a_{n,d}\|\overline{Q}_n\|_F$. Hence the same ratio holds with $\widehat{Q}_n$ in place of $\overline{Q}_n$, and (123) proves the proposition. □

Proposition 4 concerns the population average in (122): after $\hat{f}_n$ is fitted, its gradient outer product is averaged over a fresh draw $\boldsymbol{x} \sim \mathcal{N}(0, I_d)$. Thus the proposition removes the finite-sample error in estimating the AGOP and shows that even this ideal average satisfies $\mathrm{Tr}(P_\star G(\hat{f}_n))/\mathrm{Tr}(G(\hat{f}_n)) = o_{\mathbb{P}}(1)$.

RFM instead computes an empirical average of these gradient outer products. If it is evaluated on an independent sample $\{\widetilde{\boldsymbol{x}}_\nu\}_{\nu=1}^m$, then

$$\widehat{G}_m^{\mathrm{ind}}(\hat{f}_n) := \frac{1}{m}\sum_{\nu=1}^m \nabla\hat{f}_n(\widetilde{\boldsymbol{x}}_\nu)\nabla\hat{f}_n(\widetilde{\boldsymbol{x}}_\nu)^\top$$

concentrates around (122) when $m \gg d$, and hence has the same asymptotic normalized trace projection. If the training inputs are reused to compute the AGOP, $\hat{f}_n$ and the empirical average are dependent; controlling that case requires an additional concentration argument and is not part of Proposition 4.

J.1.1 Neural LoFi recovery with $O(d\log d)$ samples

We now show that, in contrast, the subspace $\widehat{V}$ estimated by Neural LoFi satisfies

$$\|\sin\Theta(\widehat{V}, V^\star)\|_{\mathrm{op}} \xrightarrow{\mathbb{P}} 0 \qquad \text{whenever} \qquad \frac{n}{d\log d} \longrightarrow \infty.$$

For this target, its second-order estimator is

$$\widehat{C}_{\mathrm{LoFi}} := \frac{1}{n}\sum_{\mu=1}^n y_\mu \boldsymbol{x}_\mu \boldsymbol{x}_\mu^\top.$$

Since $\mathbb{E}[y] = 0$, $\boldsymbol{x}\boldsymbol{x}^\top = I_d + \sqrt{2}\,H_2(\boldsymbol{x})$, and the degree-two Hermite basis is orthonormal,

$$C_{\mathrm{LoFi}} := \mathbb{E}[y\boldsymbol{x}\boldsymbol{x}^\top] = \sqrt{2}\,\mathbb{E}[\langle A_\star, H_2(\boldsymbol{x})\rangle H_2(\boldsymbol{x})] = \sqrt{2}\,A_\star. \tag{129}$$

Equivalently, $\widehat{C}_{\mathrm{LoFi}} = \sqrt{2}\,\overline{Q}_n + \bar{y} I_d$, so Neural LoFi uses the same empirical quadratic coefficient $\overline{Q}_n$ that appears in the KRR expansion, but retains only its leading r eigenvectors ordered by absolute eigenvalue and discards the remaining directions.

The relevant noise scale is therefore the operator norm rather than the Frobenius norm. A Gaussian-chaos matrix concentration bound, obtained for example by truncating $|y|$ and $\|\boldsymbol{x}\|_2$ and applying matrix Bernstein, gives

$$\|\widehat{C}_{\mathrm{LoFi}} - \sqrt{2}A_\star\|_{\mathrm{op}} = O_{\mathbb{P}}\left(\sqrt{\frac{d\log d}{n}} + \frac{d\log d}{n}\right). \tag{130}$$

Assume that the nonzero eigenvalues of $B_\star$ are bounded away from zero, and let $\widehat{V}$ contain the r eigenvectors of $\widehat{C}_{\mathrm{LoFi}}$ with largest absolute eigenvalues. The Davis–Kahan theorem and (130) imply

$$\|\sin\Theta(\widehat{V}, V^\star)\|_{\mathrm{op}} \lesssim \frac{\|\widehat{C}_{\mathrm{LoFi}} - \sqrt{2}A_\star\|_{\mathrm{op}}}{\min_j |\lambda_j(B_\star)|}.$$

Consequently, for every fixed $\eta > 0$, there is a constant C_η such that $n \geq C_\eta d\log d$ implies

$$\|\sin\Theta(\widehat{V}, V^\star)\|_{\mathrm{op}} \leq \eta$$

with high probability. If $n/(d\log d) \to \infty$, this subspace error converges to zero. The gain comes from the rank-r spectral truncation: Neural LoFi isolates the operator-norm spike even though the untruncated quadratic coefficient still has Frobenius noise of order $\sqrt{D_2/n}$.

J.2 Implication for the hierarchical model

The calculation above suggests the analogous obstruction for Deep RFM on the hierarchical target of Appendix E. In the quadratic model, the KRR predictor must estimate a D_2-dimensional quadratic coefficient before $\mathrm{Tr}(P_\star G(\hat{f}_n))/\,\mathrm{Tr}(G(\hat{f}_n))$ is nonvanishing. In the hierarchical model, the first target component carrying the first-layer features $h^{(1)}$ lies in the degree-$2q$ Hermite subspace, whose dimension is $D_{2q} = \Theta(d^{2q})$. We therefore expect the KRR fit used at the first Deep RFM round, and hence its AGOP update, to remain uninformative about $h^{(1)}$ below this scale. The following proposition states the KRR part of this argument.

Proposition 5 (Sub-threshold KRR prediction in the hierarchical Hermite model)**.** *Consider the model of Appendix E with first-layer degree q and $d_1 = d^\epsilon$, $\epsilon < q$. Let $\hat{f}_n$ be a KRR or random-feature predictor in the polynomial asymptotic regime of [17, 46, 122]. Fix $\delta > 0$. If $n = O(d^{2q-\delta})$, then with high probability*

$$\mathbb{E}[\hat{f}_n(\boldsymbol{x})^2] = o_d(1), \qquad \mathbb{E}[y\hat{f}_n(\boldsymbol{x})] = o_d(1).$$

Thus below the d^{2q} scale the predictor is L^2-vanishing and remains at baseline test error.

Proof. For the target of Appendix E, Lemma 4.2 of [40] and Lemma 2 of [42] give

$$\|\Pi_{\leq 2q-1}\, y\|_{L^2(P_x)} = o_d(1).$$

The results of [17, 46, 122] imply that, at scale $n = O(d^{2q-\delta})$, the KRR/random-feature predictor is asymptotically supported only on the Hermite components of y of total degree below $2q$. Since these components vanish, the fitted predictor vanishes in $L^2(P_x)$, which gives both claims. □

The quadratic proposition shows explicitly how a sub-threshold KRR fit produces an uninformative AGOP. Proposition 5 gives the corresponding d^{2q} prediction threshold for the hierarchical target. By the same mechanism, we expect Deep RFM not to recover the first-layer latent features $h^{(1)}$ before this scale. A complete recursive theorem would also require concentration of the dependent AGOP updates across rounds.

Remark 3 (Neural LoFi recovery)**.** *As shown in Theorem 3, for quadratic first-layer features, if*

$$n \gg d^2 d_1, \qquad p_1 \gg \sqrt{d_1}\, d^2,$$

then

$$\frac{D_2}{c_2^2} A^{(1)}_{\mathrm{RF}} \widehat{C}^{(2)}_{\mathrm{RF}} A^{(1)\top}_{\mathrm{RF}} = \nu_1 A^{(2)} + o_{\mathbb{P}}(d_1^{-1/2})$$

in operator norm. Hence the projected random-feature operator has the same limiting eigenspaces as the Hermite estimator and recovers the d_1 planted directions. Neural LoFi obtains these directions by diagonalizing the label-weighted feature moment $\widehat{C}^{(2)}_{\mathrm{RF}}$, whereas Deep RFM forms its next feature metric from the gradients of a fitted KRR predictor.

More generally, for first-layer features of degree q, the relevant Hermite subspace has dimension $D_q = \Theta(d^q)$. The effective-dimension argument of Appendix G.2 gives the analogous recovery scale

$$n \gg D_q d_1 = \Theta(d^{q+\epsilon}).$$

Since $\epsilon < q$, this is asymptotically smaller than the d^{2q} KRR scale in Proposition 5.

K Additional Numerical Explorations

K.1 Neural LoFi Kernel Limit

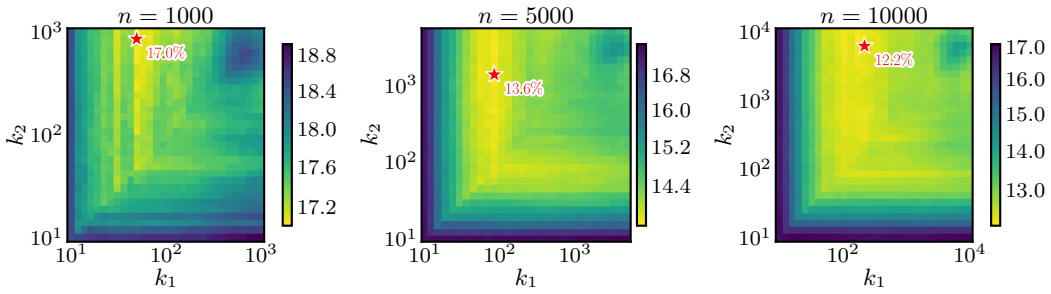


Figure 7: Test Error (%) as a function of the kept features for Kernel LoFi for different training dataset sizes n. Stars indicate the best (k_1, k_2) in each grid. The optimal retained dimension is finite and different from the usual kernel case (upper right corner), mirroring the behavior of the finite-width Neural LoFi heatmaps in Figure 3

The kernel formulation of Neural LoFi developed in Appendix F replaces the random projections $\boldsymbol{R}_\ell$ of Algorithm 1 by their kernel expectations and computes the generalized eigenvalue problem of Proposition 3 on the $n \times n$ Gram matrix. The resulting estimator is deterministic in the projection randomness and depends on $(\boldsymbol{x}_{1:n}, y_{1:n})$ only through the kernel $K_{\ell-1}$ and the labels.

Figure 7 reports the test error of kernel LoFi on binary CIFAR-10 over the per-layer retained ranks (k_1, k_2), at two training-set sizes. First, the error landscape is U-shaped in the same way as its finite-width counterpart in Figure 3: keeping too few features discards predictive signal, keeping too many includes eigendirections that are not separated from the noise bulk at this sample size, and the optimum sits at intermediate (k_1, k_2). Second, the location of this optimum shifts toward larger ranks as n grows. Both effects are those predicted by the relevance–complexity criterion of Theorem 1 and the emergence threshold of eq. (17): more samples lower the noise floor $\tau_\ell^k(n)$, allowing additional low-degree directions to cross it.

Read together with the convergence experiment of Figure 9, in which the finite-width LoFi error approaches the kernel-LoFi error from above as the projection width p grows, this establishes that the U-shape and the sample-dependent rank optimum are properties of the supervised spectral mechanism itself rather than artifacts of the random projections used in practice. This is also what distinguishes kernel LoFi from standard fixed-kernel methods such as the NNGP or NTK: the operator diagonalized at each layer is built from the labels, so the geometry in which the next layer searches for signal changes with the task. Kernel LoFi is, in this sense, the simplest deterministic object that captures the layerwise task-adaptivity of feature learning described in the main text.

The Neural LoFi kernel also gives a feature-space perspective on why overparameterization and pruning are not contradictory. Classical pruning methods show that many weights or connections can be removed after training with little loss in performance [106, 107], while the lottery-ticket hypothesis suggests that large dense networks may contain smaller trainable subnetworks [108]. In Neural LoFi, width and rank play complementary roles: large width creates a rich feature space in which task-correlated directions can be discovered, while spectral pruning retains only the directions that finite data can reliably support. Thus large networks are useful for discovery, but low-rank feature selection controls the effective dimension used for prediction.

K.2 Measuring the feature alignment of LoFi with GD

The comparison in Fig. 1 is made at the level of subspaces, not individual hidden coordinates. This is the natural invariant: hidden coordinates in both GD and Neural LoFi can be rotated, permuted, rescaled, or sign-flipped without changing the represented feature span.

Let $\mathcal{D}_{\text{test}} = \{\boldsymbol{x}_\mu\}_{\mu=1}^m$ be the held-out set on which the comparison is evaluated. For Neural LoFi, we use the filtered coordinates obtained immediately after the spectral step and before the random feature lift. Thus, for layer ℓ,

$$\left[Z_\ell^{\text{LoFi}}\right]_{\mu,:} = g_\ell^{\text{LoFi}}(\boldsymbol{x}_\mu) = \boldsymbol{z}_{\ell-1}^{\text{LoFi}}(\boldsymbol{x}_\mu) V_{\ell,k_\ell}^{\text{LoFi}}, \qquad Z_\ell^{\text{LoFi}} \in \mathbb{R}^{m \times k_\ell},$$

where $V_{\ell,k_\ell}^{\mathrm{LoFi}}$ contains the k_ℓ retained LoFi spectral directions. The lifted representation

$$\boldsymbol{z}_\ell^{\mathrm{LoFi}} = p_\ell^{-1/2}\sigma(R_\ell g_\ell^{\mathrm{LoFi}})$$

is not used in the overlap.

For the GD network at step t, write the layer-ℓ weight matrix as

$$W_\ell^{\mathrm{GD}}(t) = U_\ell(t)\Sigma_\ell(t)V_\ell(t)^\top,$$

and let $V_{\ell,r}^{\mathrm{GD}}(t)$ be the matrix of the top r right singular vectors. We run the GD network normally up to layer $\ell-1$, and then project the current representation onto these singular directions:

$$\left[Z_{\ell,r}^{\text{GD-SVD}}(t)\right]_{\mu,:} = \boldsymbol{z}_{\ell-1}^{\mathrm{GD}}(\boldsymbol{x}_\mu, t)V_{\ell,r}^{\mathrm{GD}}(t), \qquad Z_{\ell,r}^{\text{GD-SVD}}(t) \in \mathbb{R}^{m\times r}.$$

We apply neither the layer-ℓ nonlinearity nor any later GD layer or classifier. Hence both sides of the comparison are spectral coordinates exposed by the target layer before the next nonlinear expansion.

Let $H = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ be the centering matrix, and let $\mathrm{orth}(A)$ denote an orthonormal basis for the column span of A, obtained from the nonzero left singular vectors. Define

$$Q_\ell^{\mathrm{LoFi}} = \mathrm{orth}\Big(HZ_\ell^{\mathrm{LoFi}}\Big), \qquad Q_{\ell,r}^{\mathrm{GD}}(t) = \mathrm{orth}\Big(HZ_{\ell,r}^{\text{GD-SVD}}(t)\Big).$$

The directed recovery score plotted in Fig. 1 is

$$\mathrm{Recovery}_{\ell,r}(t) = \frac{\left\|\left(Q_\ell^{\mathrm{LoFi}}\right)^\top Q_{\ell,r}^{\mathrm{GD}}(t)\right\|_F^2}{\mathrm{rank}\Big(HZ_{\ell,r}^{\text{GD-SVD}}(t)\Big)}. \tag{131}$$

Equivalently, this is the average squared length of an orthonormal GD-SVD feature direction after projection onto the LoFi filtered-feature span. The score lies in $[0,1]$, equals 1 exactly when the centered GD-SVD span is contained in the centered LoFi span, and is invariant to any invertible reparameterization of the columns within either feature matrix. The companion "precision" score, useful when the LoFi span is substantially wider, uses the same numerator but divides by $\mathrm{rank}(HZ_\ell^{\mathrm{LoFi}})$.

In Fig. 1, we set $r = 8$ and report this score for each hidden layer of a four-layer fully connected network on the binary CIFAR-10 animal-vs.-vehicle task. The plotted curve is the normalized gain

$$\frac{\mathrm{Recovery}_{\ell,r}(t) - \mathrm{Recovery}_{\ell,r}(0)}{1 - \mathrm{Recovery}_{\ell,r}(0)}.$$

In the layerwise GD experiment, only one hidden layer is trained at a time; the final readout-training phase is omitted from the displayed time interval.

Perhaps more interesting are the plots in Fig. 1 where, for each hidden layer, the normalized overlap gain $(O_t - O_{\mathrm{init}})/(1 - O_{\mathrm{init}})$, where O_t is the empirical subspace overlap between the features spanned by the top-8 singular vectors of a network trained by GD and the features extracted by Neural-Lofi (more details in Appendix K.2, evaluated on $N = 10{,}000$ held-out samples and averaged over eight random initializations. The bottom-right panel trains the hidden layers sequentially, with earlier layers fixed during later phases, while the top-right panel trains all layers jointly. The layerwise protocol exhibits staircase-like growth: the layer currently being optimized develops new features aligned with the LoFi features for that layer, while already-trained layers remain approximately stable. The joint-training panel lacks these clean layerwise transitions, supporting the interpretation of Neural LoFi as a layerwise spectral approximation to the feature-discovery phase of GD. This is further supported by the aforementioned Fig. 1, that shows direct test-error comparisons between GD and Neural LoFi.

K.3 Generalization and Spectral Performance: GD vs. LoFi

In this experiment we aim to train networks for image binary classification (Animals vs. Vehicles in CIFAR-10) using both Neural LoFi and Gradient Descent (GD), and compare their generalization performance across a range of training set sizes. We investigate the generalization properties of Neural LoFi relative to GD by evaluating their test performance across varying training set sizes

$n \in [10^2, 5 \times 10^4]$. To ensure a controlled comparison, we fix the architectures for both FC and CNN models to have a comparable number of parameters and identical random projection dimensions.

For GD, we employ a compute-constrained training protocol where the total number of gradient steps is held constant across dataset sizes. We adopt an adaptive scaling law for the learning rate, $\eta \propto \sqrt{B/n}$, where B is the batch size, and utilize a cosine annealing schedule with a linear warmup. This ensures stable convergence dynamics without the need for exhaustive per-configuration hyperparameter tuning. In contrast, LoFi remains a one-pass algorithm requiring only the selection of the eigenvector bottleneck k_ℓ.

We also compare Neural LoFi with GD training protocols: with layer-wise training as well as with standard backpropagation gradient descent. Fig. 1 shows that end-to-end gradient descent (top row) and layer-wise training followed by the Neural LoFi ridge readout (bottom row) compare with Neural LoFi in the same way. Neural LoFi matches GD trained for about 500 steps (per layer). GD trained for 1000 to 2000 steps overtakes it at large n, by about one percentage point at $n = 5 \times 10^4$ (end-to-end: 10.8% vs. 12.0% for the FCN, 6.8% vs. 7.7% for the CNN). For the FCN, Neural LoFi is better than GD below $n \approx 10^3$; for the CNN, GD trained for 1000 steps or more is ahead at every n.

Figure 1 illustrates the test error as a function of the training set size. We observe that LoFi achieves generalization performance comparable to GD trained for a few hundred steps (per layer), and better than GD in the low-sample regime of the fully connected network. By reporting GD performance at fixed step budgets, we show that LoFi, despite being a one-pass spectral method, captures features comparable to those learned by GD in its early training.

While GD eventually outperforms LoFi as the training budget increases, this gap is expected; GD iteratively refines features across the full model capacity, whereas LoFi performs a sequential, layer-wise spectral projection. Neither method uses auxiliary regularization such as dropout or weight decay: Neural LoFi selects its ranks on one half of the test set, and GD is read at fixed step counts. Thus, the results in Figure 1 suggest that the spectral alignment of LoFi is not merely a theoretical property but a robust driver of generalization in deep architectures.

K.4 Filtering of Features

Figure 8 reports the test error as a function of the number of retained features k for a single hidden layer, where only the intermediate representation is subject to Neural LoFi's feature selection while the input and output layers are left untouched. This is the one-dimensional slice of the two-dimensional heatmaps of Figure 3, isolating the role of k.

Within any single panel every curve is U-shaped: retaining too few features underfits the target (high bias), while retaining too many carries over the noise of the unfiltered random features (high variance). The minimum defines an optimal rank $k^\star$, so Neural LoFi's filtering realizes a bias-variance trade-off, here controlled by an explicit number of retained directions rather than by the ridge penalty λ alone. The trend is robust across the fixed-λ rows and the cross-validated one.

Reading along a row (increasing the number of random projections p) lowers the error at the optimum and moves $k^\star$ to larger values: a wider random-feature bank spans a richer function class, so more of its directions are worth keeping and progressively less has to be cut away. The $p \to \infty$ limit (dashed) is the lower envelope of the finite-p family. Thus p should be taken as large as possible, yet filtering remains beneficial at every p.

Reading down a column (increasing the sample size n) has the same qualitative effect through a different mechanism: with more data the label-aware covariance estimates better features, so again $k^\star$ increases and the error decreases as there is less overfitting to prune.

Together these trends suggest a design point: build the widest feature bank you can afford and then recover generalization by projecting onto the task-relevant low-dimensional subspace. This is the same rationale behind network pruning [108, 123] and behind rainbow networks [87], in which each layer is a random feature map composed with a learned low-rank rotation that aligns features to the task. Neural LoFi makes that alignment explicit and label-aware, and Figure 8 quantifies how much of the width can be pruned as a function of p and n.

K.5 Predicting the sample complexity for feature recovery

The variational analysis of Section 3.1 predicts that the k-th eigenvector of the layer-ℓ signed covariance becomes recoverable once the population correlation $\rho_\ell^{(k)}$ exceeds the empirical noise

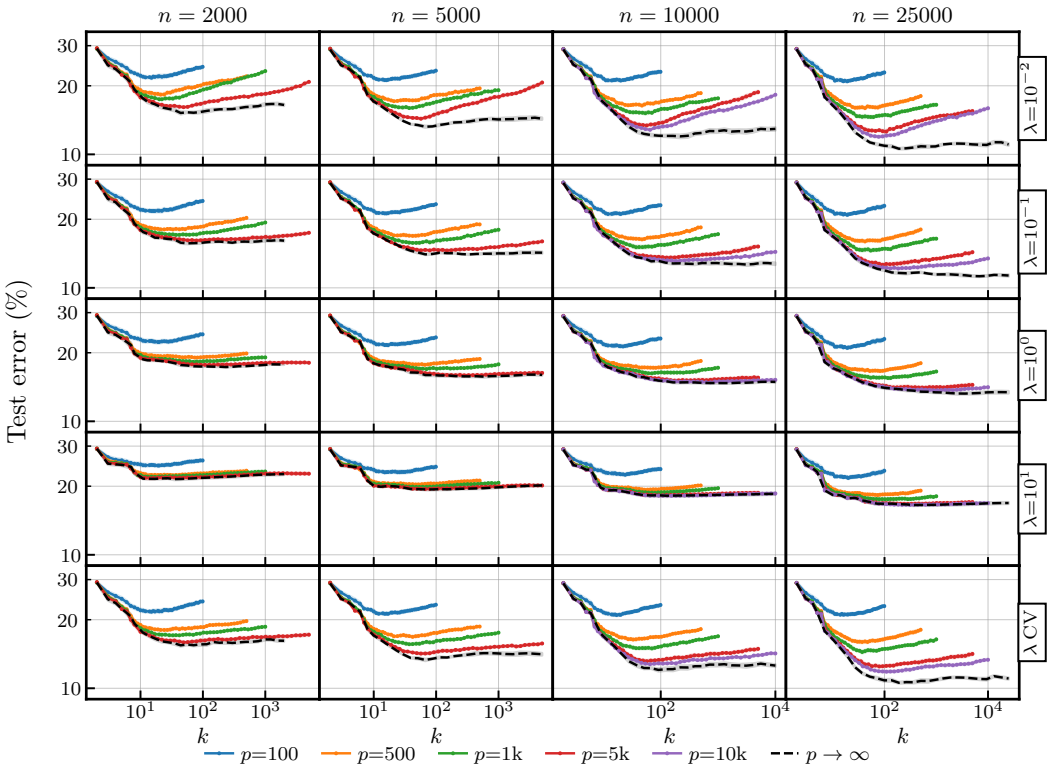


Figure 8: Test error (%) as a function of the number of retained features k at the hidden layer of a one-hidden-layer Neural LoFi with ReLU activation on CIFAR-10. Each panel fixes the ridge regularization λ (rows, from 10^{-2} to 10^{1}; the bottom row tunes λ by cross-validation) and the training-set size n (columns), and overlays curves for a range of random-projection widths p, including the infinite-width limit $p \to \infty$ (dashed). Only the intermediate representation is filtered; the input and output layers are not reduced.

floor $\tau_\ell^k(n)$, with the latter controlled by the residual effective dimension of the kernel induced by the previous layer. This is an asymptotic statement, and the quantitative prediction Eq. (117) rests on assumptions that CIFAR-10 features are not designed to satisfy. The experiment in Fig. 4 (and its convolutional analog Fig. 10) is intended as a stress test of the prediction in that regime.

We use the full binary CIFAR-10 training set ($N = 60{,}000$, animals vs. vehicles) as a population proxy: running Neural LoFi on this set yields stable reference eigenvectors $\{\hat{\boldsymbol{v}}_i^{(N)}\}_{i\geq 1}$ at the layer of interest. We then refit the same pipeline on random subsets of size $n \leq N$, holding the random projections $\boldsymbol{W}_1, \boldsymbol{W}_2$ fixed across subsets, and record the index-aligned overlap $|\langle \hat{\boldsymbol{v}}_i^{(n)}, \hat{\boldsymbol{v}}_i^{(N)} \rangle|^2$, averaged over 100 independent dataset permutations. In the convolutional figure we plot the eight overlaps after sorting them descending within each draw, which is robust to the $|\lambda|$-ordering swaps that arise when two leading eigenvalues of opposite sign have similar magnitude. The predicted thresholds n_ℓ^k are obtained from the proxy at $n = N$ by the recursive procedure of App. G and use no information from the smaller-n runs.

The predicted thresholds (dashed verticals) track the order and approximate scale of the empirical transitions across the eight leading eigenvectors, which themselves are sharp on a logarithmic scale and saturate in the order predicted by the recursive effective-dimension recipe — consistent with the BBP/EJ picture of Section 3.1. The bound is qualitative and not tight, so the prediction should be read as the correct scaling of n_ℓ^k with the residual effective dimension and the spectral gap rather than as an absolute threshold; even at this level, however, a single eigendecomposition on a reference set is enough to indicate the order of magnitude at which each eigendirection becomes extractable.

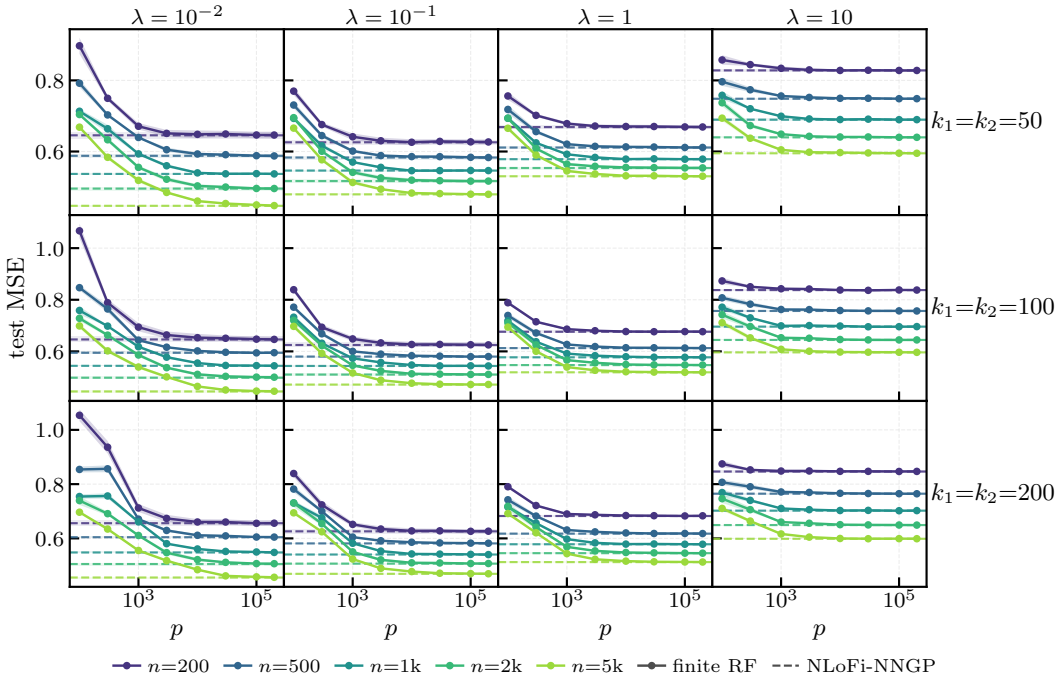


Figure 9: Convergence of finite-width NLoFi to the NLoFi-NNGP limit as the hidden width p grows. Each panel shows the test MSE versus p for four training-set sizes n. Solid curves are finite-RF NLoFi; dashed horizontal lines mark the corresponding NLoFi-NNGP limits at the same n. Rows vary the number of retained eigenfeatures per layer ($k_1 = k_2 \in \{50, 100, 200\}$); columns vary the ridge regularisation λ. Across all settings the finite-width curves converge from above to the kernel limit.

K.6 Spectrum of the signed covariance

The core of LoFi is the spectral decomposition of the signed covariance operator, thus it is crucial to understand the structure of its spectrum.

We analyze the eigenvalue spectra learned by ridge spectral networks on a binary CIFAR-10 classification task (animal vs. vehicle). The architecture comprises two convolutional layers with $p = 512$ channels, each followed by 2×2 max pooling and L_2 normalization of the features—concluding with a fully-connected layer of $p = 512$ units. All layers are trained using signed-covariance eigendecomposition with eigenvalue-based feature selection (ridge spectral training). We evaluate the model across a range of training set sizes ($n \in \{200, 500, 2000, 50000\}$) while maintaining a fixed network architecture. The grid plots in Figure 11 illustrate the full eigenvalue spectrum for each layer, utilizing a symlog x-axis to resolve both fine-grained and large-magnitude spectral components. The top five eigenvalues per layer are highlighted with red triangles, revealing how the spectral structure evolves with data scale and illustrating the relative importance of dominant versus distributed features across the network depth.

K.7 Feature Visualization

Input importance To probe which input pixels drive the activation along $v_k^{(\ell)}$, we back-propagate through the random features and accumulate the absolute Jacobian–vector product over the training set,

$$I_k^{(\ell)}(d) \;=\; \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial x_{i,d}}\left(v_k^{(\ell)\top}\phi_\ell(x_i)\right)\right| \;=\; \frac{1}{N}\sum_{i=1}^{N}\left|J_\ell(x_i)^\top v_k^{(\ell)}\right|_d, \tag{132}$$

where $J_\ell(x) = \partial\phi_\ell(x)/\partial x \in \mathbb{R}^{P\times D}$. The raw map $I_k^{(\ell)}$ is convolved with an isotropic Fourier low-pass filter $m(f) = (1 + \|f\|/f_0)^{-\alpha}$ to suppress high-frequency noise inherited from the random projections. For the input layer ($\ell = 0$) the Jacobian collapses to the identity and (132) reduces to $I_k^{(0)}(d) = \left(v_k^{(0)}\right)_d^2$.

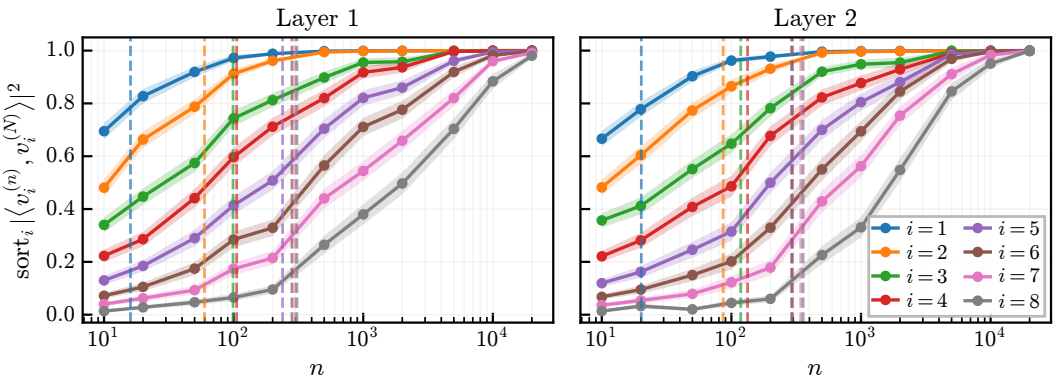


Figure 10: **Predicting when individual features emerge on CIFAR-10 with convolutional networks.** Convolutional analog of Fig. 4 in the main text. For a CNN Neural LoFi model on the CIFAR-10 animal-vs.-vehicle task (random feature matrices $\boldsymbol{W}_1, \boldsymbol{W}_2$ held fixed across subsamples), we track the squared overlap $|\langle v_i^{(n)}, v_i^{(N)}\rangle|^2$ between eigenvectors estimated from n samples and the large-sample reference eigenvectors at layer 1 (Left) and 2 (Right). At each sample size we order the eight tracked overlaps from largest to smallest within each draw, so $i = 1$ is the best-aligned axis at that n and so on; curves show mean $\pm$ SEM over 100 random subsamples. Dashed verticals are the predicted emergence thresholds n_ℓ^k from (17),(18), sorted ascending and color-matched to the curves (see Eq. (117) in the Appendix). The sharp rise of each curve near its predicted threshold shows that the effective-dimension criterion predicts the order and approximate scale at which successive task-relevant directions become learnable.

Different eigenvectors carry very different amounts of label signal. Rather than treating them uniformly, we aggregate the K_ℓ retained per-k importance maps with weights given by the absolute signed-covariance eigenvalues,

$$\bar{I}^{(\ell)}(d) \;=\; \frac{\sum_{k=0}^{K_\ell-1} |\lambda_k^{(\ell)}|\, I_k^{(\ell)}(d)}{\sum_{k=0}^{K_\ell-1} |\lambda_k^{(\ell)}|}. \tag{133}$$

This emphasises directions that most strongly co-vary with the target while preserving the spatial information carried by the lower-ranked, but still label-aligned, eigenvectors.

In Figure 12 each panel shows $\bar{I}^{(\ell)}$ reshaped to the 64×64 image grid and averaged over the three colour channels. We min–max rescale each panel independently to $[0, 1]$. The shared colorbar reports this per-panel normalised intensity. The vertical label on the leftmost panel names the binary CelebA attribute used as y.

Filter and activations for CNNs In Section 4 we have already discussed the filters and activations learned by LoFi for CNNs. Here we provide additional visualizations of these filters and activations, for different layers and training set sizes. In Figure 13 we show the first-layer filters learned by LoFi, as well as the activations of the top-ranked and mid-ranked eigenvectors at each convolutional layer. Despite being larger than the filters learned on CIFAR-10 (Figure 4), the filters learned on CelebA are still interpretable as edge detectors, with a variety of orientations and frequencies. It is interesting to compare the features learned by different eigenvectors: the top eigenvector tends to be a brightness filter and converges to the same representation when going deeper. The mid-ranked eigenvector, instead, learns more complex features even at the first layer, obtaining a self-evident representation of the sample after the last convolutional layer.

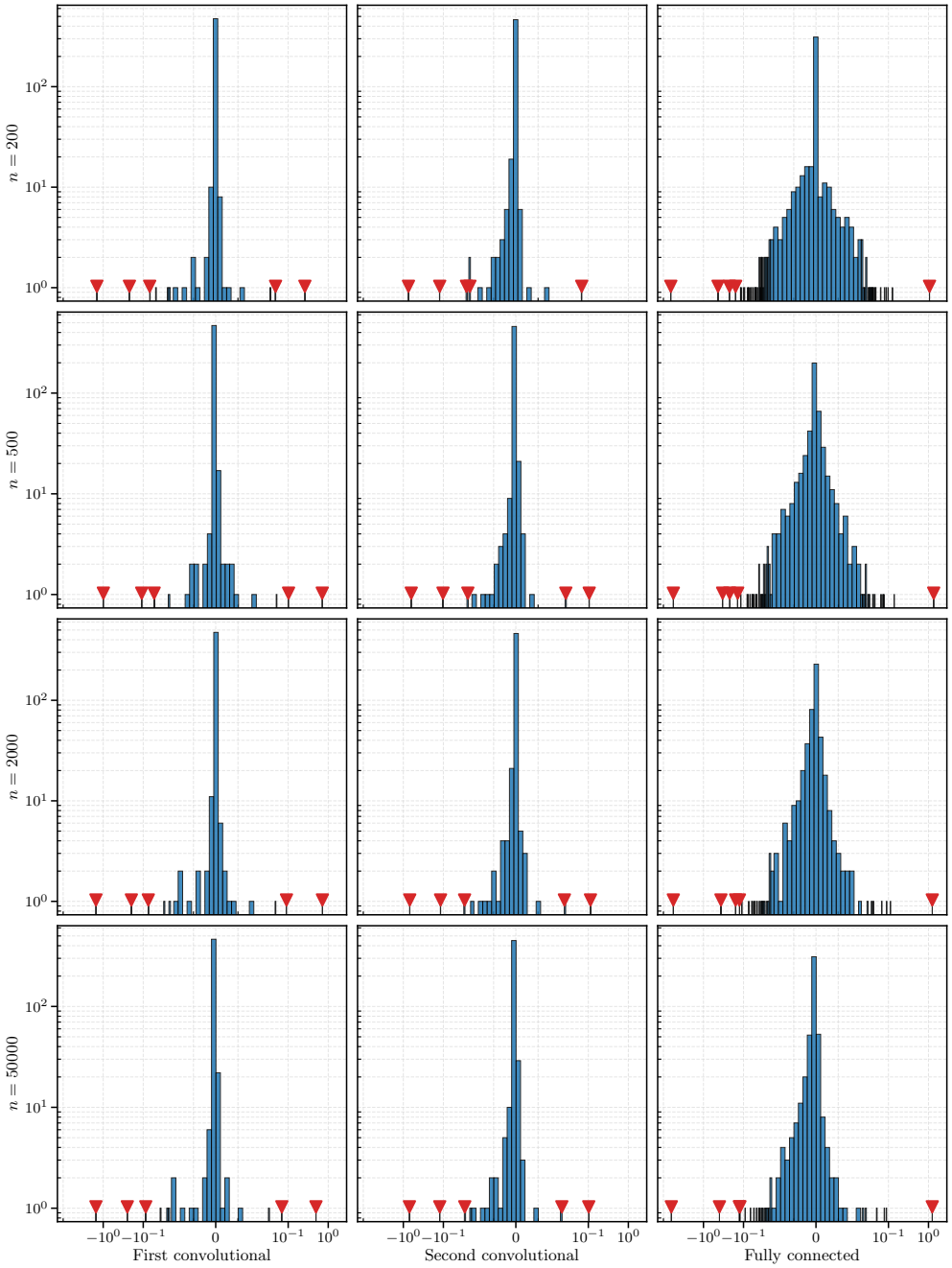


Figure 11: **Spectral Distribution:** Histograms of eigenvalues across network layers (columns) and dataset sizes (rows). Red markers indicate the five most dominant eigenvalues. The symlog scale reveals the emergence of spectral structure and the separation of lead features from the bulk distribution as n grows. The random feature dimension in this experiment is $p = 512$ for all layers.

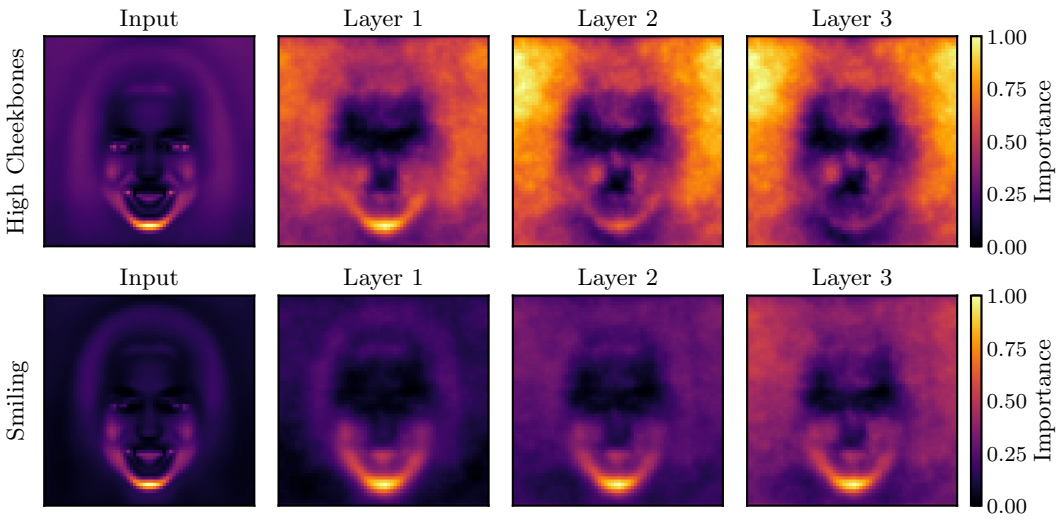


Figure 12: Layer-wise feature importance $\bar{I}^{(\ell)}$ (eq. (133)) on CelebA [124] for two binary attributes. *(Top, High Cheekbones)* Importance concentrates progressively on the cheekbone region across layers, while the chin area, salient at the input, is progressively suppressed in deeper layers. *(Bottom, Smiling)* Importance remains focused on the mouth and jaw region throughout all layers, reflecting that the discriminative signal for this attribute is preserved and not discarded by Neural LoFi's feature selection. This contrast illustrates that the retained features are task-dependent and geometrically meaningful.

L Code Implementation

All experiments are carried out with a self-contained Python package built on PyTorch [125], Numpy [126], Scipy [127] and scikit-learn [128]. The source code is available in the attached zip file.

For each layer $\ell = 1, \ldots, L$, the trainer (i) estimates the RMS norm $c_\ell = \left(\frac{1}{n}\sum_i \|\mathbf{h}_{\ell-1}^{(i)}\|^2\right)^{1/2}$ to keep pre-activations $O(1)$; (ii) draws a frozen random map $\mathbf{W}_\ell \in \mathbb{R}^{d_{\ell-1}\times p_\ell}$, $W_{\ell,ij} \sim \mathcal{N}(0,1)$, and forms $\mathbf{Z}_\ell = p_\ell^{-1/2}\,\sigma_\ell\big(\mathbf{H}_{\ell-1}\mathbf{W}_\ell/c_\ell\big)$; (iii) computes the top-k_ℓ eigenvectors $\mathbf{V}_\ell$ of the signed covariance $\mathbf{C}_\ell = \frac{1}{n}\mathbf{Z}_\ell^\top \operatorname{diag}(\mathbf{y})\,\mathbf{Z}_\ell$ ranked by $|\lambda|$; and (iv) sets $\mathbf{H}_\ell = \mathbf{Z}_\ell\mathbf{V}_\ell$.

The eigendecomposition of $\mathbf{C}_\ell$ is the dominant cost, and three paths are dispatched automatically based on (n, p_ℓ, k_ℓ): for $p_\ell \le 25{,}000$ on GPU, $\mathbf{C}_\ell$ is formed explicitly and diagonalised with `torch.linalg.eigh`; for moderate p_ℓ on CPU, the full LAPACK eigensolver is used when $k_\ell \ge p_\ell/4$ and Lanczos (`scipy.sparse.linalg.eigsh`) otherwise; for $p_\ell > 50{,}000$ and $k_\ell \le p_\ell/2$, a `LinearOperator` is passed to `eigsh` so that $\mathbf{C}_\ell$ is never materialised, requiring only $O(np_\ell)$ memory and two passes per Lanczos iteration. A deterministic starting vector $\mathbf{v}_0 = \mathbf{1}/\sqrt{p_\ell}$ ensures reproducibility, and when $\mathbf{Z}_\ell$ does not fit in GPU memory, features are streamed in batches and $\mathbf{C}_\ell$ is accumulated via outer products before calling `eigsh`.

The final representation $\mathbf{H}_L$ is passed to `sklearn.linear_model.RidgeCV`; classification accuracy is obtained by thresholding at zero (targets ± 1).

Each individual run is lightweight and requires no specialized compute infrastructure: typical runs of the algorithm complete in well under 30 minutes on a single workstation (Dell Precision 7960 Tower), with the largest experiments (e.g. the CelebA CNN) taking longer. The plots presented in this paper are intended to illustrate the theory rather than to claim state-of-the-art performance or implementation.

L.1 Figure Reproducibility

We list below the hyperparameters used for each figure (panel by panel). Unless stated otherwise, random-feature weights are Gaussian, biases are zero, the final readout is ridge regression, and curves are averaged over multiple seeds.

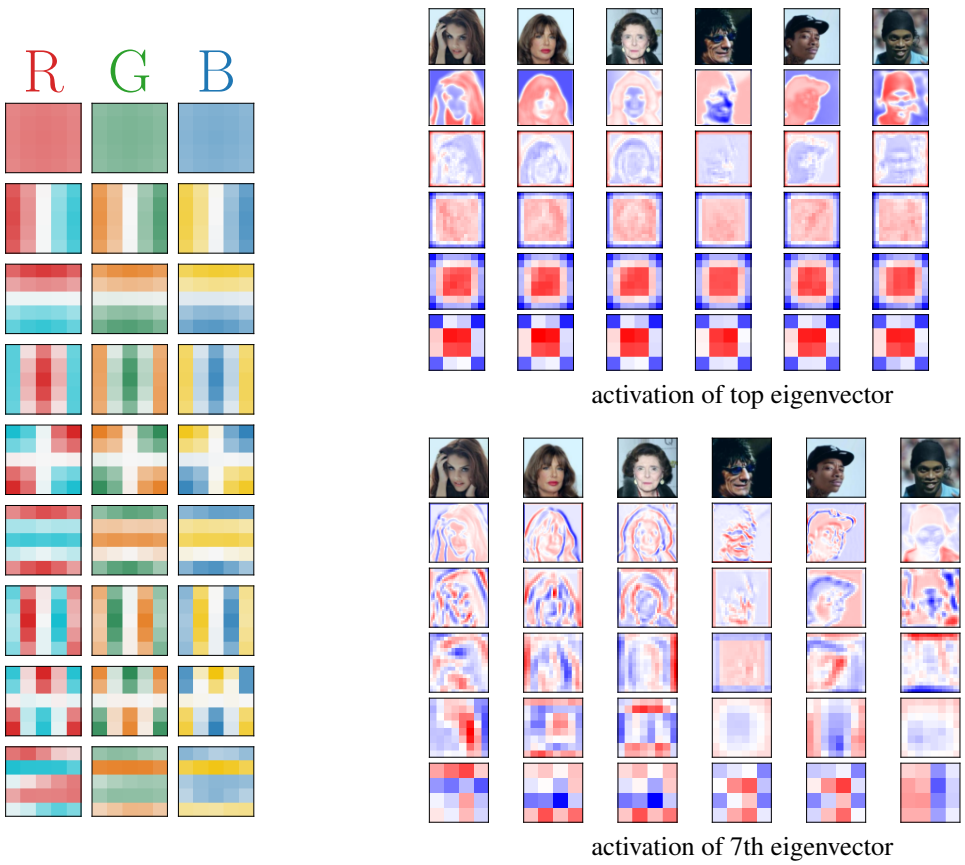


Figure 13: **Filters and activations for CNNs.** We train a 6 convolutional + 1 fully connected layer neural network on CelebA [124] for binary classification of the "Gender" attribute, using Neural LoFi with signed-covariance eigendecomposition and eigenvalue-based feature selection. *(Left)* The 5×5 first-layer filters learned by Neural LoFi, visualized as RGB images. *(Right)* The activations of the top-ranked (1st) and mid-ranked (7th) eigenvectors at each convolutional layer (lower rows are deeper), for 6 images of the test set.

Figure 3, left. Binary CIFAR-10 (animals vs. vehicles), $L = 3$ ReLU LoFi layers. Test error vs. n for ridge, 3-layer random features, and LoFi with $p \in \{5000, 10000\}$; LoFi ranks (k_1, k_2) optimally found after a logarithmical grid search in $[2, p-1]^2$. Ridge λ optimally tuned with `RidgeCV` from [128] in $[10^{-6}, 10^{6}]$ with 500 points; 10 seeds. The LoFi points are evaluated on a held-out test set: the ranks (k_1, k_2) are selected by the test error on one half of the CIFAR-10 test set (5000 images) and the plotted error is measured on the disjoint other half (5000 images); the ridge and random-feature baselines involve no rank selection and are evaluated on the first half.

Figure 3, right. Same setup. Test error grid over (k_1, k_2) at fixed $p = 5000$, $k_3 = p$, for $n \in \{5000, 10000, 20000, 50000\}$. Stars mark the best (k_1, k_2) in each grid. 10 seeds.

Figure 1, right. Four-layer FC ReLU net on binary CIFAR-10; width $p = 1000$, LoFi ranks $k = 20$, $n = 50000$. The plotted score is Eq. (131) using the filtered LoFi features and the top $r = 8$ GD-SVD features at each layer, converted to the normalized gain $(O_t - O_{\text{init}})/(1 - O_{\text{init}})$. GD is trained layerwise for 200 steps per hidden layer, and the final readout-training phase is omitted from the displayed 800-step window.

Figure 4, right (filters). Binary CIFAR-10 animal-vehicle. Convolutional LoFi: 4 conv layers (3×3, 2×2 pool, L_2 norm) + 1 FC layer; channels $p = 4096$, ranks 32,64,128,256, ReLU; $n = 50000$. Top first-layer filters from the three RGB channels.

Figure 4, right (activations). Same architecture. Activation maps of the 6th LoFi feature on test images at successive depths.

Figure 2, left, center. Hierarchical teacher of App. E with $k = 2$, $\epsilon = \frac{1}{2}$ and $g^\star = \tanh$; $d \in \{80, 100, 120, 140\}$. RF widths $(p_1, p_2) \in \{(20000, 512), (30000, 768), (40000, 1024), (50000, 1280)\}$, spherical weights, and activations as in App. E.2. The top eigenvectors are computed with a power iteration using at most 15 iterations and oversampling parameter 10. Readout: polynomial kernel of maximal degree 5, ridge regularization 10^{-6}, kernel regularization 10^{-4}. Left: test MSE vs. $\alpha = \log(n)/\log(d)$. Center: overlap between $\widehat{h}^{(1)}$ and $h^{(1)}$ vs. α. 10 seeds.

Figure 2, right. Same hierarchical teacher with $q = 2$, $d = 100$, $\epsilon = \frac{1}{2}$, $\alpha = 3$, and $g^\star = \tanh$, using a first-layer random-feature width $p_1 = 10000$ and the activations of App. E.2. The displayed spectrum is computed with a randomized eigensolver. One seed.

Figure 5 Same hierarchical teacher as in Figure 2, right, with $q = 2$, $d = 100$, $\epsilon = \frac{1}{2}$, and $g^\star = \tanh$, using a first-layer random-feature width $p_1 = 10000$ and the activations of App. E.2. The figure shows the spectrum of the first-layer spectral operator for several values of $\alpha \in \{1.5, 2.0, 2.5, 3.0, 3.5\}$, each computed with a randomized eigensolver. One seed per value of α.

Figure 7 Kernel LoFi (App. F) on binary CIFAR-10 (animals vs. vehicles), $L = 3$ layers, ReLU activation. Test error grid over $(k_1, k_2) \in [10, n-1]^2$ for $n \in \{1000, 5000\}$; stars mark the best (k_1, k_2). 10 seeds. The final optimization for the ridge parameter of Kernel Ridge Regression is done for $\lambda \in [10^{-5}, 1]$ with 20 log-spaced points.

Figure 1, left. Binary CIFAR-10 animal-vehicle. FCN: three ReLU layers of width 5000, $n \in [10^2, 5 \times 10^4]$; CNN: conv 3×3 (4096 channels), conv 3×3 (4096 channels) + 4×4 max-pool, FC 5000, $n \in [5 \times 10^2, 5 \times 10^4]$. GD networks have the Neural LoFi widths, with the first layer frozen at the Neural LoFi random features of the same seed; plain full-batch GD (no momentum, no weight decay) on the squared loss with ± 1 labels, $\eta = 0.1$ (FCN) and $\eta = 0.005$ (CNN) for every n, halved for a layer whose loss diverges. Top: end-to-end GD for 3000 steps, error read at exactly 200, 500, 1000, 2000 steps. Bottom: layerwise GD with $T \in \{50, 100, 200, 500, 750, 1000, 2000\}$ steps per layer, each layer trained through the untrained layers above it, then a leave-one-out ridge readout on the last layer (as in Neural LoFi, λ in $[10^{-6}, 10^6]$). Neural LoFi: ranks selected on one half of the test set (FCN) or on the full test set (CNN). GD errors are reported on the held-out half of the test set. FCN: 10 seeds; CNN: 2 seeds.

Figure 8. Binary CIFAR-10, single LoFi hidden layer (ReLU), only the hidden representation reduced to rank k; input and output layers unreduced. Specifically the network is

$$\phi(x) = \sigma\Big(\tfrac{1}{\sqrt{d}}\, W^{(1)} x\Big) \in \mathbb{R}^p, \qquad W^{(1)} \in \mathbb{R}^{p \times d}, \tag{134}$$

$$g(x) = \sigma\Big(\tfrac{1}{\sqrt{k}}\, W^{(2)} V\, \phi(x)\Big) \in \mathbb{R}^p, \qquad W^{(2)} \in \mathbb{R}^{p \times k},\ V \in \mathbb{R}^{k \times p}, \tag{135}$$

$$f(x) = w^\top g(x) + b, \qquad w \in \mathbb{R}^p,\ b \in \mathbb{R}, \tag{136}$$

where σ is the ReLU. The two projection matrices $W^{(1)}, W^{(2)}$ are drawn i.i.d. Gaussian and *frozen*; they are never trained. Panels are a grid: rows fix the ridge penalty $\lambda \in \{10^{-2}, 10^{-1}, 10^0, 10^1\}$ (bottom row: λ tuned per point with `RidgeCV` from [128] in $[10^{-6}, 10^6]$ over 100 points), columns fix the training-set size $n \in \{2000, 5000, 10000, 25000\}$. Each panel overlays $p \in \{100, 500, 1000, 5000, 10000\}$ and the infinite-width limit $p \to \infty$ (two-layer ReLU NNGP kernel, dashed; same per-row λ, double-centered readout). Retained rank k spans $2, \ldots, p-1$ (finite, ~ 50 points) and $2, \ldots, n-1$ (kernel, 30 points). Shading: ± 1 SEM over seeds (10 finite, 5 kernel).

Figure 9. Binary CIFAR-10 (animals vs. vehicles), 3-layer ReLU LoFi with last layer before Ridge not reduced. Test MSE vs. width p for $n \in \{200, 500, 1000, 2000, 5000\}$. All shaded areas are average over 10 seeds.

Figure 4, left. Binary CIFAR-10 (animals vs. vehicles); a 2-layer ReLU LoFi with a population proxy obtained from the full dataset ($N = 60{,}000$). For each sample size n, the LoFi pipeline is fit on a fresh permutation of n training points (with $\boldsymbol{W}_1, \boldsymbol{W}_2$ held fixed across permutations); we plot the index-aligned overlap $|\langle \hat{\boldsymbol{v}}_i^{(n)}, \hat{\boldsymbol{v}}_i^{(N)} \rangle|^2$ for the top $i = 1, \ldots, 6$ eigenvectors of the layer-ℓ signed covariance. Mean $\pm$ SEM over 100 dataset permutations. Dashed verticals: predicted thresholds n_ℓ^k from eq. (117).

Figure 6. Binary CIFAR-10 (animals vs. vehicles); a 3-layer fully connected ReLU LoFi model. For the input representation and the three successive LoFi layer representations, we use the layer features directly, sample random directions, build an empirical orthonormal polynomial basis in each one-dimensional projection, and plot the median squared label projection of the degree-k component after subtracting the fitted $1/n$ finite-sample noise floor.

Figure 10 Binary CIFAR-10 (animals vs. vehicles); a 2-layer convolutional ReLU LoFi (random convolutions with $P = 5{,}000$ channels, 5×5 kernels, padding 2, stride 1; layer-1 eigenreduction $K_1 = 100$) with a population proxy obtained from $N = 30{,}000$ samples. For each sample size n, the LoFi pipeline is fit on a fresh permutation of n training points (with $\boldsymbol{W}_1, \boldsymbol{W}_2$ held fixed across permutations); we plot the per-layer diagonal overlaps $|\langle \hat{\boldsymbol{v}}_i^{(n)}, \hat{\boldsymbol{v}}_i^{(N)} \rangle|^2$ for $i = 1, \ldots, 8$, sorted descending within each draw so that $i = 1$ is best-aligned axis at that n. Mean $\pm$ SEM over 100 dataset permutations. Dashed verticals: predicted thresholds n_ℓ^k from eq. (117).

Figure 11. Binary CIFAR-10. Two conv layers (kernel 3×3, $p = 512$, 2×2 pool, L_2 norm) + FC layer ($p = 512$); $n \in \{200, 500, 2000, 50000\}$, ranks $k_\ell = 32, 64, 128$. Symlog x-axis; top-5 eigenvalues marked.

Figure 12, top. CelebA at 64×64, attribute *High Cheekbones* from CelebA. $L = 3$ FC LoFi layers (ReLU), $P = 50000$, $(k_1, k_2, k_3) = (100, 50, 20)$, $n = 200{,}000$, seed 0. Importance via Eq. (133); Fourier low-pass parameters $f_0 = 0.15$, $\alpha = 3.0$.

Figure 12, bottom. Same setup, attribute *Smiling* from CelebA.

Figure 13, left. CelebA *Gender*. Six conv LoFi layers (5×5, 2×2 pool, L_2 norm) + FC layer; channels $p = 512$, ranks 16, 32,64,128,256,512 , ReLU; $n = 500000$. First-layer 5×5 filters as RGB images.

Figure 13, right. Same architecture. Activations of the 1st-ranked (top) and 7th-ranked (bottom) eigenvectors at each conv layer (shallow$\rightarrow$deep), on six test images.

NeurIPS Paper Checklist

(i) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes],

Justification: The abstract and introduction state the main contributions (introducing Neural LoFi as a spectral surrogate for hierarchical feature learning, deriving an RKHS variational characterization, and validating the mechanism on FC and convolutional networks), all of which are developed in Sections 2–4.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(ii) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Limitations are explicitly discussed in the final paragraph of Section 4.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important

role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(iii) **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theoretical results are stated with assumptions and proofs in the main text and Appendices B, C, D, E and F.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(iv) **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental setups (datasets, architectures, training protocols, learning-rate schedule) are described in detail in Appendix L.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(v) **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: See attached file.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(vi) **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Training protocol, architectures, projection dimensions, and feature ranks are reported in Section 4 and Appendix K.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

(vii) **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Figures with shaded regions report standard deviations over random seeds covering initialization, random feature draws and random dataset sampling.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(viii) **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The relevant information are reported in Appendix L.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(ix) **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work is foundational theoretical research on supervised learning with standard public benchmarks and conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(x) **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper is foundational theoretical work on the mechanisms of feature learning and has no direct path to specific positive or negative societal applications beyond improving general scientific understanding of deep learning.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(xi) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases no pre-trained generative models, scraped datasets, or other high-risk artifacts.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(xii) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: CIFAR-10 and CelebA are cited and used under their original research licenses as well as standard libraries are used as released by their authors.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(xiii) **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code is provided with documentations and docstrings.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(xiv) **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(xv) **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human-subjects research was conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

(xvi) **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the methods (used at most for writing/editing).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.