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

author names withheld

Under Review for the 4th Workshop on High-dimensional Learning Dynamics, 2026

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 feature learning beyond the lazy regime. It predicts how representations are selected layer by layer, and gives a concrete mechanism by which depth progressively constructs new features from old ones. We complement the theory with mechanistic experiments on fully connected and convolutional architectures, showing that Neural LoFi improves over random-feature baselines, recovers meaningful structured filters, and predicts representations aligned with early gradient-descent feature discovery.

1. Introduction

A striking feature of modern deep learning [49] is that neural networks learn complex high-dimensional functions from data with extraordinary effectiveness [82]. Yet this empirical success still outpaces theory. 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 increasingly complex patterns, from local edge-like detectors to semantic motifs [98]. Transfer-learning experiments indicate that earlier layers contain more general features, whereas deeper layers specialize to the task [97]. The *platonic representation* viewpoint emphasizes that models often learn aligned representations across architectures and training procedures [44]. These observations point to a central challenge: *can we understand which features are discovered during training, in which order, and at what sample complexity?*

Several theoretical perspectives have clarified parts of this picture. In the lazy regime [23], equivalently in the Neural Tangent Kernel (NTK) limit [45, 50], training becomes 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 parameter evolution and representation learning [22, 56, 75, 83], while tensor-program and dynamical mean-field theory approaches provide a framework for infinite-width limits, including feature-learning regimes under suitable parametrizations [16, 40, 66, 96]. In parallel, stylized high-dimensional models, such as

single-index and multi-index settings, analyze when neural networks recover latent structure from data [13, 24, 38, 88], and recent work clarifies the computational role of depth in synthetic toy models [18, 28, 37, 73]. 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: *Neural Low-Degree Filtering* (Neural LoFi). Neural LoFi is a tractable iterative spectral surrogate for feature learning: Given a current representation, each layer forms a label-weighted moment operator, selects its leading eigendirections, and lifts the projected features through nonlinear random features. This procedure is motivated by a small-initialization limit of gradient-based training, where the feature-learning component of the layerwise dynamics is governed by a Hessian-like, label-weighted second-order operator.

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 with 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 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 helps 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 high-degree input structure into low-degree structure in an intermediate representation. This leads to a principle of *low-degree compositionality*: deep learning is efficient when each stage of the target is compositional and visible through low-degree correlations in the current representation. The same viewpoint 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 an adaptive multilayer kernel construction driven by supervised low-degree feature selection.

2. Neural LoFi: core mechanism and message

Neural LoFi is a deliberately simplified model of feature learning. Instead of following the full coupled dynamics of backpropagation, it isolates a layerwise *filter–lift* mechanism. Suppose that after layer $\ell - 1$ the network represents an input by $\boldsymbol{z}_{\ell-1}(\boldsymbol{x})$. Neural LoFi forms the supervised first- and second-order moments

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

The vector $\hat{\boldsymbol{u}}_\ell$ captures the linear correlation already visible in the current representation. The eigendirections of $\widehat{\boldsymbol{C}}^{(\ell)}$ capture the first genuinely nonlinear feature-learning signal: directions whose squared activation is correlated with the label. This operator is motivated by the small-initialization, early-time expansion of layerwise gradient descent, where the feature-learning part of the update acts as a supervised power iteration of $\widehat{\boldsymbol{C}}^{(\ell)}$; the full Neural LoFi discussion is in Appendix A.1, and the formal gradient-descent derivation is in Appendix C.

The algorithmic surrogate is then simple. Let $\widehat{\boldsymbol{V}}_\ell$ contain the normalized linear direction and the top k_ℓ eigendirections of $\widehat{\boldsymbol{C}}^{(\ell)}$ ranked by eigenvalue magnitude. Neural LoFi filters the current representation to

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

and then lifts these task-relevant coordinates through a fresh nonlinear random-feature map,

$$\boldsymbol{z}_\ell(\boldsymbol{x}) = p_\ell^{-1/2}\sigma(\boldsymbol{R}_\ell \boldsymbol{g}_\ell(\boldsymbol{x})).$$

Iterating this filter–lift loop produces a sequence of task-adaptive representations, or equivalently a sequence of supervised kernels, rather than a single fixed feature geometry chosen before seeing the labels. The full layerwise procedure appears in Appendix A.1 as Algorithm 1.

The key interpretation is a relevance–complexity trade-off. In the RKHS $\mathcal{H}_{\ell-1}$ induced by the current representation, the second-order LoFi features solve, informally,

$$\max_{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1} \left|\widehat{\mathbb{E}}_n\left[y\,\varphi(\boldsymbol{x})^2\right]\right|.$$

Thus a feature is selected when it is both simple in the current geometry and predictive through a low-degree supervised statistic; the precise RKHS statement appears in Appendix A.1 as Theorem 2, with kernel details in Appendix G. This also gives a finite-sample criterion for feature emergence. After already selected directions are removed, a new feature becomes visible when its population correlation $\rho_\ell^{(k)}$ exceeds the empirical noise floor $\tau_\ell^k(n)$,

$$\rho_\ell^{(k)} \gg \tau_\ell^k(n),$$

with the latter controlled by the residual effective dimension of the current kernel. Below this threshold, empirical eigendirections are dominated by noise; above it, a task-relevant direction separates and can be learned. Appendix A.2.1 develops this emergence criterion, and Appendix H gives the effective-dimension bound controlling $\tau_\ell^k(n)$.

The resulting message is that depth is useful when the target is *low-degree compositional*: what is hard or high-degree in the input can become a sequence of statistically detectable low-degree problems in learned intermediate representations. Neural LoFi makes this principle explicit: each layer keeps the few simple directions that are currently label-correlated, discards much irrelevant variation, and lifts the retained coordinates so that the next layer can expose new structure. Appendix A.2.2 gives the longer discussion of this principle.

3. Numerical illustrations: from a solvable model to real data

Solvable models — We first consider a toy model in which the full Neural LoFi mechanism can be inspected mathematically. Following the hierarchical polynomial constructions of [36, 63, 86, 91], 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

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

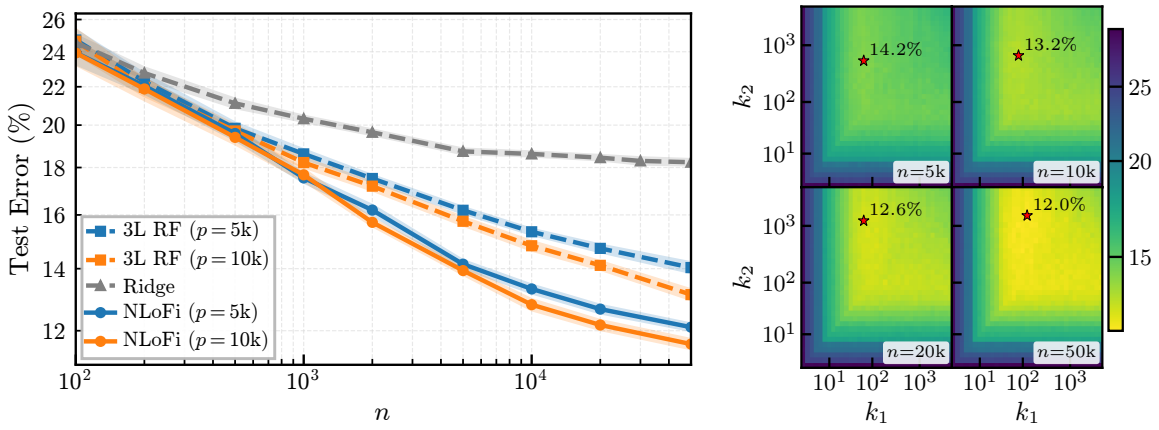

Figure 1: *Fully connected Neural LoFi on CIFAR-10 animal-vs.-vehicle. Left:* Test error vs training samples for ridge, three-layer random features, and Neural LoFi ($p = 5\text{k}, 10\text{k}$). *Right:* Test error over retained features (k_1, k_2) for several training-set sizes; stars mark the best grid point.

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

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

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 separates global input-output complexity from local low-degree learnability, making the model a clean testbed for low-degree compositionality.

In this setting, the label-weighted Neural LoFi operator has a transparent signal-plus-noise structure. At the first layer, its population part is aligned with the span of the planted directions $\{\boldsymbol{A}_i^{(1)}\}_{i=1}^{d_1}$, while finite samples create a spectral noise bulk. As n increases, the informative directions separate from this bulk through BBPEA-type transitions; the associated eigenvectors recover the hidden features. Once this first representation is recovered, the second stage $h^{(2)}$ becomes visible through another low-degree spectral problem. The final target is therefore learned by progressively building the correct intermediate representation, rather than by fitting the full high-degree map $\boldsymbol{x} \mapsto y$ in one shot. Appendix F provides the corresponding mathematical theorems and synthetic experiments, demonstrating feature emergence at the predicted sample scales (Eq.20), spectral outlier formation, alignment with the planted features, and recovery of the final compositional target.

Fully connected networks (FCN) — We now turn to real data as mechanistic illustrations, not as a competitive-training claim. Fig. 1 (FCN on CIFAR-10 [46]) illustrates two qualitative predictions: i) the label-weighted LoFi operator extracts useful directions beyond fixed random features, improving over ridge and random-feature baselines. ii) feature selection is non-trivial: keeping more features is not always best. 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. Fig. 2 compares Neural LoFi with backpropagation: the overlap grows early in training, indicating that LoFi approximates the initial feature-discovery phase of SGD (see Fig. 7 for direct test-error comparisons), then plateaus or decreases as backpropagation moves beyond the one-step LoFi approximation; additional spectra, Neural-LoFi kernel experiments, and feature-emergence predictions are reported in Apps. I.3 and I.5.

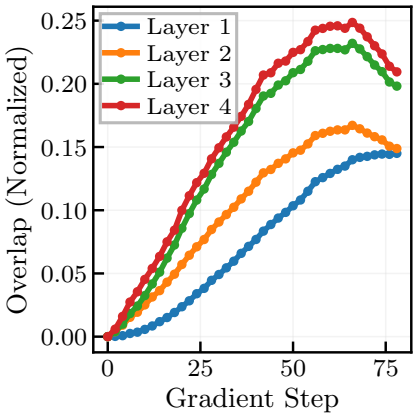

Figure 2: Normalized overlap between SGD and Neural LoFi features during training for a four-layer FCN on CIFAR-10; see Appendix I.2.

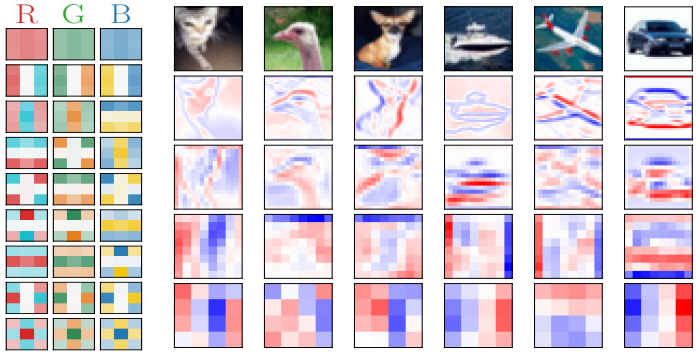

Figure 3: Convolutional Neural LoFi on CIFAR-10 animal-vs.-vehicle. *Left:* top first-layer RGB filters. *Right:* activations of the sixth LoFi feature across depths.

Convolutional layers (CNN) — Neural LoFi is not restricted to fully connected architectures: the moment operator can be built in the feature space exposed by the architecture, naturally incorporating locality, weight sharing, equivariance, or graph structure [17]. 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:

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

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. The convolutional experiment gives direct visual evidence for low-degree compositionality. Applied to pixels, LoFi recovers color-sensitive, contrast, and edge-like filters, later resembling finite-difference or Laplacian-like stencils, echoing classical edge-filter emergence in natural image models [65] and early CNN filters learned by gradient descent [47, 71, 98]. These structures emerge *without backpropagation or end-to-end optimization*, simply by diagonalizing the Neural LoFi label-weighted moment operator. Across LoFi layers, the same feature evolves from generic edges to structured spatial patterns (Fig. 3), matching the theory that repeated low-degree extraction builds progressively structured features. Further spectra and datasets are in Appendix I.

Conclusion

Taken together, out theory based on the analysis of Neural LoFi predicts and explain many aspects of deep learning: depth helps by turning a hard global target into a sequence of detectable low-degree signals. Neural LoFi makes this picture mathematically explicit, offering a concrete route to study deep feature learning through spectral transitions rather than treating learned representations as a black box. We hope this will trigger further work on this topic.

References

[1] Emmanuel Abbe, Enric Boix-Adsera, 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.

[2] Emmanuel Abbe, Enric Boix Adsera, 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.

[3] 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.

[4] 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.

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

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

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

[8] 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. URL https://openreview.net/forum?id=m3Sz3tFxIV.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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. URL https://openreview.net/forum?id=Lw0kC75dY0.

[20] 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.

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

[22] 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.

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

[24] 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.

[25] 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. URL https://openreview.net/forum?id=5X6PL4906S.

[26] 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.

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

[28] 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. URL https://openreview.net/forum?id=5JcDVsV8pf.

[29] Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. doi: 10.1137/0707001.

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

[31] 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. URL https://openreview.net/forum?id=Q3yLIIkt7z.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[43] 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.

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

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

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

[47] 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. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

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

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

[50] 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.

[51] 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.

[52] 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.

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

[54] 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.

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

[56] 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.

[57] 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.

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

[59] 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.

[60] 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.

[61] 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.

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

[63] 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.

[64] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] Aaditya Radhakrishnan, Daniel Beaglehole, Pratik Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

[70] 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.

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

[72] 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.

[73] 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.

[74] 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.

[75] 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.

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

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

[78] 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.

[79] 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.

[80] 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.

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

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

[83] 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.

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

[85] 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.

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

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

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

[96] 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.

[97] 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.

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

[99] 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.

Appendix A. Further discussion on Neural LoFi

A.1. Neural Low-Degree Filtering

A.1.1. 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. C. 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. E). 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

$$\boldsymbol{z}_0(\boldsymbol{x}) = \boldsymbol{x}, \qquad \boldsymbol{h}_\ell(\boldsymbol{x}) = \boldsymbol{W}_\ell \boldsymbol{z}_{\ell-1}(\boldsymbol{x}), \qquad \boldsymbol{z}_\ell(\boldsymbol{x}) \in \mathbb{R}^{p_\ell} = \sigma(\boldsymbol{h}_\ell(\boldsymbol{x})), \qquad \ell = 1, \dots, L, \tag{4}$$

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

$$\hat{f}(x) = \langle \boldsymbol{a}_L, \boldsymbol{z}_L(x) \rangle. \tag{5}$$

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 C, that under a layer-wise vanishing initialization scheme $a_L \ll W_{L-1} \ll \cdots W_1$, 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:

$$\hat{\boldsymbol{u}}^\ell := \frac{1}{n} \sum_{\mu=1}^n y_\mu \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu), \quad \widehat{\boldsymbol{C}}^{(\ell)} := \frac{1}{n} \sum_{\mu=1}^n y_\mu \, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)^\top. \tag{6}$$

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 with initialization scale $a_L \ll W_{L-1} \ll \cdots W_1$. Then, for any $\eta > 0$, and layer ℓ, any fixed neuron $w_{\ell,i}$ for $i \in p_\ell$, satisfies for small enough time:*

$$\boldsymbol{w}_{\ell,i}(t+1) - \boldsymbol{w}_{\ell,i}(t) \approx \eta \, c_0 \hat{\boldsymbol{u}}_\ell + \eta, c_1 \widehat{C}^{(\ell)} \, \boldsymbol{w}_{\ell,i}(t) + O(\left\|\boldsymbol{w}_{\ell,i}(0)\right\|^2), \tag{7}$$

for some constants $c_0, c_1 > 0$.

The first term in Eq. 7 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 $\widehat{C}^{(\ell)} = \hat{V} \Lambda \hat{V}^\top$ denote the eigendecomposition.

For small step-size, the linear term $c_1\widehat{C}^{(\ell)}\,\boldsymbol{w}_{\ell,i}(t)$ yields the per-neuron approximation $\boldsymbol{w}_{\ell,i}(t) \approx \exp(c_1\widehat{\boldsymbol{C}}^{(\ell)}t)\,\boldsymbol{w}_{\ell,i}(0)$. 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:

$$W_\ell(t) \;\approx\; W_\ell(0)\,\exp\!\big(c_1\,\widehat{\boldsymbol{C}}^{(\ell)}\,t\big) \;=\; \underbrace{W_\ell(0)\,\hat{\boldsymbol{V}}}_{\text{random transformation}} \;\times\; \underbrace{\exp(c_1\,\Lambda\,t)\,\hat{\boldsymbol{V}}^\top}_{\text{spectral filter}\,\rightarrow\,\text{top-eigenvector projection}}\quad. \tag{8}$$

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 $\exp(c_1\lambda_r^{(\ell)}t)$, exponentially amplifying directions with the largest eigenvalues, so an early-stopped power iteration will approximate hard projection onto the top-k eigenvectors as t grows. 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 weighing in Equation 8 holds under vanishing initalization and small time-scales, we expect, in general, the network to maintain a preference towards features corresponding to larger eigenvalues for other training regimes. We discuss how weighing regimes arise under different training settings for two-layer networks in App. D. The simplest choice of such weights is top-k selection: $w(\lambda_r^{(\ell)}) = 1$ for $r \le k$ for some k, and 0 otherwise, which we use in Alg.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.*

A.1.2. THE NEURAL LOW-DEGREE FILTERING ALGORITHM

The decomposition in Eq. (8) 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. (8), producing the next representation.

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

$$\boldsymbol{g}_\ell(\boldsymbol{x}) = \widehat{\boldsymbol{V}}_\ell^\top\,\boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \tag{9}$$

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 $\widehat{\boldsymbol{C}}^{(\ell)}$ extracts the most relevant directions in feature space $\boldsymbol{z}_{\ell-1}(\boldsymbol{x})$ weighted by their 2^{nd}-order correlation with y.

Algorithm 1: Neural Low-Degree Filtering

Input: 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:

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

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

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

$\boldsymbol{z}_\ell(\boldsymbol{x}) \leftarrow \frac{\sigma(\boldsymbol{R}_\ell \boldsymbol{g}_\ell(\boldsymbol{x}))}{\sqrt{p_\ell}}$, $\boldsymbol{R}_\ell \in \mathbb{R}^{p_\ell \times k_\ell}$, $(\boldsymbol{R}_\ell)_i \sim \mathcal{N}(0, s_{k,\ell} I_{k_\ell+1})$.; `// Random feature map with variance` $s_{k,\ell} \leftarrow \hat{\mathbb{E}}_n[\|g_\ell(\boldsymbol{x})\|^2]^{-1}$

end

return *Final predictor* $\hat{f}(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{z}_L(\boldsymbol{x}) \rangle$ *with* $\boldsymbol{a}$ *fitted with Ridge or Logistic Regression.*

A.1.3. 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 weights R_ℓ. However, we show below that the resulting features $\langle \hat{\boldsymbol{u}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle, \langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle_{j=1,\dots,k_\ell}$ 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

$$K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{z}_{\ell-1}(\boldsymbol{x}), \boldsymbol{z}_{\ell-1}(\boldsymbol{x}') \rangle \tag{10}$$

be the kernel induced by the current representation, and let $\mathcal{H}_{\ell-1}$ be its Reproducing Kernel Hilbert Space (RKHS [81, 84]). Through an analogue of the classical representer theorem (Lemma 10), we show that for any $j \in k_\ell$, the features $\varphi_j(\boldsymbol{x}) \coloneqq \langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}) \rangle$, 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 criteria:

$$\hat{\boldsymbol{v}}_1^{(\ell)} \in \operatorname*{arg\,max}_{\|\boldsymbol{v}\|_2=1} \left| \frac{1}{n} \sum_{\mu=1}^n y_\mu \langle \boldsymbol{v}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu) \rangle^2 \right| \rightarrow \varphi_1^{(\ell)} \in \operatorname*{arg\,max}_{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1} \left| \frac{1}{n} \sum_{\mu=1}^n y_\mu \, \varphi(\boldsymbol{x}_\mu)^2 \right|. \tag{11}$$

Neural LoFi therefore 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.G, how this can also be turned into a practical kernel version of our approach. This is formalized through the following result, furthermore showing that they converge to the eigenfunctions of the limiting infinite-width kernel as the layer width grows:

Theorem 2 (Variational characterization and infinite-width convergence) *Let* $\hat{\boldsymbol{v}}_0^\ell$ *denote the linear component and* $[\hat{\boldsymbol{v}}_1^{(\ell)}, \dots, \hat{\boldsymbol{v}}_{k_\ell}^{(\ell)}]$ *the ordered eigenvectors from Algorithm 1. Define the linear*

feature

$$\psi^\ell(\boldsymbol{x}) \coloneqq \langle \hat{\boldsymbol{u}}^\ell,\, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\rangle,$$

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

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

$$\psi^\ell \;=\; \operatorname*{arg\,max}_{\psi:\, \|\psi\|_{\mathcal{H}_{\ell-1}}=1} \left| \widehat{\mathbb{E}}_n\big[y\,\psi(\boldsymbol{x})\big] \right|. \tag{12}$$

(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:*

$$\varphi_k \;=\; \operatorname*{arg\,max}_{\substack{\varphi:\, \|\varphi\|_{\mathcal{H}_{\ell-1}}=1 \\ \varphi \perp \varphi_1, \dots, \varphi_{k-1}}} \left| \widehat{\mathbb{E}}_n\big[y\,\varphi(\boldsymbol{x})^2\big] \right|. \tag{13}$$

(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* $\{\phi_k^\infty\}_{k\ge 1}$ *be its eigenfunctions ordered by decreasing eigenvalue magnitude, with the selected limiting eigenvalues (or clusters) separated from the rest of the 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, and* $\phi_1^\infty, \cdots, \phi_k^\infty$ *satisfy a set of deterministic variational criterion.*

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

$$K_\ell(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{r}\sim\pi_\ell}\Big[\sigma(\boldsymbol{r}^\top \boldsymbol{g}_\ell(\boldsymbol{x}))\sigma(\boldsymbol{r}^\top \boldsymbol{g}_\ell(\boldsymbol{x}'))\Big]. \tag{14}$$

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.

A.2. 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.

A.2.1. EMERGENCE AND EFFECTIVE DIMENSION

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 [6, 72, 80, 92]. Neural LoFi provides a simple

mechanism for this phenomenon. By Theorem 2, the kth feature extracted by layer ℓ maximizes the following second-order *empirical* correlation:

$$\widehat{\rho}_\ell^{(k)} := \sup_{\|\varphi\|_{\mathcal{H}_{\ell-1}}=1, \varphi\perp\varphi_1,\cdots,\varphi_{k-1}} \left|\widehat{\mathbb{E}}_n\left[y\,\varphi(\boldsymbol{x})^2\right]\right|, \tag{15}$$

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:

$$\mathcal{S}_k^\ell := \left\{\varphi\in\mathcal{H}_{\ell-1} : \|\varphi\|_{\mathcal{H}_{\ell-1}}=1,\ \varphi\perp\varphi_1,\cdots,\varphi_{k-1}\right\}. \tag{16}$$

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

$$c_\ell(\varphi) := \mathbb{E}\left[y\,\varphi(\boldsymbol{x})^2\right], \qquad \widehat{c}_{\ell,n}(\varphi) := \widehat{\mathbb{E}}_n\left[y\,\varphi(\boldsymbol{x})^2\right]. \tag{17}$$

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 $\mathcal{S}_k^\ell$,

$$\tau_\ell^k(n) := \sup_{\varphi\in\mathcal{S}_k^\ell}\left|\widehat{c}_{\ell,n}(\varphi)-c_\ell(\varphi)\right|. \tag{18}$$

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

$$\rho_\ell^{(k)} := \sup_{\varphi\in\mathcal{S}_k^\ell}\left|\mathbb{E}\left[y\,\varphi(\boldsymbol{x})^2\right]\right|. \tag{19}$$

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

$$\rho_\ell^{(k)} \gg \tau_\ell^k(n). \tag{20}$$

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 *a la* Baik-Ben Arous-Péché/Edwards-Jones (BBPEA) [9, 32]. We show in Appendix H, using local Rademacher-complexity bounds over $\mathcal{S}_k^\ell$, that up to poly-logarithmic factors:

$$\tau_\ell^k(n) \lesssim r_\ell^k\sqrt{\frac{D_{\ell,k}^{\mathrm{eff}}(r_\ell^k)}{n}}, \tag{21}$$

where $D_{\ell,k}^{\mathrm{eff}}(r)$ is the *residual effective dimension* of features with scale r (see Def.13) within the current RKHS after removing the previously extracted features, and $r_\ell^k := \arg\max_r(r\sqrt{D_{\ell,k}^{\mathrm{eff}}(r)})$ is the dominant scale of fluctuations among candidate features. This can be estimated from the empirical spectrum of the kernel, making Eq.(20) a data-driven diagnostic for emergence of concepts.

We develop this formalization of emergence further in Appendix H. The criterion recovers known sample-complexity scales in solvable hierarchical models [30, 86]. 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 3, individual eigenvector overlaps rise near the predicted thresholds; (see Fig. 10 in Appendix). This makes Neural LoFi a quantitative theory of layerwise concept emergence.

A.2.2. 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 [60, 87], 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 (15) 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 converts one hard estimation problem in the input space into several easier estimation problems in adapted representation spaces.

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: [18, 20, 28, 37, 73, 86]. 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 [28]. 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. (15), 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 [95].

Appendix B. 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 [77], spiked matrix–tensor models [74, 78, 79], and BBPEA-type selection mechanisms [9, 32]. Related label-weighted second-order operators also appear in single-index and multi-index estimation, where they yield spectral initializations and recovery procedures [53, 54, 61, 89]. More recently, similar operators have been connected to the early dynamics of shallow neural networks through Hessian or Hessian-like interpretations [15, 30, 62, 99]. Neural LoFi extends this viewpoint to iterative multi-layer methods.

Beyond second order — While the second-order operator (6) 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 [2, 11, 24, 25, 88]. 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 [7, 43, 51, 93]. This limitation also points to a natural future directions: Multi-pass gradient descent can break the constraints imposed by information and leap exponents by repeatedly transforming the effective label and representation [27] as well as through staircase mechanisms [1, 12, 78, 85]. An interesting direction for future works 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.D.

Scattering, coarse-graining, and multiscale representations — Our construction is also related in spirit to scattering transforms and their descendants [3]. 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 [55].

Mechanistic interpretability — Neural LoFi is complementary to mechanistic interpretability, which aims to reverse-engineer trained networks into interpretable features and circuits [33, 64]. 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 [34].

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, iterative algorithms for adaptive feature learning and dimensionality reduction [69, 70]. This is close in philosophy to Neural LoFi: both seek a simple iterative surrogate for representation learning. The constructions differ, however. AGOP-based methods use gradient covariances of a learned predictor, whereas Neural LoFi crucially uses the next order approximation with the Hessian.

Prunning — Neural LoFi gives a feature-space perspective on why large networks and pruning are not contradictory [35, 41, 48]: Large width provides a rich space for discovering task-correlated directions, while spectral pruning keeps only the few directions that finite data can reliably support.

Appendix C. Neural LoFi from Gradient Descent

In this section, our goal is to prove Proposition 1, 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

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

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

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

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

Assumption 3 (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 4 *$\sigma : \mathbb{R} \to \mathbb{R}$ 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 5 (Layerwise GD approximation) *Under Assumptions 3–4, 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.:

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

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

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

where $c_1 = \sigma''(0)$. For every $0 \leq t \leq T_\ell$ the accumulated residual obeys

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

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 6 *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 3.*

C.1. Proof of Theorem 5

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 $\ell+1, \ldots, L$. Define the layer-to-layer Jacobian

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

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 3, 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

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

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

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

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 $\gamma_\ell = \alpha_{L+1}\prod_{m=\ell+1}^{L}\alpha_m$,

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

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 4 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

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

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). ∎

Appendix D. Two-layer networks

D.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. [1, 2, 4, 12–14, 24–27, 39].)

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 [12], and how the regime IE ≤ 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. [15, 30, 62, 99]. We then describe how, when 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. [2, 4, 12, 14, 27, 39].

From the LoFi correlation criterion to the information exponent Specializing the first-layer ($\ell = 1$) Neural LoFi criterion of equation (15) 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. Expanding $f^\star$ in the Hermite basis, $f^\star(\boldsymbol{x}) = \sum_{k\geq 0} \langle f^\star, H_k\rangle\, H_k(U^{\star\top}\boldsymbol{x})$, 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 [12], define the *information exponent* of $f^\star$ as

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

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 $C^\star := \mathbb{E}[y\, \boldsymbol{x}\boldsymbol{x}^\top] - \mathbb{E}[y]\, I_d$;
- if $\mathrm{IE}(f^\star) \geq 3$, 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 [1, 12, 13, 24].

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 7 (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 [2, 12]. When $\mathrm{IE} \ge 3$, Proposition 7 fails by construction, and recovery of $V^\star$ requires either non-Gaussian preprocessing of y (e.g. polynomial reweightings, as in [24, 25]) or genuinely deeper compositionality, which is the regime studied in Appendix C 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 [8]. 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 dynamcis through saddles by a single eigendecomposition of $\hat{C}^{(1)}$.

Finaly, we note that with the use of a different loss other than the squared loss or with data reuse or label transformations [24, 25], the criteria 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 [11, 25] ≥ 2 instead of the information exponent [12].

Appendix E. Vector labels

In the main text we stated Theorem 5 for scalar labels $y\in\mathbb{R}$. The variational criteria extend in a natural way to vector-valued labels $\boldsymbol{y}\in\mathbb{R}^d$, as encountered in multi-class classification (e.g. one-hot or softmax targets), multi-task regression, or any problem with a multi-dimensional response. We sketch here the corresponding generalization.

Per-coordinate correlation operator Given a feature $\psi:\mathcal{X}\to\mathbb{R}$ and a vector label $\boldsymbol{y}=(y_1,\dots,y_m)$, define the *label-feature correlation vector*

$$\mathbf{c}_n[\psi] \;:=\; \big(\widehat{\mathbb{E}}_n[y_1\,\psi(\boldsymbol{x})],\,\dots,\,\widehat{\mathbb{E}}_n[y_m\,\psi(\boldsymbol{x})]\big)\;\in\;\mathbb{R}^d, \tag{45}$$

and analogously the second-order correlation vector $\mathbf{c}_n^{(2)}[\varphi] := \big(\widehat{\mathbb{E}}_n[y_a\,\varphi(\boldsymbol{x})^2]\big)_{a=1}^m$. The scalar quantities $\widehat{\mathbb{E}}_n[y\,\psi(\boldsymbol{x})]$ and $\widehat{\mathbb{E}}_n[y\,\varphi(\boldsymbol{x})^2]$ used in Theorem 5 are recovered when $m=1$.

To obtain a variational principle one needs a scalar score on $\mathbf{c}_n[\psi]$. A natural generalization is to use the squared Euclidean norm of the correlation vector,

$$\mathcal{S}^{\ell_2}(\psi) \;:=\; \|\mathbf{c}_n[\psi]\|_2^2 \;=\; \sum_{a=1}^{d}\big(\widehat{\mathbb{E}}_n[y_a\,\psi(\boldsymbol{x})]\big)^2, \tag{46}$$

which selects features that are simultaneously well aligned with as many label coordinates as possible.

With the above generalization, the analogues of parts (i)–(ii) of Theorem 5 read as follows:

(i′) *Linear features.* The linear features are defined recursively as:

$$\psi_k^\ell \;\in\; \arg\max_{\psi:\,\|\psi\|_{\mathcal{H}_{\ell-1}}=1,\psi\perp\psi_1^\ell,\dots,\psi_{k-1}^\ell} \big\|\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\psi(\boldsymbol{x})]\big\|_2^2, \tag{47}$$

Note that unlike the scalar label setting in Theorem 5, the linear correlation criteria now produces m features, corresponding to the m-singular vectors of the correlation matrix $\frac{1}{n}\sum_{\mu=1}^n y_\mu \boldsymbol{z}_{\ell-1}^\top(\boldsymbol{x})$

(ii′) *Second-order features.* For each $k=1,\dots,k_\ell$,

$$\varphi_k \;\in\; \arg\max_{\substack{\varphi:\,\|\varphi\|_{\mathcal{H}_{\ell-1}}=1\\ \varphi\perp\varphi_1,\dots,\varphi_{k-1}}} \big\|\widehat{\mathbb{E}}_n[\boldsymbol{y}\,\varphi(\boldsymbol{x})^2]\big\|_2^2, \tag{48}$$

successively orthogonalized to the previously selected features.

Appendix F. A Tractable 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 [86], building on the fundamental earlier works [36, 63, 91]. This allows 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. [18–20, 28, 36, 37, 63, 73, 86, 91]). 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.

F.1. Setting

We focus on the high-dimensional Gaussian teacher-student model of [86], 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

$$\boldsymbol{x} \in \mathbb{R}^d \quad \longrightarrow \quad h^{(1)}(\boldsymbol{x}) \in \mathbb{R}^{d_1} \quad \longrightarrow \quad h^{(2)}(\boldsymbol{x}) \in \mathbb{R} \quad \longrightarrow \quad y. \tag{49}$$

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.[1] We write $\langle \cdot, \cdot \rangle$ for the Frobenius product between tensors of the same order.[2] 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

$$A_i^{(1)} \in (\mathbb{R}^d)^{\otimes q}_{\mathrm{sym}}, \qquad i = 1, \dots, d_1, \qquad d_1 = d^{\epsilon}, \tag{50}$$

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

$$A^{(2)} \in \mathbb{R}^{d_1 \times d_1}. \tag{51}$$

1. 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 $\sqrt{q!}\, H_q(z) = (-1)^q e^{z^2/2} \frac{d^q}{dz^q} \left(e^{-z^2/2} \right)$.
2. For example, if $\boldsymbol{x} \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times m}$ is symmetric, then $\langle A, H_2(\boldsymbol{x}) \rangle = \frac{1}{\sqrt{2}} \left(\boldsymbol{x}^\top A \boldsymbol{x} - \mathrm{Tr}(A) \right)$.

The latent variables and the label are then defined as

$$h_i^{(1)}(\boldsymbol{x}) = \left\langle A_i^{(1)}, H_q(\boldsymbol{x}) \right\rangle, \qquad i = 1, \dots, d_1, \tag{52}$$

$$h^{(2)}(\boldsymbol{x}) = \left\langle A^{(2)}, H_2(h^{(1)}(\boldsymbol{x})) \right\rangle, \tag{53}$$

$$y = g^\star(h^{(2)}(\boldsymbol{x})). \tag{54}$$

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 [86] 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.

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.

F.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

$$\phi_\mu^{(1)} = \frac{1}{\sqrt{p_1}} \sigma_1(W_1 x_\mu) \in \mathbb{R}^{p_1}. \tag{55}$$

The first spectral operator is

$$\widehat{C}_1 = \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \phi_\mu^{(1)} \phi_\mu^{(1)\top}. \tag{56}$$

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

$$\widehat{h}_\mu^{(1)} = \widehat{V}_1^\top \phi_\mu^{(1)} \in \mathbb{R}^{d_1}. \tag{57}$$

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

$$\phi_\mu^{(2)} = \frac{1}{\sqrt{p_2}} \sigma_2(W_2 \widehat{h}_\mu^{(1)}) \in \mathbb{R}^{p_2}. \tag{58}$$

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

$$\widehat{v}_2 = \frac{1}{n} \sum_{\mu=1}^{n} y_\mu \phi_\mu^{(2)}. \tag{59}$$

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

$$\widehat{h}_\mu^{(2)} = \widehat{v}_2^\top \phi_\mu^{(2)}. \tag{60}$$

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,

$$\widehat{g} \in \arg\min_{g \in \mathcal{G}} \frac{1}{n} \sum_{\mu=1}^{n} \Big(y_\mu - g(\widehat{h}_\mu^{(2)}) \Big)^2 + \lambda \|g\|_{\mathcal{G}}^2. \tag{61}$$

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.

F.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

$$n \gg d^{q+\epsilon}. \tag{62}$$

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

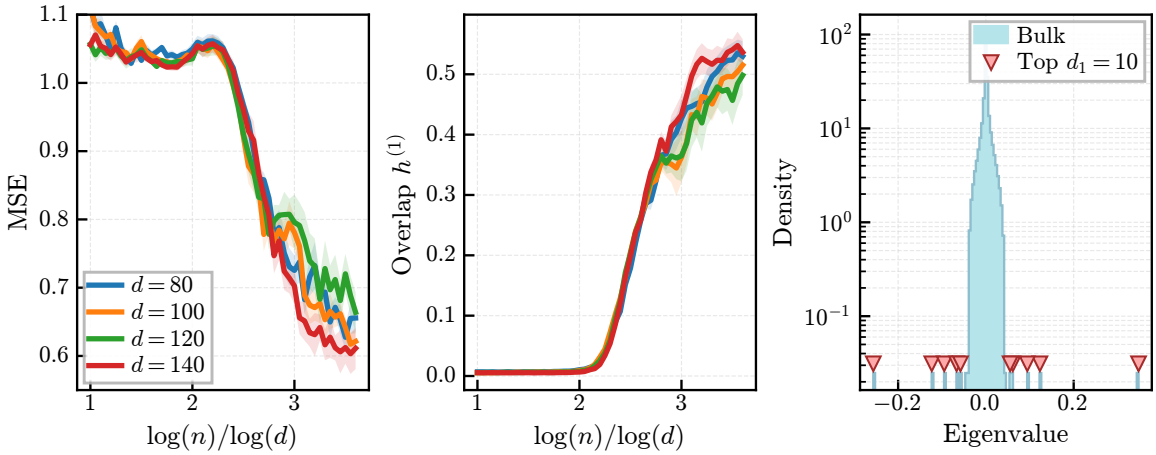

Figure 4: **Neural LoFi on the theoretical setting.** We consider the two-level target defined in Section F, with quadratic first layer ($k = 2$), latent dimension $d_1 = \lfloor d^\epsilon \rfloor$, $\epsilon = \frac{1}{2}$, and final readout $g^\star(t) = \tanh(t)$. For $d \in \{80, 100, 120, 140\}$, we use first-layer random-feature widths $p_1 \in \{20000, 30000, 40000, 50000\}$ and second-layer widths $p_2 \in \{512, 768, 1024, 1280\}$. The final one-dimensional non-linearity is fitted with a polynomial kernel of maximal degree 5, using ridge regularization 10^{-6} and kernel regularization 10^{-4}. **Left:** Mean Squared Error (MSE) of the final predictor as a function of α, for input dimensions $d \in \{80, 100, 120, 140\}$. **Center:** Overlap between the recovered first-layer representation $\widehat{h}^{(1)}$ and the ground-truth latent variables $h^{(1)}$. **Right:** Spectrum of the first random-feature spectral operator $\widehat{C}_1$ for $d = 100$ at $\alpha = 3$, where $d_1 = 10$.

This prediction is the concrete instance of the emergence criterion in Eq. (21) in the main test. Indeed, Eq. (21) states that the empirical fluctuation level is governed by the effective dimension of the current residual feature class. Section H.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 4 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.

F.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 [86].

Let us expand the first-layer activation in Hermite as

$$\sigma_1(z) = \sum_{q\geq 2} c_q H_q(z), \qquad c_q = \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\sigma_1(z)H_q(z)\right], \tag{63}$$

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

$$H_q(\langle \boldsymbol{w}_{1,a}, \boldsymbol{x}\rangle) = \left\langle \boldsymbol{w}_{1,a}^{\otimes q}, H_q(\boldsymbol{x})\right\rangle. \tag{64}$$

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:

$$\frac{1}{\sqrt{p_1}} H_q(W_1\boldsymbol{x}) = P_q H_q(\boldsymbol{x}), \quad \text{with} \quad P_q := \frac{1}{\sqrt{p_1}} \begin{bmatrix} F[\boldsymbol{w}_{1,1}^{\otimes q}]^\top \\ \vdots \\ F[\boldsymbol{w}_{1,p_1}^{\otimes q}]^\top \end{bmatrix} \in \mathbb{R}^{p_1\times D_q}. \tag{65}$$

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

$$\phi_\mu^{(1)} = \frac{1}{\sqrt{p_1}}\sigma_1(W_1\boldsymbol{x}_\mu) = \sum_{q\geq 2} c_q P_q H_q(\boldsymbol{x}_\mu). \tag{66}$$

We denote by

$$\widehat{C}_H^{(q)} := \frac{1}{n}\sum_{\mu=1}^{n} y_\mu H_q(\boldsymbol{x}_\mu) H_q(\boldsymbol{x}_\mu)^\top \tag{67}$$

the degree-q Hermite moment matrix appearing in the middle. If one uses the centered second-order Hermite convention of [86], 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

$$\widehat{C}_{\mathrm{RF}}^{(q)} = c_q^2 P_q \widehat{C}_H^{(q)} P_q^\top. \tag{68}$$

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 [94]. Define the normalized RF Gram

$$G_2 := D_2 P_2^\top P_2. \tag{69}$$

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 [94], one has the decomposition

$$G_2 = I_{D_2} + K_2 + R_2, \qquad K_2 = \theta_d e e^\top, \qquad |\theta_d| \leq \widetilde{O}(d), \tag{70}$$

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

$$\|R_2\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right), \tag{71}$$

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

$$A^{(1)}_{\mathrm{RF}} := \sqrt{D_2}\, A^{(1)} P_2^\top \in \mathbb{R}^{d_1 \times p_1}. \tag{72}$$

We can now state the projected RF analogue of Theorem 3.1 in [86].

Theorem 8 (Projected RF Hermite recovery, quadratic case) *Assume the setting and normalization of Theorem 3.1 in [86], 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:*

$$\begin{aligned}
&\sqrt{d_1}\left\| \frac{D_2}{c_2^2} A^{(1)}_{\mathrm{RF}} \widehat{C}^{(2)}_{\mathrm{RF}} A^{(1)\top}_{\mathrm{RF}} - \nu_1 A^{(2)} \right\|_{\mathrm{op}} \\
&\quad \le \widetilde{O}\left(\sqrt{\frac{d^2 d_1}{n}}\right) + \widetilde{O}\left(\frac{d_1}{\sqrt{d}}\right) + \widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right) \\
&\qquad + \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1^2}{d^2} + \sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\left(1 + \sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\right).
\end{aligned} \tag{73}$$

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 $A^{(1)}_{\mathrm{RF}}$, $\widehat{C}^{(2)}_{\mathrm{RF}}$, and G_2, we have the exact identity

$$\frac{D_2}{c_2^2} A^{(1)}_{\mathrm{RF}} \widehat{C}^{(2)}_{\mathrm{RF}} A^{(1)\top}_{\mathrm{RF}} = A^{(1)} G_2 \widehat{C}^{(2)}_H G_2 A^{(1)\top}. \tag{74}$$

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

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:

$$\|A^{(1)}e\|_2 \le \widetilde{O}\left(\frac{\sqrt{d_1}}{d}\right), \tag{75}$$

$$\|C\|_{\mathrm{op}} \le \widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right), \tag{76}$$

$$\|e^\top C A^{(1)\top}\|_2 \le \widetilde{O}\left(\sqrt{\frac{d_1}{n}} + \frac{1}{d}\right), \tag{77}$$

$$|e^\top C e| \le \widetilde{O}\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d_1}}{d^2}\right). \tag{78}$$

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 [86], 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

$$G_2 C G_2 - C = K_2 C + C K_2 + K_2 C K_2 + \text{terms containing } R_2.$$

For the first trace term,

$$A^{(1)} K_2 C A^{(1)\top} = \theta_d (A^{(1)}e)(e^\top C A^{(1)\top}),$$

and therefore, using (75)–(77),

$$\sqrt{d_1}\|A^{(1)} K_2 C A^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1}{d}\right).$$

The transpose term $A^{(1)} C K_2 A^{(1)\top}$ is bounded identically. For the quadratic trace term,

$$A^{(1)} K_2 C K_2 A^{(1)\top} = \theta_d^2 (A^{(1)}e)(e^\top C e)(A^{(1)}e)^\top,$$

so by (75) and (78),

$$\sqrt{d_1}\|A^{(1)} K_2 C K_2 A^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}} + \frac{d_1^2}{d^2}\right).$$

Finally, all terms containing R_2 are controlled by $\|R_2\|_{\mathrm{op}}$ and (76). The worst mixed terms are $K_2 C R_2$ and $R_2 C K_2$, and they satisfy

$$\sqrt{d_1}\|A^{(1)} K_2 C R_2 A^{(1)\top}\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}} + \frac{D_2}{p_1}\right)\right),$$

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

$$\begin{aligned}\sqrt{d_1}\left\|A^{(1)}\left(G_2CG_2-C\right)A^{(1)\top}\right\|_{\mathrm{op}} &\le \widetilde{O}\left(d_1\sqrt{\frac{d_1}{n}}+\frac{d_1}{d}+\frac{d_1^2}{d^2}\right)\\ &\quad+\widetilde{O}\left(\sqrt{d_1}\left(\sqrt{\frac{D_2}{p_1}}+\frac{D_2}{p_1}\right)\left(1+\sqrt{\frac{D_2}{p_1}}+\frac{D_2}{p_1}\right)\right). \quad (79)\end{aligned}$$

Theorem 3.1 of [86] gives

$$\sqrt{d_1}\left\|A^{(1)}\widehat{C}_H^{(2)}A^{(1)\top}-\nu_1A^{(2)}\right\|_{\mathrm{op}} \le \widetilde{O}\left(\sqrt{\frac{d^2d_1}{n}}\right)+\widetilde{O}\left(\frac{d_1}{\sqrt{d}}\right)+\widetilde{O}\left(\frac{1}{\sqrt{d_1}}\right).$$

Combining this with (79) proves (73). ■

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 [94]. These contraction terms are subleading on the random planted subspace by the same vanishing-contraction estimates used in Appendix A.1–A.4 of [86]. 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

$$n \gg D_q d_1, \qquad p_1 \gg D_q, \quad (80)$$

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.

F.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

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{n_{\mathrm{test}}}\sum_{\mu=1}^{n_{\mathrm{test}}}\left(\hat{y}_\mu - y_\mu\right)^2, \quad (81)$$

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

$$\mathrm{Overlap}(h^{(1)}, \widehat{h}^{(1)}) = \left\| H^{(1)} \hat{H}^{(1)\top} \right\|_F^2, \quad H^{(1)} = \left(h_\mu^{(1)}\right)_{\mu \le n_{\mathrm{test}}}, \quad \hat{H}^{(1)} = \left(\hat{h}_\mu^{(1)}\right)_{\mu \le n_{\mathrm{test}}}. \tag{82}$$

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

$$d_1 = \lfloor d^\epsilon \rfloor, \qquad \epsilon = \frac{1}{2}, \qquad n = \lfloor d^\alpha \rfloor, \tag{83}$$

and generate labels according to

$$y = g^\star(h^{(2)}(\boldsymbol{x})), \qquad g^\star(t) = \tanh(t). \tag{84}$$

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,

$$\boldsymbol{w}_{1,a} \sim \mathrm{Unif}(\mathbb{S}^{d-1}), \qquad \boldsymbol{w}_{2,a} \sim \mathrm{Unif}(\mathbb{S}^{d_1-1}), \tag{85}$$

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

$$\sigma_{\perp\{0,1\}}(z) = \mathrm{ReLU}(z) - c_0 H_0(z) - c_1 H_1(z), \qquad c_r = \mathbb{E}_{G\sim\mathcal{N}(0,1)}\left[\mathrm{ReLU}(G) H_r(G)\right]. \tag{86}$$

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

$$p_1 \gtrsim D_2, \qquad D_2 = \binom{d+1}{2}, \tag{87}$$

in agreement with the finite-width requirement appearing in Theorem 8. In the second layer, the width p_2 is chosen large enough with an equivalent regime as for the first layer, $p_2 \ge d_1^2$, 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 and we follow the learning strategy detailed in Sec.F.2 being simply Neural LoFi algorithm, we only incorporate a 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 4 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.

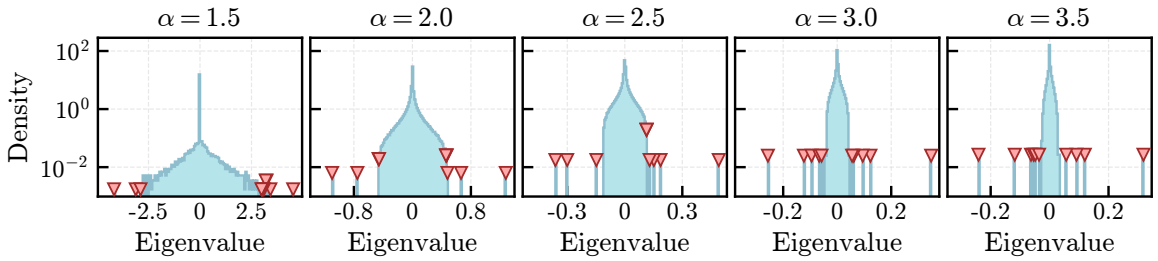

Figure 5: **Spectral emergence in the random-feature Neural LoFi estimator.** Spectrum of the first random-feature spectral operator $\widehat{C}_1$ for the hierarchical teacher-student model, shown at increasing sample exponents $\alpha = \log(n)/\log(d)$. Blue histograms display the bulk eigenvalue 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 the first-layer signal predicted by the sample-complexity scale $n \gg d^{2+\epsilon}$ in the quadratic setting $q = 2$.

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.

Appendix G. Neural LoFi Kernel

G.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 [81] 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

$$f(\boldsymbol{x}) = \langle f, K_{\boldsymbol{x}} \rangle_{\mathcal{H}} \qquad \text{for all } f \in \mathcal{H}. \tag{88}$$

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 (88) 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 [5], every symmetric positive semi-definite kernel K uniquely determines an RKHS $\mathcal{H}_K$ in which (88) 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 9 (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*

$$Tf \;=\; \sum_{i\geq 1} \lambda_i \,\langle f, e_i\rangle\, e_i \qquad \textit{for all } f \in \mathcal{H}. \tag{89}$$

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

$$(T_K f)(\boldsymbol{x}) \;:=\; \int_{\mathcal{X}} K(\boldsymbol{x}, \boldsymbol{x}')\, f(\boldsymbol{x}')\, d\mu(\boldsymbol{x}'). \tag{90}$$

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 9 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 [59] 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

$$K(\boldsymbol{x}, \boldsymbol{x}') \;=\; \sum_{i\geq 1} \lambda_i\, e_i(\boldsymbol{x})\, e_i(\boldsymbol{x}'). \tag{91}$$

Relation between the integral operator and the RKHS The Mercer decomposition (91) provides an explicit isometric description of the RKHS $\mathcal{H}_K$ in terms of the spectral data of T_K. Restricting to indices with $\lambda_i > 0$,

$$\mathcal{H}_K \;=\; \Big\{ f = \sum_{i:\lambda_i>0} a_i\, e_i \;:\; \|f\|_{\mathcal{H}_K}^2 \;:=\; \sum_{i:\lambda_i>0} \frac{a_i^2}{\lambda_i} < \infty \Big\}, \tag{92}$$

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).

G.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 10 (Representer property of LoFi features) *Let $\{\hat{\boldsymbol{v}}_j^{(\ell)}\}_{j=1}^{k_\ell}$ denote any minimizers of the empirical objective in (11). 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.*

$$\hat{\boldsymbol{v}}_j^{(\ell)} \in \operatorname{span}\big\{\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_1), \dots, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_n)\big\}, \qquad j = 1, \dots, k_\ell. \tag{93}$$

Proof [Proof of Theorem 10] Fix any index $j \in \{1, \dots, k_\ell\}$. The empirical objective in (11) depends on $\hat{\boldsymbol{v}}_j^{(\ell)}$ only through the inner products $\langle \hat{\boldsymbol{v}}_j^{(\ell)}, \boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\rangle$, $\mu = 1, \dots, n$, 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 $\mathcal{S} := \operatorname{span}\{\boldsymbol{z}_{\ell-1}(\boldsymbol{x}_\mu)\}_{\mu=1}^n \subset \mathbb{R}^{p_{\ell-1}}$ 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 (93). ■

G.3. RKHS-Euclidean Equivalence

We now establish a fundamental result that underlies all subsequent proofs in this section. By Lemma 10, we know features lie in the span of training features. This allows us to identify RKHS constraints with Euclidean constraints on weight vectors.

Lemma 11 (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.

G.4. Proof of Theorem 2

We prove each part of Theorem 2 in turn, using Lemma 11 to convert between the RKHS and Euclidean formulations.

Proof of part (i). By Lemma 11, the objective equals

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

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 11, 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 10, any optimal feature φ_j lies in the span of the n training feature vectors:

$$\varphi_j(\boldsymbol{x}) = \sum_{\mu=1}^{n} \alpha_{j,\mu} K_{\ell-1}(\boldsymbol{x}, \boldsymbol{x}_\mu). \tag{94}$$

Define $\boldsymbol{\alpha}_j = (\alpha_{j,1}, \ldots, \alpha_{j,n})^\top$ as the coefficient vector for feature j. By Lemma 11, the RKHS norm of this feature is

$$\|\varphi_j\|_{\mathcal{H}_{\ell-1}}^2 = \boldsymbol{\alpha}_j^\top G_{\ell-1} \boldsymbol{\alpha}_j, \tag{95}$$

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 2 (ii) becomes

$$\max_{\boldsymbol{\alpha}:\|\varphi\|_{\mathcal{H}_{\ell-1}}=1} \left|\frac{1}{n}\sum_{\mu=1}^{n} y_\mu \varphi(\boldsymbol{x}_\mu)^2\right| = \max_{\boldsymbol{\alpha}:\boldsymbol{\alpha}^\top G_{\ell-1}\boldsymbol{\alpha}=1} \left|\boldsymbol{\alpha}^\top \widehat{\boldsymbol{C}}^{(\ell)} \boldsymbol{\alpha}\right|, \tag{96}$$

where $\widehat{\boldsymbol{C}}^{(\ell)} = \frac{1}{n}\sum_{\mu=1}^{n} y_\mu K_{\ell-1}(\boldsymbol{x}_\mu, \cdot) \otimes K_{\ell-1}(\boldsymbol{x}_\mu, \cdot)$ is the label-weighted kernel outer-product operator.

Proposition 12 (Generalized Eigenvector Problem) *The optimal dual coefficients $\{\hat{\boldsymbol{\alpha}}_j\}_{j=1}^{k_\ell}$ that sequentially maximize the above constrained objective satisfy the generalized eigenvector problem*

$$G_{\ell-1}^{1/2}\, Y\, G_{\ell-1}^{1/2}\, \boldsymbol{\alpha}_j = \lambda_j\, \boldsymbol{\alpha}_j, \quad j = 1, \ldots, k_\ell, \tag{97}$$

where $Y = \mathrm{diag}(y_1, \ldots, y_n)$ is the label matrix, and λ_j are the generalized eigenvalues ordered by magnitude. The solutions $\{\hat{\boldsymbol{\alpha}}_j\}_{j=1}^{k_\ell}$ recover the dual coefficients of the second-order features, enabling a kernel implementation of Neural LoFi without 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

$$\|\boldsymbol{g}_r^{(p)} - \boldsymbol{g}_r^{\infty}\|_{L^2(\rho_{\boldsymbol{x}})} \xrightarrow[p\to\infty]{P} 0 \tag{98}$$

and, for the training inputs,

$$\|\boldsymbol{G}_r^{(p)} - \boldsymbol{G}_r^{\infty}\|_{\mathrm{op}} \xrightarrow[p\to\infty]{P} 0, \qquad \sum_{\mu=1}^{n} \|K_r^{(p)}(\cdot, \boldsymbol{x}_\mu) - K_r^{\infty}(\cdot, \boldsymbol{x}_\mu)\|_{L^2(\rho_{\boldsymbol{x}})}^2 \xrightarrow[p\to\infty]{P} 0, \tag{99}$$

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 (99). 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 $\boldsymbol{x}_\mu$,

$$\mathbb{E}_{\boldsymbol{a}}\big\|K_r^{(p)}(\cdot, \boldsymbol{x}_\mu) - \mathcal{K}_r[\boldsymbol{g}_r^{(p)}](\cdot, \boldsymbol{x}_\mu)\big\|_{L^2(\rho_{\boldsymbol{x}})}^2 \le \frac{C_\mu}{p_r}, \tag{100}$$

and the same iid law of large numbers applies to the finitely many Gram entries. Thus (99) 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 $K_p = K^{(p)}_{\ell-1}$, $K_\infty = K^{\infty}_{\ell-1}$, and $\boldsymbol{G}_p, \boldsymbol{G}_\infty$ for their Gram matrices. By (99),

$$\|\boldsymbol{G}_p - \boldsymbol{G}_\infty\|_{\mathrm{op}} \xrightarrow[p\to\infty]{P} 0, \qquad \sum_{\mu=1}^{n} \|K_p(\cdot, \boldsymbol{x}_\mu) - K_\infty(\cdot, \boldsymbol{x}_\mu)\|^2_{L^2(\rho_{\boldsymbol{x}})} \xrightarrow[p\to\infty]{P} 0. \tag{101}$$

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, (100) gives the sharper Hilbert-space law of large numbers needed for out-of-sample features.

By Lemma 10, 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

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

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

$$\boldsymbol{B}_p \boldsymbol{\beta} = \lambda \boldsymbol{\beta}, \qquad \boldsymbol{B}_p = \frac{1}{n}\boldsymbol{G}_p^{1/2} Y \boldsymbol{G}_p^{1/2}, \qquad \boldsymbol{\beta} = \boldsymbol{G}_p^{1/2}\boldsymbol{\alpha}. \tag{102}$$

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 [29] 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,

$$\boldsymbol{\beta}_{p,k} \xrightarrow[p\to\infty]{P} \boldsymbol{\beta}_{\infty,k}. \tag{103}$$

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 (103) give $\boldsymbol{\alpha}_{p,k} \to \boldsymbol{\alpha}_{\infty,k}$ in probability. Hence, with

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

we have

$$\begin{aligned} \|\widehat{\varphi}_{p,k} - \phi^\infty_k\|_{L^2(\rho_{\boldsymbol{x}})} &\le \|\boldsymbol{k}_p - \boldsymbol{k}_\infty\|_{L^2(\rho_{\boldsymbol{x}};\mathbb{R}^n)} \|\boldsymbol{\alpha}_{p,k}\|_2 \\ &\quad + \|\boldsymbol{k}_\infty\|_{L^2(\rho_{\boldsymbol{x}};\mathbb{R}^n)} \|\boldsymbol{\alpha}_{p,k} - \boldsymbol{\alpha}_{\infty,k}\|_2 \xrightarrow[p\to\infty]{P} 0. \end{aligned} \tag{104}$$

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 $\boldsymbol{g}_\ell^{(p)} = (\psi_p^\ell, \widehat{\varphi}_{p,1}, \ldots, \widehat{\varphi}_{p,k_\ell})$ 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.

□

Appendix H. 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 [6, 72, 80, 92]. We comeback in this section on the discussion of the emergence of learned features at each layer, and in particular, we now make the noise level $\tau_\ell^k(n)$ in (20) 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 13 (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*

$$D_k^\ell(r) := \mathrm{Tr}\left[T_{\ell,k}(T_{\ell,k} + rI)^{-1}\right] = \sum_{j\geq 1} \frac{\lambda_j^{\ell,k}}{\lambda_j^{\ell,k} + r}, \tag{105}$$

The above notion of effective dimension is standard in kernel regression [21, 76], and similar quantities appear in local Rademacher analyses of kernel classes [10, 58]. $D_k^\ell(r)$ can be interpreted as counting the number of residual directions whose variance is above the resolution r.

Assumption 14 (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 ([57]).

Theorem 15 (Emergence noise floor) *Suppose that Assumption 14 holds and the kernel $K_{\ell-1}$ and labels y are uniformly bounded. Let*

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

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 $1-\delta$,

$$\tau_\ell^k(n) := \sup_{\varphi\in\mathcal{S}_k^\ell} \left|\widehat{c}_{\ell,n}(\varphi) - c_\ell(\varphi)\right| \leq \widetilde{C}_{\ell,H}\, r_\ell^{k,\star}\sqrt{\frac{D_k^\ell(r_\ell^{k,\star})}{n}}. \tag{106}$$

This gives, up to polylogarithmic factors, the sample scale

$$n \gtrsim \frac{\widetilde{C}_{\ell,H}^2 (r_\ell^{(k)})^2 D_k^\ell(r_\ell^{(k)})}{(\rho_\ell^{(k)})^2}.$$

Proof

For $r > 0$, let

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

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

$$\tau_\ell^k(r,n) := \sup_{\varphi\in\mathcal{S}_k^\ell(r)} \left|\hat{\mathbb{E}}[y\varphi^2] - \mathbb{E}[y\varphi^2]\right|. \tag{107}$$

We will show that:

$$\tau_\ell^k(r,n) \lesssim \widetilde{C}_{\ell,H}\, r\sqrt{\frac{D_k^\ell(r)}{n}} \tag{108}$$

The above bound follows through an extension of Theorem 2.1 in [58] which bounds the rademacher complexity of $S_k^\ell(r)$. Concretely, Theorem 2.1 in [58] provies that for a unit ball of an RKHS of a bounded kernel with eigenvalues $\{\lambda_j\}_{j\geq 1}$:

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

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 13.

We now adapt the proof of Theorem 2.1 of [58] to the quadratic process. By the boundedness of the labels y, $\tau_\ell^k(r,n)$ can be bounded upto constants by the rademacher complexity of φ^2 over $S_k^\ell(r)$, defined as:

$$\mathbb{E}_\epsilon[\sup_{\varphi\in\mathcal{S}_k^\ell(r)}\left|\sum_{i=1}^n \epsilon_i\varphi(\boldsymbol{X}_i)^2\right|], \tag{109}$$

where $\epsilon \in \{\pm 1\}$ 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 [58] 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

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

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:

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

$$\sup_{\beta\in I_r}\left|\sum_{i=1}^n \epsilon_i\varphi_\beta(\boldsymbol{X}_i)^2\right| \lesssim \sup_{\|\theta\|_2\le 1}\left|\sum_{i=1}^n \epsilon_i\left(\sum_j \theta_j\sqrt{a_j(r)}\phi_j^{\ell,k}(\boldsymbol{X}_i)\right)^2\right|. \tag{110}$$

Define:

$$v_{\ell,k,r}(\boldsymbol{x}) := \left(\sqrt{a_j(r)}\,\phi_j^{\ell,k}(\boldsymbol{x})\right)_{j\ge 1},$$

then the RHS in Equation 110 can be expressed as:

$$\left\|\sum_{i=1}^n \epsilon_i v_{\ell,k,r}(\boldsymbol{X}_i)\otimes v_{\ell,k,r}(\boldsymbol{X}_i)\right\|_{\mathrm{op}}. \tag{111}$$

Let

$$\psi_{\ell,k}(r)^2 := \sum_j a_j(r) \asymp rD_k^\ell(r).$$

We now bound Equation 111 through a matrix concentration argument which requires bounding the operator norm of the following fourth-moment matrix:

$$M_{\ell,k}(r) := \mathbb{E}\left[\left\|v_{\ell,k,r}(\boldsymbol{X})\right\|_2^2 v_{\ell,k,r}(\boldsymbol{X})\otimes v_{\ell,k,r}(\boldsymbol{X})\right].$$

We bound $\left\|M_{\ell,k}(r)\right\|$ through Assumption 14. 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 14, we obtain:

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

Therefore

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

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 (108). Taking the maximum over shells gives the stated bound with $r_\ell^{k,\star}$. ■

H.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:

$$\boldsymbol{C}^{(\ell)} \coloneqq \mathbb{E}[y\, \boldsymbol{z}_{\ell-1}(\boldsymbol{x})\boldsymbol{z}_{\ell-1}(\boldsymbol{x})^\top], \tag{112}$$

and the unweighted covariance:

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

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:

$$\boldsymbol{\Sigma}_k^{(\ell)} \coloneqq (I - \boldsymbol{v}_{k-1}\boldsymbol{v}_{k-1}^\top)\boldsymbol{\Sigma}_{k-1}^{(\ell)}(I - \boldsymbol{v}_{k-1}\boldsymbol{v}_{k-1}^\top). \tag{114}$$

(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:

$$D_\ell^k(r) \coloneqq \sum_{j\geq 1} \frac{\lambda_j^{\ell,k}}{\lambda_j^{\ell,k} + r}. \tag{115}$$

(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:

$$n_\ell^k \sim \left(\frac{r_\ell^k}{\rho_\ell^{(k)}}\right)^2 D_\ell^k, \tag{116}$$

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

H.2. Effective dimension in the Hermite toy model

The abstract quantity $D_k^\ell(r)$ becomes concrete in the toy model Appendix F. 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 F.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 F,

$$h_i^{(1)}(\boldsymbol{x}) = \langle A_i^{(1)}, H_k(\boldsymbol{x})\rangle, \qquad i = 1,\dots,d_1,$$

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:

$$\mathbb{E}[y(h_i^{(1)}(\boldsymbol{x}))^2] \sim \frac{1}{\sqrt{d_1}} \tag{117}$$

Substituting the above scaling into the sample complexity prediction prescribed by Theorem 15 gives the $d^{k+\epsilon}$ first-stage sample scale stated in Appendix F.

H.3. Effective dimension in the multi-index models

We now discuss how the sample complexity prediction in Proposition 15 recovers the rates in the setting of multi-index models with power-law dependence on the label.

We follow here [30, 31] 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 15. The features at layer ℓ along each candidate direction have unit variance under the Gaussian input, so the variance scale is order one, $\tau = r_\ell^{(k)} = O(1)$. 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 $\rho_\ell^{(i)} = O(i^{-\gamma})$, where $\gamma > 0$ is the exponent governing

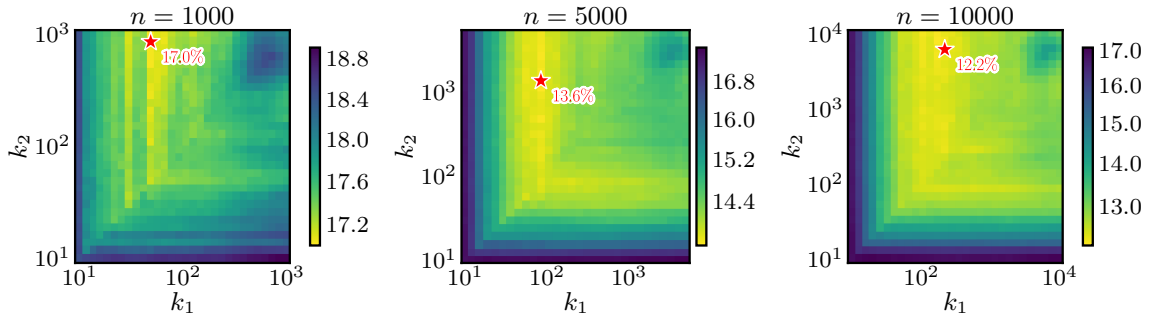

Figure 6: Test Error (%) as a function of the kept features for Kernel LoFi for different training dataset sizes n. Stars indicate the best (k_1, k_2) in each grid. The optimal retained dimension is finite and different from the usual kernel case (upper right corner), mirroring the behavior of the finite-width Neural LoFi heatmaps in Figure 1

the label spectrum. Plugging $D_k^\ell \asymp d$, $\tau = O(1)$ and $\rho^{(i)} = O(i^{-\gamma})$ into the recovery condition $n \gg (r_\ell^{(k)})^2 D_k^\ell / (\rho_\ell^{(k)})^2$ of Proposition 15 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 [30, 31].

Appendix I. Additional Numerical Explorations

I.1. Neural-Lofi Kernel Limit

The kernel formulation of Neural LoFi developed in Appendix G replaces the random projections $\boldsymbol{R}_\ell$ of the spectral filter at layer ℓ by solving the generalized eigenvalue Algorithm 1 by their kernel expectations and computes problem of Proposition 12 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 6 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 1: 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 2 and the emergence threshold of eq. (20): 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 [41, 48], while the lottery-ticket hypothesis suggests that large dense networks may contain smaller trainable subnetworks [35]. 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.

I.2. Measuring the feature alignment of LoFi with GD

In order to establish if the features learned by LoFi are a good approximation of those learned by GD, we aim to measure how the alignment grows during GD training, when compared with the features learned by LoFi.

Let's consider a setting where we train the same architecture with both GD and LoFi, and we measure the overlap between the features at each layer. More concretely, let $\boldsymbol{z}_\ell^{\text{LoFi}}(x_\mu)$ denote the features at layer ℓ learned by LoFi, and let $\boldsymbol{z}_\ell^{\text{GD}}(x_\mu, t)$ denote the features at the same layer learned by GD at training step t; both are vector representations of the same dimension p_ℓ. We can then compute the *featurewise overlap* as

$$\big[F_\ell(t)\big]_{ij} = \frac{1}{n}\sum_{\mu=1}^{n} \frac{\left(\boldsymbol{z}_{\ell,i}^{\text{LoFi}}(x_\mu) - \left\langle \boldsymbol{z}_{\ell,i}^{\text{LoFi}} \right\rangle\right)\left(\boldsymbol{z}_{\ell,j}^{\text{GD}}(x_\mu, t) - \left\langle \boldsymbol{z}_{\ell,j}^{\text{GD}}(t) \right\rangle\right)}{\sqrt{\left(\boldsymbol{z}_{\ell,i}^{\text{LoFi}}(x_\mu) - \left\langle \boldsymbol{z}_{\ell,i}^{\text{LoFi}} \right\rangle\right)^2}\sqrt{\left(\boldsymbol{z}_{\ell,j}^{\text{GD}}(x_\mu, t) - \left\langle \boldsymbol{z}_{\ell,j}^{\text{GD}}(t) \right\rangle\right)^2}}, \tag{118}$$

where $\left\langle \boldsymbol{z}_{\ell,i}^{\text{LoFi}} \right\rangle$ and $\left\langle \boldsymbol{z}_{\ell,j}^{\text{GD}}(t) \right\rangle$ denote the mean of the respective features across the dataset. This matrix $F_\ell(t) \in \mathbb{R}^{p_\ell \times p_\ell}$ captures the pairwise correlations between the features learned by LoFi and GD at layer ℓ and training step t. Feature are permutation invariant, so we consider the *Frobenius norm* of the overlap matrix as a measure of overall alignment, normalized by its initial value at random initialization:

$$\text{Normalized Overlap}_\ell(t) = \frac{\|F_\ell(t)\|_F - \|F_\ell(0)\|_F}{\|F_\ell(0)\|_F}. \tag{119}$$

The normalization is required because having every layer a different p_ℓ and different spreadness of the features, the initial overlap at random initialization can be different across layers, and we want to measure the relative growth of the alignment during training.

In Figure 2, we plot the evolution of $\text{Alignment}_\ell(t)$ for each layer ℓ during GD training, for a 4 layer fully-connected network trained on a binary classification task on CIFAR-10.

I.3. Generalization and Spectral Performance: GD vs. LoFi

In tis 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

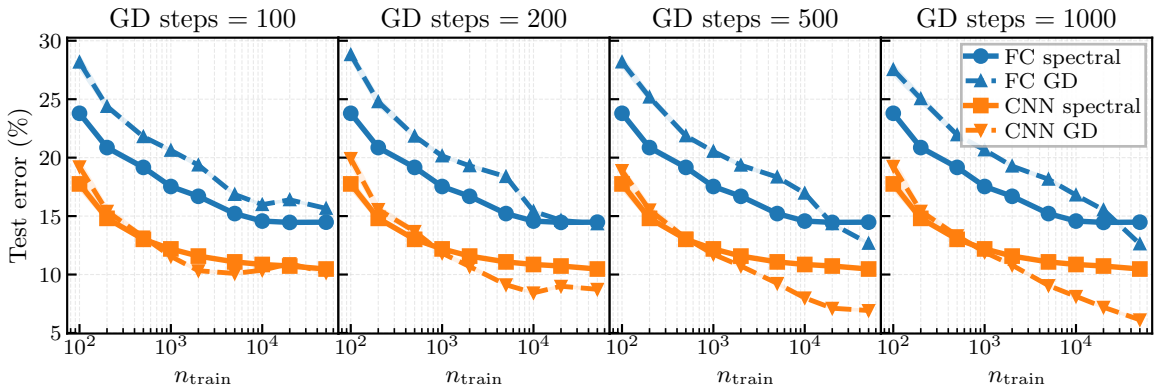

Figure 7: Test error comparison between GD and LoFi on binary CIFAR-10 (Animals vs. Vehicles). GD curves are shown for varying total gradient steps. In the low-data regime, LoFi matches or exceeds the performance of GD, confirming its efficiency as a supervised feature extractor.

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.

Figure 7 illustrates the test error as a function of the training set size. We observe that LoFi achieves generalization performance comparable to, and in low-sample regimes occasionally superior to, GD. By reporting GD performance at varying step intervals, we demonstrate that LoFi, despite being a one-pass spectral method, captures a feature set equivalent to that learned by GD during its early-to-mid training stages.

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. Crucially, these results are obtained without auxiliary regularization (e.g., dropout, early stopping) or optimized filtering levels for LoFi. Thus, the results in Figure 7 suggest that the spectral alignment of LoFi is not merely a theoretical property but a robust driver of generalization in deep architectures.

I.4. Filtering of Features

Figure 8 shows the test error as a function of k for a single hidden layer, where only the intermediate representation is subject to Neural LoFi's feature selection. The curve structure mirrors the two-dimensional heatmaps of Figure 1, now collapsed to a single axis. As p increases the resulting test error improve and the optimal number of retained features $k^\star$ increases. Crucially, selecting too

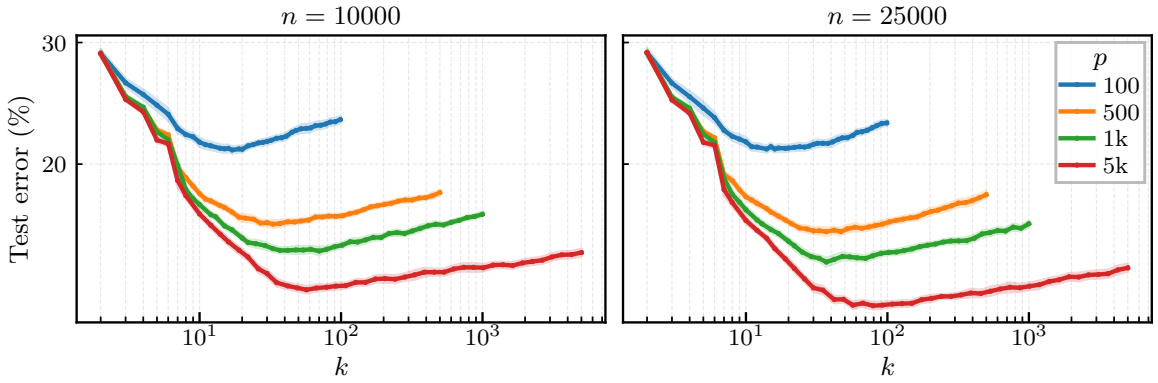

Figure 8: Test error (%) as a function of the number of retained features k at the hidden layer, for a one-hidden-layer Neural LoFi with ReLU activation on CIFAR-10, for varying number of random projections p and two training set sizes. The input and output layers are not reduced. The optimal k increases with p, and larger p consistently yields lower test error.

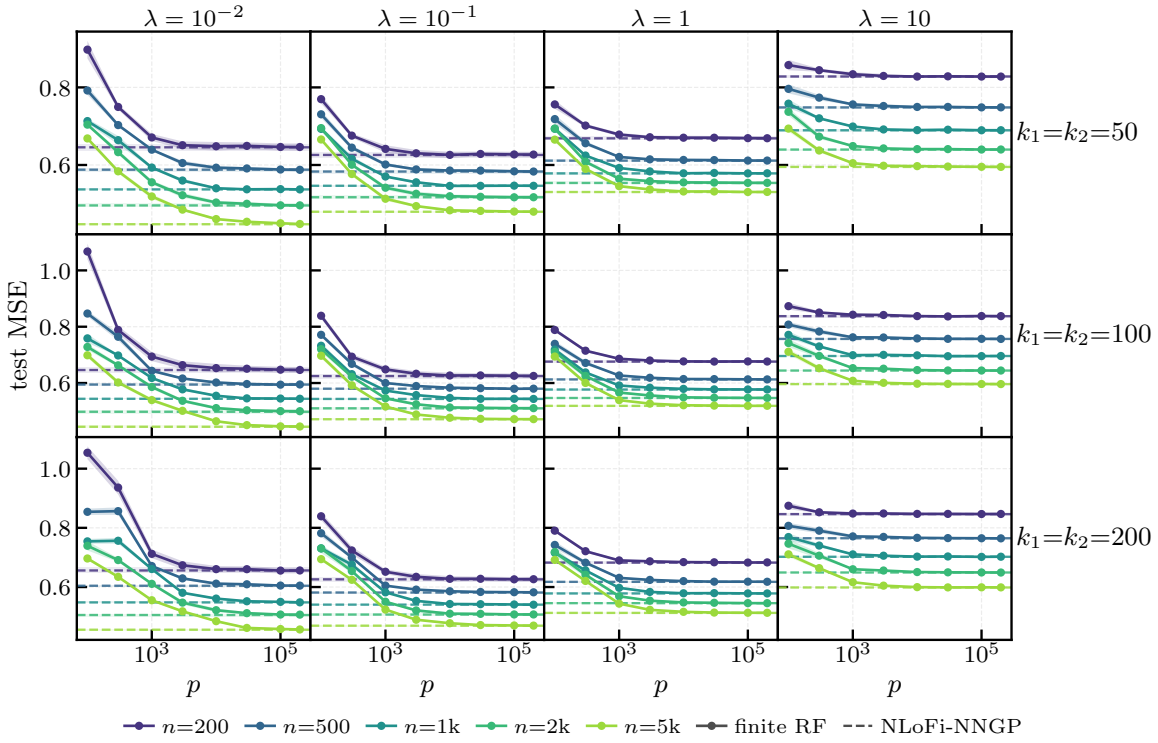

Figure 9: Convergence of finite-width NLoFi to the NLoFi-NNGP limit as the hidden width p grows. Each panel shows the test MSE versus p for four training-set sizes n. Solid curves are finite-RF NLoFi; dashed horizontal lines mark the corresponding NLoFi-NNGP limits at the same n. Rows vary the number 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.

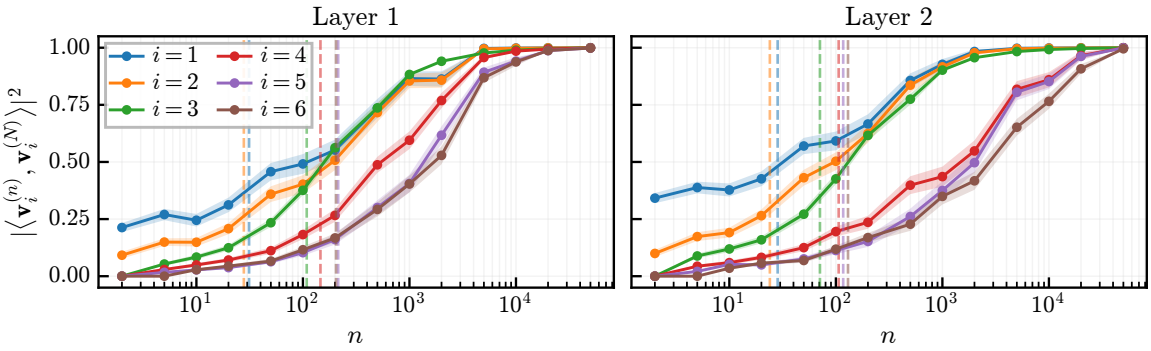

Figure 10: **Predicting when individual features emerge on CIFAR-10.**: We track the recovery of the signed-covariance eigenbasis for a three-layer fully connected Neural LoFi model on the CIFAR-10 animal-vs.-vehicle task. For each layer $\ell \in \{1, 2\}$ and eigenvector index $i = 1, \ldots, 6$, we plot the squared overlap $|\langle v_i^{(n)}, v_i^{(N)} \rangle|^2$ between the eigenvector estimated from n samples and a large-sample reference eigenvector computed with $N = 60{,}000$ samples. Curves show the mean $\pm$ SEM over 100 random subsamples at fixed random features (W_1, W_2). Dashed vertical lines indicate the predicted emergence thresholds n_ℓ^i from Eq. (116). The sharp rise of each overlap occurs near its predicted threshold, showing that the effective-dimension criterion predicts not only aggregate performance but the sample scale at which individual task-relevant directions become learnable.

few or too many features degrades performance, confirming that the U-shaped trade-off observed in Figure 1.

I.5. Predicting the sample complexity for feature recovery

The variational analysis of Section A.2.1 predicts that the k-th eigenvector of the layer-ℓ signed covariance becomes recoverable when the population correlation $\rho_\ell^{(k)}$ exceeds the empirical noise 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 and abstract statement; whether the corresponding *quantitative* prediction Eq. (116) has any predictive content on real images is not obvious. The bound underlying it relies on assumptions that are not satisfied by CIFAR-10 features by design. The experiment in Figure 10 is intended as a stress test of the prediction in this 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. The predicted thresholds n_ℓ^k are obtained from the proxy at $n = N$ by the recursive procedure of App. H, 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 six leading eigenvectors, which themselves are sharp on a logarithmic scale and proceed in order of decreasing $|\lambda_k|$, as in the BBPEA picture of Section A.2.1.

The bound underlying Eq. (116) 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, the agreement is enough for Eq. (116) to serve as a diagnostic: a single eigendecomposition on a reference set indicates the order of magnitude at which each eigendirection becomes extractable.

I.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.

I.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,

$$I_k^{(\ell)}(d) \;=\; \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\partial}{\partial x_{i,d}}\Big(v_k^{(\ell)\top}\phi_\ell(x_i)\Big)\right| \;=\; \frac{1}{N}\sum_{i=1}^{N}\Big|J_\ell(x_i)^\top v_k^{(\ell)}\Big|_d\,, \tag{120}$$

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 (120) reduces to $I_k^{(0)}(d) = \big(v_k^{(0)}\big)_d^2$.

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,

$$\bar{I}^{(\ell)}(d) \;=\; \frac{\sum_{k=0}^{K_\ell-1}|\lambda_k^{(\ell)}|\, I_k^{(\ell)}(d)}{\sum_{k=0}^{K_\ell-1}|\lambda_k^{(\ell)}|}. \tag{121}$$

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.

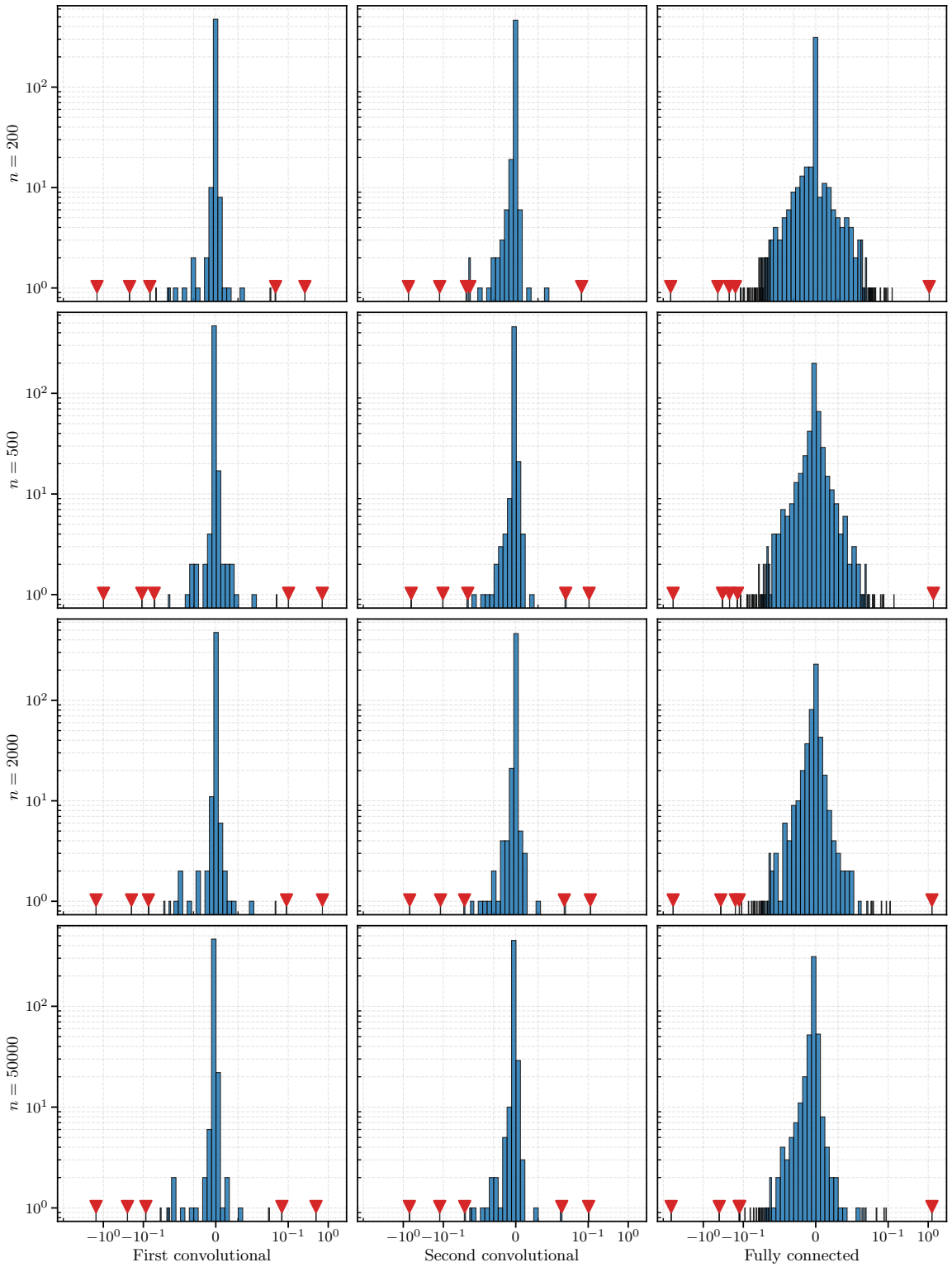

Figure 11: **Spectral Distribution:** Histograms of eigenvalues across network layers (columns) and dataset sizes (rows). Red markers indicate the five most dominant eigenvalues. The symlog scale reveals the emergence of spectral structure and the separation of lead features from the bulk distribution as n grows. The random feature dimesnion in this experiment is $p = 512$ for all layers.

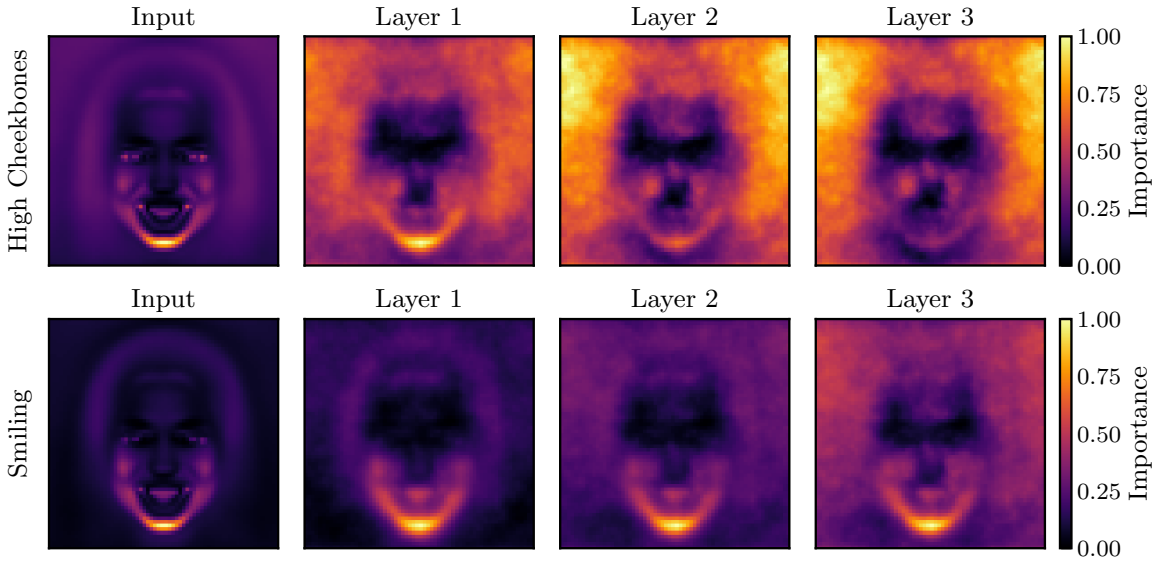

Figure 12: Layer-wise feature importance $\bar{I}^{(\ell)}$ (eq. (121)) on CelebA [52] for two binary attributes. *(Top, High Cheekbones)* Importance concentrates progressively on the cheekbone region across layers, while the chin area, salient at the input, is progressively suppressed in deeper layers. *(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.

Filter and activations for CNNs In Section 3 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 to the filters learned on CIFAR-10 (Figure 3) 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 filter of brightness, and converge 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.

Appendix J. Code Implementation

All experiments are carried out with a self-contained Python package built on PyTorch [67], Numpy [42], Scipy [90] and scikit-learn [68]. The source code is available in the attached zip file.

For each layer $\ell = 1, \dots, L$, the trainer (i) estimates the RMS norm $c_\ell = \big(\frac{1}{n}\sum_i \|\mathbf{h}_{\ell-1}^{(i)}\|^2\big)^{1/2}$ 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

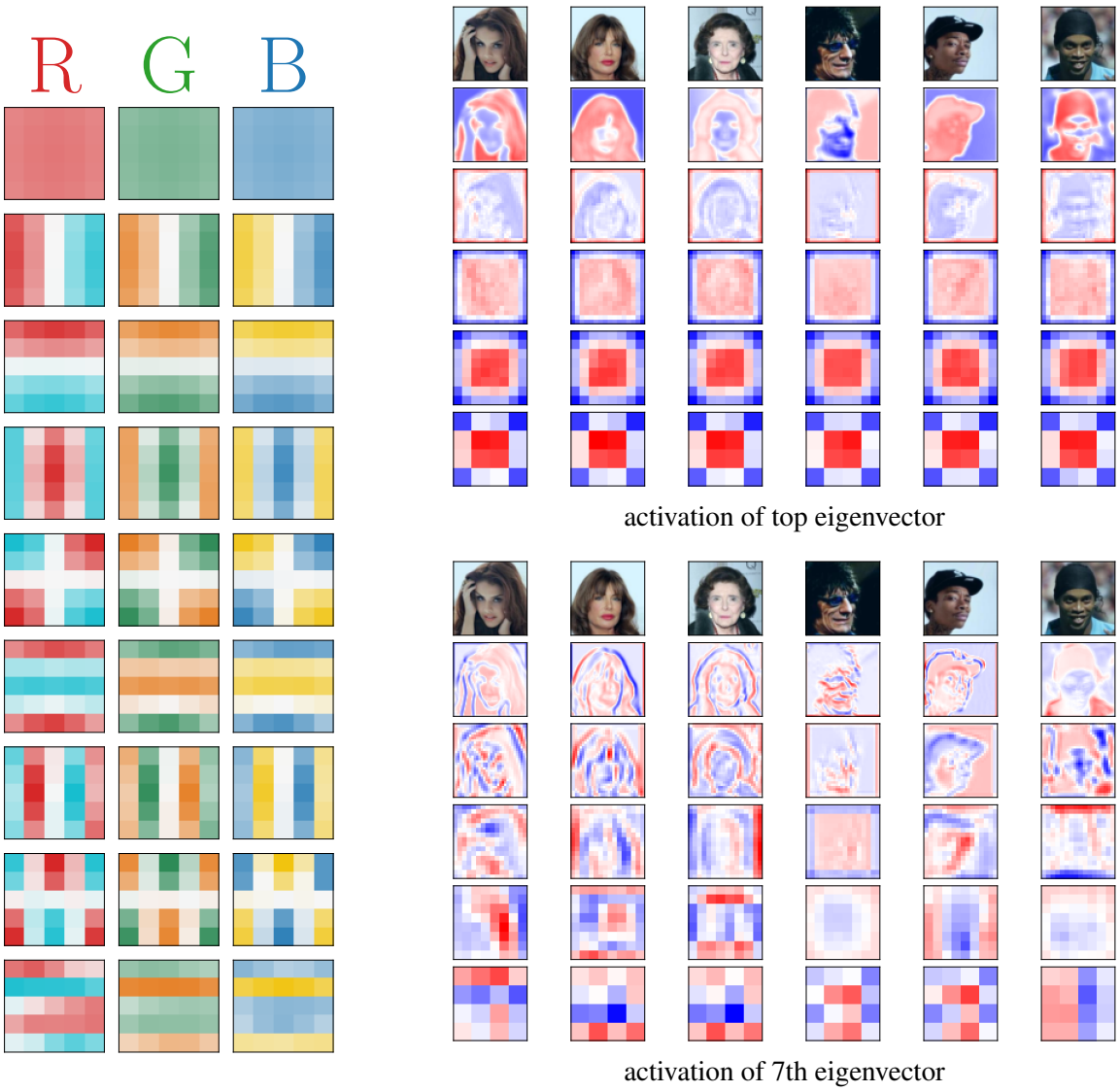

Figure 13: **Filters and activations for CNNs.** We train a 6 convolutional + 1 fully connected layer neuural network on CelebA [52] for binary classification of the "Gender" attribute, using Neural LoFi with signed-covariance eigendecomposition and eigenvalue-based feature selection. *(Left)* The 5×5 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 (th lower the deeper), for 6 images of the test set.

`torch.linalg.eigh`; for moderate p_ℓ on CPU, the full LAPACK eigensolver is used when $k_\ell \geq p_\ell/4$ and Lanczos (`scipy.sparse.linalg.eigsh`) otherwise; for $p_\ell > 50{,}000$ and $k_\ell \leq p_\ell/2$, 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 bythresholding at zero (targets ± 1).

J.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 1, 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 [68] in $[10^{-6}, 10^6]$ with 500 points; 10 seeds.

Figure 1, 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 2. Four-layer FC ReLU net on binary CIFAR-10; width $p = 1000$, LoFi ranks $k = 25$, $n = 50000$. SGD: batch size 512, lr 0.06 with cosine + warmup, 98 steps.

Figure 3, left. 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 3, right. Same architecture. Activation maps of the 6th LoFi feature on test images at successive depths.

Figure 4, left, center. Hierarchical teacher of App. F 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. F.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 4, 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. F.2. The displayed spectrum is computed with a randomized eigensolver. One seed.

Figure 5 Same hierarchical teacher as in Figure 4, 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. F.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 6 Kernel LoFi (App. G) 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 7. Binary CIFAR-10 animal vehicle, $n \in [10^2, 5 \times 10^4]$. Matched architectures with $p = 4096$, The LoFi ranks are kept fixed between architectures $32, 64, 128, 256$. SGD: batch $B = 512$, $\eta \propto \sqrt{B/n}$ (peak 0.01), cosine + warmup, total steps fixed across n. 10 seeds.

Figure 8. Binary CIFAR-10, single LoFi hidden layer (ReLU), only the hidden representation reduced to rank k. $p \in \{100, 500, 1000, 5000\}$, $n \in \{10000, 25000\}$, λ optimally tuned with `RidgeCV` from [68] in $[10^{-6}, 10^6]$ with 500 points. Shading: ± 1 std over 5 seeds.

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 10 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. (116).

Figure 11. Binary CIFAR-10. Two conv layers (kernel 3×3, $p = 512$, $2{\times}2$ 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 $64{\times}64$, 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. (121); 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 ($5{\times}5$, $2{\times}2$ pool, L_2 norm) + FC layer; channels $p = 512$, ranks 16, 32,64,128,256,512 , ReLU; $n = 500000$. First-layer $5{\times}5$ 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.