Feature learning is decoupled from generalization in neural networks

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2025

Abstract

Neural networks outperform kernel methods, sometimes by orders of magnitude, e.g. on staircase functions. This advantage stems from the ability of neural networks to learn features, adapting their hidden representations to better capture the data. We introduce a concept we call feature quality to measure this performance improvement. We examine existing theories of feature learning and demonstrate empirically that they primarily assess the strength of feature learning, rather than the quality of the learned features themselves. Consequently, current theories of feature learning do not provide a sufficient foundation for developing theories of neural network generalization.

1. Introduction

Neural networks (NNs) generalize remarkably well in diverse domains, from computer vision to natural language processing or to protein folding [12, 33, 37]. However, understanding the mechanisms behind this success remains a fundamental challenge in machine learning. A leading hypothesis attributes this success to feature learning (FL) - the network's ability to adapt its hidden representations to discover useful patterns from data. For instance, visualization techniques reveal that convolutional NNs naturally develop hierarchical feature representations, progressively learning to detect edges, textures, patterns, object parts, and finally complete objects [49]. When comparing NNs to their linearized approximations [32], NNs achieve dramatically better sample complexity on many tasks (see discussion in Section 2.1). This suggests that NNs learn better features through training than those present at initialization. Current literature characterizes FL as a change in the Neural Tangent Kernel (NTK, [32]), Conjugate Kernel (CK, [38]), or via some weight-based metric. These approaches measure FL by quantifying how much a trained network deviates from its linear approximation at initialization. While there is a general notion that FL improves generalization, we argue this relationship is misleading: FL theories measure FL strength — the magnitude of representation change — which is fundamentally decoupled from feature quality — the actual impact on generalization. Our **contributions** are: *(i)* We define a rigorous way to measure feature quality through the FL gap Δ_{NT}. *(ii)* We demonstrate that current FL definitions measure strength rather than quality, and show these are decoupled.

See Appendix B for notation and background on kernel methods and Appendix C for related work.

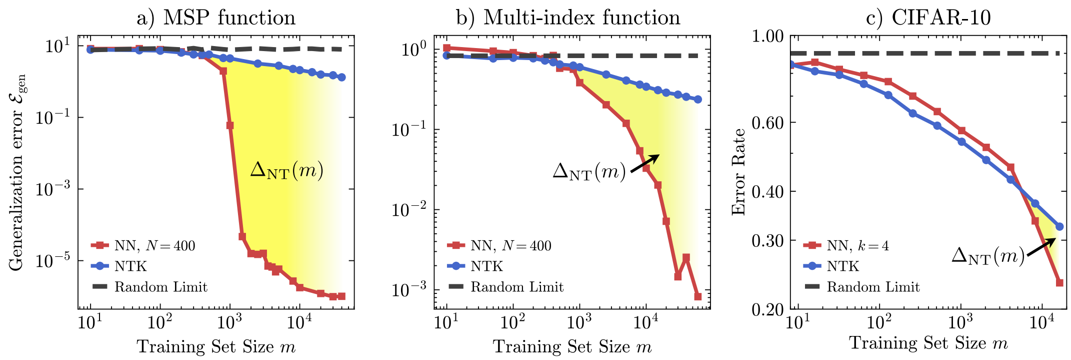

Figure 1: **Generalization error $\mathcal{E}_{\text{gen}}$ versus training set size** m for NNs and their corresponding NTK across three distinct target functions: (a) FFNN on merged staircase (MSP) functions Appendix D.1, (b) FFNN on Multi-index functions Appendix D.2, and (c) Wide ResNet on CIFAR-10. For (a) and (b), we observe a critical training set size m^* where NNs outperfrom their NTK counterparts by orders of magnitude ($m^* \sim 10^3$). We quantify this improvement in performance through the FL gap Δ_{NT} (Definition 1). For (c), the learning curve for the NN scales similarly to the NTK until $\sim 10^4$.

2. Feature Learning vs. Feature Quality

2.1. The feature learning gap

Figure 1 shows that the NTK significantly underperforms the corresponding NN after some amount of data $m^* \sim 10^3$ on the MSP functions and multi-index functions. On CIFAR-10, the NTK achieves comparable performance to the NN until $m^* \sim 10^4$[1]. MSP and multi-index functions are not isolated examples; rather, there exists a large class of target functions (collected in Table 6) where kernels have polynomial or even exponential sample complexity, and NNs achieve linear or lower degree polynomial sample complexity. These dramatic differences in sample complexity suggest that the NNs learn high-quality features not present at initialization (and thus in their CK/NTK). We quantify this feature quality through the FL gap.

Definition 1 (Feature learning gap) *Given a data generating model $p(\boldsymbol{x}, \boldsymbol{y})$, an i.i.d. dataset $\mathcal{D}$ of size m with a target function $f^* : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, a FFNN $f_{\boldsymbol{\theta}}$, and the mean predictor of the FFNN's NTK, given by $\mu_{\text{NT}}(\boldsymbol{x}) = K_{\text{NT}}(\boldsymbol{x}, \boldsymbol{X}) N_{\text{NT}}(\boldsymbol{X}, \boldsymbol{X})^{-1} \boldsymbol{Y}$, the FL gap is defined by*

$$\Delta_{\text{NT}}(m) = \mathcal{E}_{\text{gen}}(\mu_{\text{NT}}; m) - \mathcal{E}_{\text{gen}}(f_{\boldsymbol{\theta}}; m). \tag{1}$$

See Appendix E.8 for further a discussion on properties of Δ_{NT}.

2.2. Disentangling FL strength from feature quality

The literature contains multiple definitions of FL, falling into three main categories: (1) NTK-based, (2) CK-based, and (3) superposition-based definitions. These approaches characterize FL by measuring how hidden representations of $f_{\boldsymbol{\theta}(t)}$ change during training relative to the initialized

1. This is well predicted by how well the empirical NTK aligns with the target function see Appx. Figure 5.

network $f_{\boldsymbol{\theta}(0)}$. Although these measures take different forms based on changes in the NTK, CK, or other metrics, we argue they fundamentally measure *FL strength* rather than *feature quality* (i.e. how useful the features are). This is based on our empirical observations in Section 3 indicating that changes in hidden representations do not guarantee the learning of high-quality features, as quantified by Δ_{NT}. Conversely, an NN can generalize effectively with minimal changes to its representations if the initial kernel already encodes useful features [52].

Claim: Current FL definitions (explicitly or implicitly) characterize FL by measuring FL strength $S(f_{\boldsymbol{\theta}})$. However, FL strength is decoupled from feature quality, measured by the FL gap Δ_{NT}.

3. Current FL definitions do not provide a feature quality measure

Methodology Every FL theory provides a measure of feature strength $S(f_{\boldsymbol{\theta}})$. Our goal in this section is to assess whether the strength measures correlate with the FL gap Δ_{NT}, i.e. whether strong FL implies a beyond-the-kernel performance. We conduct experiments with two architectures and datasets (1) CNNs trained on CIFAR-10 and (2) FFNNs trained on MSP functions[2]. For each setup, we compare models trained on true-labeled data to those trained on shuffled labels. Any non-vacuous generalization bound must be data-dependent [7, 73]. If $S(f_{\boldsymbol{\theta}})$ correlates with feature quality (Δ_{NT}), the feature strength measures must demonstrate a qualitative distinction between NNs trained on shuffled vs. non-shuffled data. Otherwise, the result suggests a lack of correlation between feature strength and quality.

3.1. Family 1: NTK based definitions

The first FL definition we examine is based on the identification of two training regimes, the "lazy" and "rich" regimes [15, 42]. In the lazy regime, NNs behave like their linearized approximations

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_0}(\boldsymbol{x}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_0}(\boldsymbol{x}) + \mathcal{O}(\boldsymbol{\theta}^2). \tag{2}$$

Although multiple FL definitions exist in the literature [19, 28, 34, 42, 55, 64, 66], they are fundamentally related. NNs FL when they diverge from their linearization at initialization eq. (2). This can be measured by the change in the NTK.

Definition 2 (Feature learning (NTK)) *A NN $f_{\boldsymbol{\theta}}$ feature learns if the empirical NTK $\hat{K}_{\mathrm{NT}}$ changes significantly during training $\exists \varepsilon, T > 0 : \ \forall t > T \quad d(\hat{K}_{\mathrm{NT}}(\boldsymbol{\theta}_0), \hat{K}_{\mathrm{NT}}(\boldsymbol{\theta}_t)) > \varepsilon$, where d is some distance metric for kernels.*

Definition 3 (Feature) *The features are the row vectors of the feature map: $\Phi(\boldsymbol{x}; t) = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})|_{\boldsymbol{\theta}_t}$.*

For NTK-based FL definitions, a larger distance between the initial and final NTK correlates with stronger FL. Accordingly, the following definition of FL strength seems natural.

Definition 4 (FL strength (NTK)) *We set $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}}) = 1 - \kappa_{\mathrm{CKA}}(\hat{K}_{\mathrm{NT}}(\boldsymbol{\theta}_0), \hat{K}_{\mathrm{NT}}(\boldsymbol{\theta}_t))$, where κ_{CKA} is the centered-kernel alignment [35] which measures the normalized distance between kernels.*

2. While this represents a regression task, MSP functions map to whole numbers, making label shuffling well-defined, see Appendix D.1.

Critique In Figure 2(a), we train a CNN (see Table 4 for architecture and optimization details) on CIFAR-10, once with shuffled and once with non-shuffled labels. We calculate the FL strength, as defined in Definition 4 using the neural-tangents package [47, 48] and plot it against the training set size alongside the generalization error. $S(f_{\boldsymbol{\theta}})$ does not provide any qualitative difference between randomly shuffled and non-shuffled labels. We extend this analysis to FFNNs trained on MSP functions in Figure 2(b) (see Table 3 for architecture and optimization details). Here, $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}})$ increases for non-shuffled labels around $m = 3000$ samples unlike the randomly shuffled case, hinting at a correlation with Δ_{NT}. However, this difference can be eliminated by introducing the FL strength parameter γ [8, 15]. This parameter scales the network output by $\tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{\gamma} f_{\boldsymbol{\theta}}(\boldsymbol{x})$. Regarding Figure 2(c), for $\gamma = 0.01$, both the FL strength and learning curves become qualitatively indistinguishable between shuffled and non-shuffled labels. This sensitivity to γ suggests that $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}})$ is not a robust predictor of generalization, particularly since the generalization error remains largely unaffected by this scaling. These findings align with recent observations of "misgrokking" [40], where prolonged training leads to degraded generalization despite significant changes in the NTK.

3.2. Family 2: CK based definitions

A second line of work bases features on the CK. While Nam et al. [45], Yang and Hu [70] are focused on the final layer CK, Fischer et al. [23], Naveh and Ringel [46], Seroussi et al. [57] treat CKs of each layer in a Bayesian framework.

Definition 5 (Feature Learning) *An NN undergoes FL if its final layer feature map* $\Phi^{L-1}(\boldsymbol{x}; t)$ *differs from its initialization* $\Phi^{L-1}(\boldsymbol{x}; 0)$ *at any time t for some input* $\boldsymbol{x} \in \mathcal{X}$.

Definition 6 (Feature) *Given an NN, the features are the eigenfunctions* $[e_1(t), ..., e_{n_L}(t)]$ *of the last layer CK,* $K_{\Phi^{L-1}}(\boldsymbol{x}_\mu, \boldsymbol{x}_\nu; t)$ *(see Appendix B.1), ordered by their eigenvalues* λ_k.

Definition 7 (FL strength (CK)) *Given the trained (scalar-valued) network* $f_{\boldsymbol{\theta}}$*, the utility of the k-th feature* $e_k(t)$ *is* $\hat{Q}_k = \langle e_k | f_{\boldsymbol{\theta}} \rangle^2$*. The cumulative utility of the first k features is* $\hat{\Pi}(k) = \sum_{j=1}^{k} \hat{Q}_j$*, with* $0 \leq \hat{\Pi}(k) \leq 1$ *and* $\hat{\Pi}(n_L) = 1$*. We define the FL strength (CK) as* $S_{\mathrm{CK}}(f_{\boldsymbol{\theta}}) = \min_k \{\hat{\Pi}(k) > \varepsilon\}$*, where* $\varepsilon = 0.95$ *is chosen as a sensible threshold.*

When $\hat{\Pi}(k)$ approaches 1 quickly with k, it means that the NN is strongly learning features, akin to neural collapse [51] where only a minimal number of features are used.

Critique Figure 3 (a) demonstrates that for CIFAR-10, the cumulative utility exhibits a qualitatively indistinguishable trend for networks trained on randomly shuffled labels versus true labels. This holds true for FFNNs trained on the MSP function as well, as shown in Figure 3 (b). Furthermore, our analysis of the CK spectra presented in Appendix E.6 and Figures 6 and 9, reveals qualitatively similar patterns between models trained on shuffled and non-shuffled data. These findings align with previous research on neural collapse, which suggests that neural collapse (which is equivalent to strong FL for CK based FL definitions) can occur independently of generalization [25, 30, 36].

3.3. Family 3: Superposition based definitions

Elhage et al. [21] define features as "*properties of the input which a sufficiently large NN will reliably dedicate a neuron to representing*". While this definition needs further elaboration as we do not know when an NN is "sufficiently large", they later give a more practical definition of a feature which is closely related to family 2.

Definition 8 (Feature) *Given a FFNN with a feature map $\Phi^k(t)$ as defined in Definition 12, a feature f_i corresponds to a direction $\boldsymbol{v}_i \in \mathbb{R}^{n_k}$ in the hidden (activation) space.*

Features correspond to hidden-space vectors $\boldsymbol{v}_i$, whose count can exceed the layer width n_k. This non-orthogonal "superposition" allows more features than dimensions [6, 31]. Given features with values $x_{f_1}, x_{f_2}, \ldots$, the layer encodes them as $\Phi^k(\boldsymbol{x}_\mu; t) = \sum_{i=1}^{n_k} x_{f_i} \boldsymbol{v}_i$. [21] quantified superposition in autoencoders via *feature* and *sample* dimensionality. There is no canonical way to generalize these measures from an autoencoder architecture to a FFNN in the overparameterized regime. Nevertheless, here we adopt a layer-wise definition:

Definition 9 (FL strength) *For layer k, FL strength is measured through two complementary metrics*

$$D_{f_i} = \frac{||\boldsymbol{W}_i||_2^2}{\sum_j (\hat{\boldsymbol{W}}_i \cdot \boldsymbol{W}_j)^2}, \qquad D_{\boldsymbol{x}_\mu} = \frac{||\Phi^k(\boldsymbol{x}_\mu; t)||_2^2}{\sum_\nu (\hat{\Phi}^k(\boldsymbol{x}_\mu; t) \cdot \Phi^k(\boldsymbol{x}_\nu; t))^2}, \tag{3}$$

feature dimensionality D_{f_i} for feature f_i and sample dimensionality $D_{\boldsymbol{x}_\mu}$ for input $\boldsymbol{x}_\mu$, where "ˆ" denotes normalized vectors.

Feature dimensionality measures how much of a hidden dimension is 'dedicated' to representing a specific feature, ranging from 0 to 1. A feature with dimensionality 1 has its own dedicated dimension, while a feature with dimensionality closer to 0 is either not learned at all or shares its representation space with other features, existing in superposition. The same applies for sample dimensionality, but in terms of $\Phi^k(\boldsymbol{x}_\mu)$ rather than $\boldsymbol{W}_i$.

Critique Again, we train a FFNN on MSP functions with shuffled and non-shuffled labels. For a full plot of the sample dimensionality and the feature dimensionality depending on m, see appx. Figure 10. Notably, the sample dimensionality metric fails to distinguish between shuffled and non-shuffled data. However, examining the feature dimensionality reveals more nuanced patterns. In Figure 4 we show the histogram for the feature dimensions for $m = 100$ and $m = 20000$. Feature dimensionality ceases to be a useful measure beyond toy autoencoders, because in general, one does not know the relevant features of the input dimension. Thus one cannot predict from the histogram whether good features were learned. In principle, an optimal feature representation should exhibit a specific pattern: most dimensions should have values close to zero, with a select few non-zero dimensions corresponding to the important directions in hidden space. This pattern is indeed observed in layer 1 for non-shuffled data but is absent in the shuffled one. For deeper layers, this relationship dissolves. By introducing γ in training (as done in the experiment of Figure 2), one can strongly influence the distribution of D_{f_i}, see Figure 11 in the appendix. While D_{f_i} is useful for measuring superposition, it is not a definitive measure of feature quality.

4. Conclusion

We have demonstrated that current theories of FL, while capturing the strength of representation changes during training, fail to predict generalization. This decoupling between FL strength and feature quality suggests the need for a more comprehensive FL theory if FL should act as a foundation for theories of NN generalization.

References

[1] Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. (arXiv:2202.08658), August 2024. URL http://arxiv.org/abs/2202.08658. arXiv:2202.08658.

[2] R. Aiudi, R. Pacelli, P. Baglioni, A. Vezzani, R. Burioni, and P. Rotondo. Local kernel renormalization as a mechanism for feature learning in overparametrized convolutional neural networks. *Nature Communications*, 16(1):568, January 2025. ISSN 2041-1723. doi: 10.1038/s41467-024-55229-3.

[3] Shunta Akiyama and Taiji Suzuki. Excess risk of two-layer relu neural networks in teacher-student settings and its superiority to kernel methods. (arXiv:2205.14818), June 2022. URL http://arxiv.org/abs/2205.14818. arXiv:2205.14818.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? (arXiv:1905.10337), June 2020. URL http://arxiv.org/abs/1905.10337. arXiv:1905.10337.

[5] Zeyuan Allen-Zhu and Yuanzhi Li. Backward feature correction: How deep learning performs deep (hierarchical) learning. (arXiv:2001.04413), July 2023. URL http://arxiv.org/abs/2001.04413. arXiv:2001.04413.

[6] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear Algebraic Structure of Word Senses, with Applications to Polysemy, December 2018. URL http://arxiv.org/abs/1601.03764. arXiv:1601.03764 [cs].

[7] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. (arXiv:1706.05394), July 2017. doi: 10.48550/arXiv.1706.05394. URL http://arxiv.org/abs/1706.05394. arXiv:1706.05394 [stat].

[8] Alexander Atanasov, Alexandru Meterez, James B. Simon, and Cengiz Pehlevan. The optimization landscape of sgd across the feature learning strength. (arXiv:2410.04642), October 2024. doi: 10.48550/arXiv.2410.04642. URL http://arxiv.org/abs/2410.04642. arXiv:2410.04642 [cs].

[9] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *Proceedings of the 37th International Conference on Machine Learning*, page 685–694. PMLR, November 2020. URL https://proceedings.mlr.press/v119/basri20a.html.

[10] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, page 1024–1034. PMLR, November 2020. URL https://proceedings.mlr.press/v119/bordelon20a.html.

[11] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. How feature learning can improve neural scaling laws. (arXiv:2409.17858), September 2024. doi: 10.48550/arXiv.2409.17858. URL http://arxiv.org/abs/2409.17858. arXiv:2409.17858 [stat].

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Abdulkadir Canatar and Cengiz Pehlevan. A kernel analysis of feature learning in deep neural networks. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, page 1–8, Monticello, IL, USA, September 2022. IEEE. ISBN 9798350399981. doi: 10.1109/Allerton49937.2022.9929375. URL https://ieeexplore.ieee.org/document/9929375/.

[14] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23103-1.

[15] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On Lazy Training in Differentiable Programming, January 2020. URL http://arxiv.org/abs/1812.07956. arXiv:1812.07956 [math].

[16] Omry Cohen, Or Malka, and Zohar Ringel. Learning curves for deep neural networks: A gaussian field theory perspective. *Physical Review Research*, 3(2):023034, April 2021. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.3.023034. arXiv:1906.05301 [cs].

[17] Alex Damian, Jason D. Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. (arXiv:2206.15144), June 2022. doi: 10.48550/arXiv.2206.15144. URL http://arxiv.org/abs/2206.15144. arXiv:2206.15144.

[18] Amit Daniely and Eran Malach. Learning parities with neural networks. In *Advances in Neural Information Processing Systems*, volume 33, page 20356–20365. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/eaae5e04a259d09af85c108fe4d7dd0c-Abstract.html.

[19] Clémentine C. J. Dominé, Nicolas Anguita, Alexandra M. Proca, Lukas Braun, Daniel Kunin, Pedro A. M. Mediano, and Andrew M. Saxe. From Lazy to Rich: Exact Learning Dynamics in Deep Linear Networks, September 2024. URL http://arxiv.org/abs/2409.14623. arXiv:2409.14623 [cs].

[20] Konstantin Donhauser, Mingqi Wu, and Fanny Yang. How rotational invariance of common kernels prevents generalization in high dimensions. In *Proceedings of the 38th International Conference on Machine Learning*, page 2804–2814. PMLR, July 2021. URL https://proceedings.mlr.press/v139/donhauser21a.html.

[21] 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, September 2022. URL http://arxiv.org/abs/2209.10652. arXiv:2209.10652 [cs].

[22] Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. (arXiv:2407.05872), July 2024. doi: 10.48550/arXiv.2407.05872. URL http://arxiv.org/abs/2407.05872. arXiv:2407.05872 [cs].

[23] Kirsten Fischer, Javed Lindner, David Dahmen, Zohar Ringel, Michael Krämer, and Moritz Helias. Critical feature learning in deep neural networks. (arXiv:2405.10761), May 2024. doi: 10.48550/arXiv.2405.10761. URL http://arxiv.org/abs/2405.10761. arXiv:2405.10761 [cond-mat].

[24] Spencer Frei, Niladri S Chatterji, and Peter L Bartlett. Random feature amplification: Feature learning and generalization in neural networks.

[25] Tomer Galanti, Liane Galanti, and Ido Ben-Shaul. On the implicit bias towards minimal depth of deep neural networks. (arXiv:2202.09028), September 2022. doi: 10.48550/arXiv.2202.09028. URL http://arxiv.org/abs/2202.09028. arXiv:2202.09028 [cs].

[26] Amnon Geifman, Meirav Galun, David Jacobs, and Ronen Basri. On the spectral bias of convolutional neural tangent and gaussian process kernels. (arXiv:2203.09255), March 2022. doi: 10.48550/arXiv.2203.09255. URL http://arxiv.org/abs/2203.09255. arXiv:2203.09255 [cs].

[27] Amnon Geifman, Daniel Barzilai, Ronen Basri, and Meirav Galun. Controlling the inductive bias of wide neural networks by modifying the kernel's spectrum. (arXiv:2307.14531), March 2024. doi: 10.48550/arXiv.2307.14531. URL http://arxiv.org/abs/2307.14531. arXiv:2307.14531 [cs].

[28] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, November 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc4de. arXiv:1906.08034 [cs].

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

[30] Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. (arXiv:2202.08384), February 2022. doi: 10.48550/arXiv.2202.08384. URL http://arxiv.org/abs/2202.08384. arXiv:2202.08384 [cs].

[31] Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical Models of Computation in Superposition, August 2024. URL http://arxiv.org/abs/2408.05451. arXiv:2408.05451 [cs].

[32] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

[33] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[34] Dhruva Karkada. The lazy (NTK) and rich (μP) regimes: a gentle tutorial, October 2024. URL http://arxiv.org/abs/2404.19719. arXiv:2404.19719 [cs].

[35] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. (arXiv:1905.00414), July 2019. doi: 10.48550/arXiv.1905.00414. URL http://arxiv.org/abs/1905.00414. arXiv:1905.00414 [cs].

[36] Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. (arXiv:2206.04041), April 2023. doi: 10.48550/arXiv.2206.04041. URL http://arxiv.org/abs/2206.04041. arXiv:2206.04041 [cs].

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

[38] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. (arXiv:1711.00165), March 2018. doi: 10.48550/arXiv.1711.00165. URL http://arxiv.org/abs/1711.00165. arXiv:1711.00165 [stat].

[39] Yuanzhi Li, Tengyu Ma, and Hongyang R. Zhang. Learning over-parametrized two-layer relu neural networks beyond ntk. (arXiv:2007.04596), July 2020. doi: 10.48550/arXiv.2007.04596. URL http://arxiv.org/abs/2007.04596. arXiv:2007.04596 [cs].

[40] Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon S. Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. (arXiv:2311.18817), April 2024. doi: 10.48550/arXiv.2311.18817. URL http://arxiv.org/abs/2311.18817. arXiv:2311.18817 [cs].

[41] Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. (arXiv:2103.01210), March 2021. doi: 10.48550/arXiv.2103.01210. URL http://arxiv.org/abs/2103.01210. arXiv:2103.01210 [cs].

[42] Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. In *Advances in Neural Information Processing Systems*, volume 33, page 22182–22193. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/fc2022c89b61c76bbef978f1370660bf-Abstract.html.

[43] Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu. Neural networks efficiently learn low-dimensional representations with sgd. 2023.

[44] Alireza Mousavi-Hosseini, Denny Wu, and Murat A. Erdogdu. Learning multi-index models with neural networks via mean-field langevin dynamics. (arXiv:2408.07254), August 2024. doi: 10.48550/arXiv.2408.07254. URL http://arxiv.org/abs/2408.07254. arXiv:2408.07254 [stat].

[45] Yoonsoo Nam, Chris Mingard, Seok Hyeong Lee, Soufiane Hayou, and Ard Louis. Visualising feature learning in deep neural networks by diagonalizing the forward feature map. (arXiv:2410.04264), October 2024. doi: 10.48550/arXiv.2410.04264. URL http://arxiv.org/abs/2410.04264. arXiv:2410.04264 [stat].

[46] Gadi Naveh and Zohar Ringel. A self consistent theory of gaussian processes captures feature learning effects in finite cnns. In *Advances in Neural Information Processing Systems*, volume 34, page 21352–21364. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b24d21019de5e59da180f1661904f49a-Abstract.html.

[47] Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

[48] Roman Novak, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Fast finite width neural tangent kernel. In *International Conference on Machine Learning*, 2022. URL https://github.com/google/neural-tangents.

[49] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

[50] Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? (arXiv:2106.06770), October 2021. doi: 10.48550/arXiv.2106.06770. URL http://arxiv.org/abs/2106.06770. arXiv:2106.06770 [cs].

[51] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, October 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2015509117. arXiv:2008.08186 [cs].

[52] Leonardo Petrini, Francesco Cagnetta, Eric Vanden-Eijnden, and Matthieu Wyart. Learning sparse features can lead to overfitting in neural networks, October 2022. URL http://arxiv.org/abs/2206.12314. arXiv:2206.12314 [stat].

[53] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, page 5301–5310. PMLR, May 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

[54] Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. (arXiv:2102.11742), June 2021. doi: 10.48550/arXiv.2102.11742. URL http://arxiv.org/abs/2102.11742. arXiv:2102.11742 [cs].

[55] Mariia Seleznova. Analyzing finite neural networks: Can we trust neural tangent kernel theory?

[56] Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. (arXiv:2112.15383), September 2022. doi: 10.48550/arXiv.2112.15383. URL http://arxiv.org/abs/2112.15383. arXiv:2112.15383 [stat].

[57] Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *Nature Communications*, 14(1):908, February 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36361-y.

[58] Zhenmei Shi, Junyi Wei, and Yingyu Liang. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. (arXiv:2206.01717), June 2022. doi: 10.48550/arXiv.2206.01717. URL http://arxiv.org/abs/2206.01717. arXiv:2206.01717 [cs].

[59] Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data v.s. teacher-student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc61d. arXiv:1905.10843 [stat].

[60] Eszter Székely, Lorenzo Bardone, Federica Gerace, and Sebastian Goldt. Learning from higher-order statistics, efficiently: hypothesis tests, random features, and neural networks. (arXiv:2312.14922), October 2024. URL http://arxiv.org/abs/2312.14922. arXiv:2312.14922.

[61] Matus Telgarsky. Feature selection and low test error in shallow low-rotation relu networks.

[62] Umberto M. Tomasini, Antonio Sclocchi, and Matthieu Wyart. Failure and success of the spectral bias prediction for kernel ridge regression: the case of low-dimensional data. (arXiv:2202.03348), February 2022. doi: 10.48550/arXiv.2202.03348. URL http://arxiv.org/abs/2202.03348. arXiv:2202.03348 [cs].

[63] 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. (arXiv:2405.15480), October 2024. doi: 10.48550/arXiv.2405.15480. URL http://arxiv.org/abs/2405.15480. arXiv:2405.15480 [cs].

[64] Zhenfeng Tu, Santiago Aranguri, and Arthur Jacot. Mixed dynamics in linear networks: Unifying the lazy and active regimes. (arXiv:2405.17580), October 2024. doi: 10.48550/arXiv.2405.17580. URL http://arxiv.org/abs/2405.17580. arXiv:2405.17580 [cs].

[65] Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. (arXiv:2206.10012), June 2022. doi: 10.48550/arXiv.2206.10012. URL http://arxiv.org/abs/2206.10012. arXiv:2206.10012 [cs].

[66] Xiang Wang, Chenwei Wu, Jason D. Lee, Tengyu Ma, and Rong Ge. Beyond lazy training for over-parameterized tensor decomposition. (arXiv:2010.11356), October 2020. doi: 10.48550/arXiv.2010.11356. URL http://arxiv.org/abs/2010.11356. arXiv:2010.11356 [stat].

[67] Alexander Wei, Wei Hu, and Jacob Steinhardt. More than a toy: Random matrix models predict how real-world neural representations generalize. In *Proceedings of the 39th International Conference on Machine Learning*, page 23549–23588. PMLR, June 2022. URL https://proceedings.mlr.press/v162/wei22a.html.

[68] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/8744cf92c88433f8cb04a02e6db69a0d-Abstract.html.

[69] Jonathan Wenger, Felix Dangel, and Agustinus Kristiadi. On the disconnect between theory and practice of neural networks: Limits of the ntk perspective. (arXiv:2310.00137), May 2024. doi: 10.48550/arXiv.2310.00137. URL http://arxiv.org/abs/2310.00137. arXiv:2310.00137 [cs].

[70] Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, page 11727–11737. PMLR, July 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

[71] Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. (arXiv:1904.00687), February 2022. URL http://arxiv.org/abs/1904.00687. arXiv:1904.00687.

[72] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. (arXiv:1605.07146), June 2017. doi: 10.48550/arXiv.1605.07146. URL http://arxiv.org/abs/1605.07146. arXiv:1605.07146 [cs].

[73] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. (arXiv:1611.03530), February

2017. doi: 10.48550/arXiv.1611.03530. URL http://arxiv.org/abs/1611.03530. arXiv:1611.03530.

Appendix A. Figures from main text

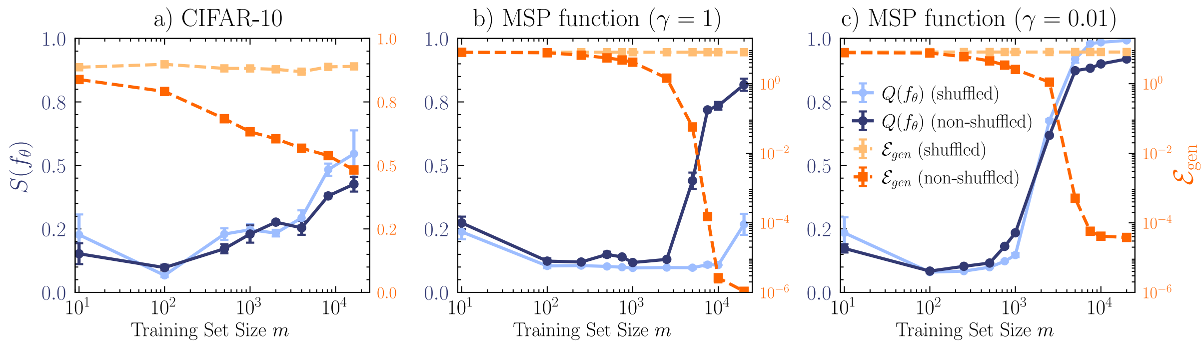

Figure 2: **FL strength $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}})$ is decoupled from generalization error $\mathcal{E}_{\mathbf{gen}}$.** (a) shows a CNN on CIFAR-10 and (b,c) an FFNN on MSP functions with true and shuffled labels. (b) clearly shows significant difference in $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}})$ between the NN and corresponding NTK after $m^* \sim 10^3$, However, this difference vanishes when scaling the network output by $\gamma = 0.01$, shown in (c), with no corresponding change in $\mathcal{E}_{\text{gen}}$. This indicates $S_{\mathrm{NT}}(f_{\boldsymbol{\theta}})$ is not predictive of $\mathcal{E}_{\text{gen}}$. As in Figure 1 there is no significant difference between the NN and corresponding NTK for the CNN in (a). See Tables 3 and 4 for the architecture.

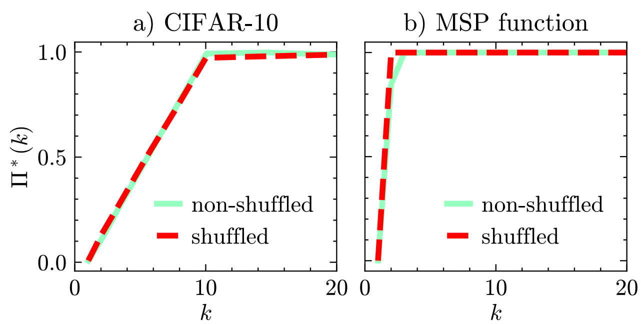

Figure 3: **The Cumulative quality of features Π^* is decoupled from generalization.** (a) a ResNet on CIFAR-10 and (b) an FFNN trained on MSP functions, each with shuffled and non-shuffled data.

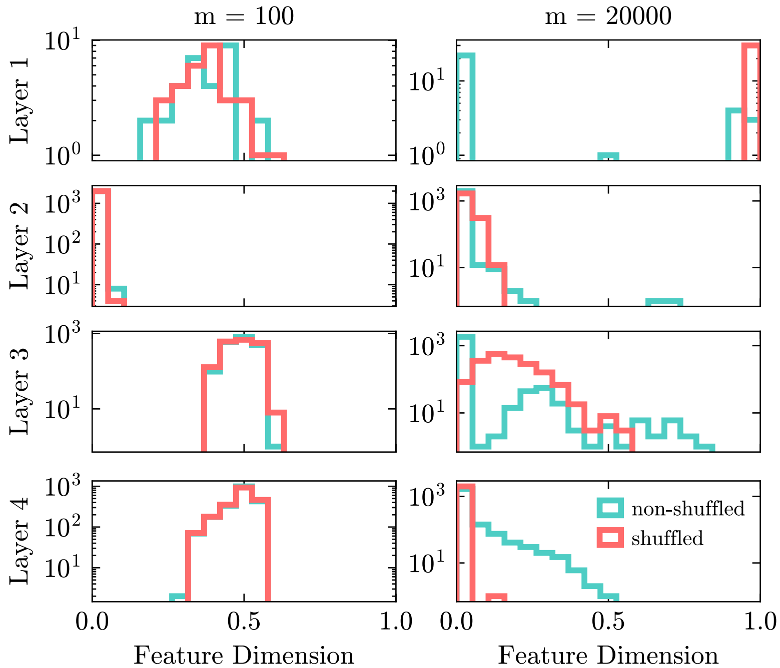

Figure 4: **Feature dimensionality and generalization are not strongly correlated** Histogram of the feature dimensionality for an FFNN with depth $L = 4$ and width $N = 2000$ trained on MSP functions for training set sizes $m = 100$ and 20000. When $m = 100$, there is no significant difference in the histograms, despite higher test losses for shuffled data (9.5 vs 6.3). At $m = 20000$, shuffled data exhibits mostly zero D_{f_i} with few non-zero features, a pattern consistent with FL, while non-shuffled data shows a diffuse distribution. The stark difference in test losses (6.4 vs 2×10^{-7}) despite these patterns demonstrates that final layer feature dimensionality poorly predicts generalization.

Appendix B. Background

Let $\mathcal{X} \subseteq \mathbb{R}^{n_0}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_L}$ be the input and output space. A dataset of size m, $\mathcal{D} = (\boldsymbol{x}_\mu, \boldsymbol{y}_\mu)_{\mu=1}^m$ is drawn i.i.d. from the data generating distribution $p(\boldsymbol{x}, \boldsymbol{y})$. For some function f, the generalization is defined with respect to a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$,

$$\mathcal{E}_{\text{gen}}(f) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p(\boldsymbol{x},\boldsymbol{y})} \left[\ell(f(\boldsymbol{x}), \boldsymbol{y})\right]. \tag{4}$$

In practice, we will approximate this quantity by averaging over a finite test set.

Definition 10 (Feed-forward Neural Network (FFNN)) *An L-layer FFNN is a recursively defined map $f_{\boldsymbol{\theta}} : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$:*

$$\boldsymbol{h}^0 = \boldsymbol{x}_\mu, \quad \boldsymbol{h}^l = \boldsymbol{W}^l \phi(\boldsymbol{h}^{l-1}) + \boldsymbol{b}^l, \tag{5}$$

and $f(\boldsymbol{x}_\mu) = \boldsymbol{W}^L \boldsymbol{h}^{L-1}(\boldsymbol{x}_\mu) + \boldsymbol{b}^L$, where $1 \leq l \leq L$, $\boldsymbol{W}^l \in \mathbb{R}^{n_l \times n_{l-1}}, \boldsymbol{b}^l \in \mathbb{R}^{n_l}$, and ϕ are nonlinear functions applied element-wise. We assume all n_l are equal for $l \neq 0, L$, and call this the width of the FFNN and denote P for the total number of parameters.

B.1. Kernel methods

Definition 11 (Kernel & Features) *A kernel is any symmetric, positive semi-definite function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let $\mathcal{H}$ be the reproducing kernel Hilbert space (RKHS) with inner product $\langle \cdot | \cdot \rangle_{\mathcal{H}}$. Then, any such kernel can be written as an inner product kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi_K(\boldsymbol{x}) | \Phi_K(\boldsymbol{x}') \rangle_{\mathcal{H}}$. The kernel's feature map is given by $\Phi_K : \mathcal{X} \to \mathcal{H}$.*

Definition 12 (l-layer feature map) *Consider the feature map of the $l < L$'th layer of an FFNN at training time t,*

$$\Phi^l(t) : \mathcal{X} \to \mathbb{R}^{n_l}, \quad \boldsymbol{x}_\mu \mapsto \boldsymbol{h}^l(\boldsymbol{x}_\mu). \tag{6}$$

The l'th layer feature kernel $K_{\Phi^l}(t) : \mathcal{X}^2 \to \mathbb{R}$ is given by

$$K_{\Phi^l}(\boldsymbol{x}_\mu, \boldsymbol{x}_\nu; t) = \Phi^l(\boldsymbol{x}_\mu; t)^\top \Phi^l(\boldsymbol{x}_\nu; t). \tag{7}$$

When evaluated over a finite dataset $\mathcal{D}$, $(K_{\Phi^l})_{\mu\nu}$ can be interpreted as the correlation matrix, measuring how similar the features of $\boldsymbol{x}_\mu$ and $\boldsymbol{x}_\nu$ are at layer l.

B.2. Learning dynamics and spectral bias for kernels

When performing kernel ridge regression with gradient descent, the residual dynamics $r_t(\boldsymbol{x}) = f(\boldsymbol{x}) - f^*(\boldsymbol{x})$ for projections on eigenfunctions e_ρ follow $\langle r_t | e_\rho \rangle_{\mathcal{H}} = e^{-\lambda_\rho t} \langle r_0 | e_\rho \rangle_{\mathcal{H}}$. For high-dimensional kernels, where the number of eigenfunctions N_ρ greatly exceeds the number of samples m, the distribution over λ_ρ determines the solution: eigenfunctions with large λ_ρ will have large coefficients [26, 27, 53]. Eigenfunctions with large λ_ρ are learned the fastest, so a trained solution will be dominated by the corresponding eigenfunctions. These often correspond to low frequency, simple components of the target function, rigorously proven for ReLU networks in [9]. The high-frequency components, which could lead to overfitting, are naturally learned more slowly.

The generalization error $\mathcal{E}_{\text{gen}}$ scales as :

$$\mathcal{E}_{\text{gen}}(m) \sim m^{-\beta}, \text{ with } \beta = \frac{1}{d} \min(\alpha_T - d, 2\alpha_S) \tag{8}$$

where the exponent β reflects how quickly different frequency components are learned, and α_T, α_S are the decay rates of the kernel in Fourier space [10, 59]. For β to remain non-vanishing as dimension d increases, the smoothness index $s = (\alpha_T - d)/2$ must scale with d (curse of dimensionality).

B.3. Kernels on CIFAR-10

NTKs of CNNs are multi-dot product kernels $k(x, z)$ that operate over the multi-sphere $\Pi_{i=1}^d \mathbb{S}^{\zeta - 1}$ where d is the number of pixels and ζ is the number of channels [26]. These kernels can be decomposed into eigenfunctions, which are multivariate spherical harmonics. The eigenvalue $\lambda_{\boldsymbol{k}}$ for the frequencies $\boldsymbol{k}$ of the multivariate spherical harmonic exhibits polynomial decay with respect to these frequencies. This decay induces an implicit bias that favors learning low-frequency functions before high-frequency ones, manifesting as a form of simplicity bias.
The multiplicity of the eigenvalues is determined by the quantity $p_i^{(L)}$, which represents the number of paths for a pixel i in a network of depth L. This path count quantifies the distinct number of ways information from a pixel can propagate through the network's convolution layers to reach a particular output. For a pixel, the number of paths decays exponentially with the distance from the

center of the receptive field, introducing a positional bias that facilitates learning spatially localized features over those requiring global image dependencies. This bias aligns with natural image statistics, where meaningful features typically exhibit local coherence. Consequently, CNNs can more efficiently learn localized high-frequency patterns compared to patterns requiring high frequencies across multiple pixels, a distinction not present in fully connected networks. This theoretical framework is further supported by [29], who demonstrate that such kernels generalize effectively when image labels depend on low frequencies (frequency bias) and the image spectrum itself is concentrated in low frequencies (positional bias), conditions commonly satisfied in real-world image datasets.

To conclude, alignment of the kernel with the target function can largely explain generalization of the NTK on specific datasets where this alignment exists, such as image classification, whereas when this alignment is absent, as in merged staircase functions, the NTK fails to generalize effectively.

B.4. Generalization theory of Kernels

To analyze the generalization behavior of the NTK, we need to first examine the theoretical foundations of generalization in kernel methods. Given that the kernels eigenfunctions are a basis of the RKHS 11, we can decompose the predicted $f^*(x) = \sum_\rho w^*_\rho \sqrt{\eta_\rho}\phi_\rho(x)$ and target function $\bar{f}(x) = \sum_\rho \bar{w}_\rho \sqrt{\eta_\rho}\phi_\rho(x)$ in terms of the eigenfunctions. This allows for decomposing the generalization error in terms of modes[3]

$$\mathcal{E}(m) = \sum_\rho \eta_\rho \left\langle (w^*_\rho - \bar{w}_\rho)^2 \right\rangle_\mathcal{D} = \sum_\rho \eta_\rho \mathcal{E}_\rho(m) \tag{9}$$

where the average over datasets can be analytically computed [10]. We note that the spectrum $\{\eta_\rho\}$ is independent of the target function, while the mode error $\mathcal{E}_\rho$ is not.

As we focus on learning curves, we want to understand how $\mathcal{E}(m)$ scales with m. Qualitatively, the scaling is dominated by two quantities [10, 14]. The first one is *spectral alignment*. It can be formally shown that for $\eta_\rho > \eta_{\rho'}$, $\mathcal{E}_\rho(m)$ decreases faster with m than $\mathcal{E}_{\rho'}(m)$. This means that, with growing training set size, eigenfunctions with larger eigenvalues of the trained function approach the one of the target function faster. Hence, if the target function is well approximated by the high eigenvalue eigenfunctions of the kernel, the generalization error will drop faster. Secondly, the *asymptotic mode error* has the form $\mathcal{E}_\rho(m) \underset{m\to\infty}{\sim} \frac{\langle \bar{w}_\rho \rangle}{\eta_\rho}$. The asymptotic error of a mode is larger if the RKHS eigenvalue η_ρ is small, even if the coefficient $\overline{w}_\rho$ of the target eigenfunction is large. Both of these observations motivate the definition of the cumulative power distribution.

Definition 13 *The cumulative power distribution is defined as the amount of overlap of the target function with the RKHS subspace up to mode ρ:*

$$C(\rho) = \frac{\sum_{\rho' \leq \rho} \eta_{\rho'} \overline{w_{\rho'}}^2}{\sum_{\rho'} \eta_{\rho'} \overline{w_{\rho'}}^2} \tag{10}$$

To conclude, the more power the target function has in the high eigenvalue subspace of the RKHS, the faster kernel ridge regression is able to learn the function with growing data set size. High

3. There is a fundamental lower bound for the test error due to zero modes.

task-model alignment results in a faster decaying learning curve, which allows a qualitative understanding of learning curves (see [10, 13, 67] for numerical studies and [62] for a critique of the theory).

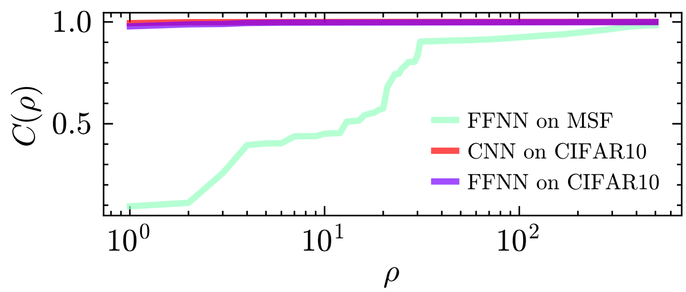

Figure 5: Cumulative power distribution for a FFNN trained on merged staircase functions as well as a FFNN and CNN trained on CIFAR-10. This can correctly predict the different generalization errors observed in Figure 1.

Appendix C. Related work

For a discussion of FL definitions, we refer readers to Section 3, which provides a thorough literature review. See Table 6 for an overview of datasets where kernels and NNs provably show a separation in sample complexity. In the line of works on sample complexity, we highlight several studies on NN scaling laws [11, 41, 58], alongside theoretical predictions of learning curves in the infinite width limit in [16]. Recent work has critically examined the explanatory power of the NTK for NN generalization [29, 50, 65, 69]. For a statistical physics-inspired predictive FL theory, we refer to [2] and [56].

Appendix D. Data sets and experiment details

D.1. Functions with the merged-staircase property

In this section we follow [1]. For any function $f : \{+1, -1\}^d \to \mathbb{R}$, we can express it using the Fourier-Walsh basis decomposition

$$f(z) = \sum_{S \subseteq [d]} \hat{f}(S) \chi_S(z), z \in \{+1, -1\}^d, \tag{11}$$

with Fourier coefficients $\hat{f}(S)$ and basis functions $\chi_S(z) := \prod_{i \in S} z_i$. This provides a representation of $f(z)$ through orthogonal monomials $\chi_S(z)$ weighted by their respective Fourier coefficients.

Definition 14 (Merged-Staircase Property) *We say a set structure* $\mathcal{S} = \{S_1, \ldots, S_m\} \subseteq 2^{[d]}$ *exhibits the Merged-Staircase Property (MSP) if there exists an ordering where each set* S_i, $i \in [m]$, *satisfies:*

$$|S_i \setminus \cup_{i' < i} S_{i'}| \geq 1. \tag{12}$$

This property ensures that each set contributes at least one novel element not contained in the union of preceding sets.

Definition 15 (Merged-Staircase Property for Functions) *Let $\mathcal{S} \subset 2^{[d]}$ be non-zero Fourier coefficients of f. We say that f satisfies the Merged-Staircase Property (MSP) if $\mathcal{S}$ has a MSP set structure.*

In the empirical experiments, we use $f(z) = z_7 + z_2 z_7 + z_0 z_2 z_7 + z_4 z_5 z_7 + z_1 + z_0 z_4 + z_3 z_7 + z_0 z_1 z_2 z_3 z_4 z_6 z_7$ with $d = 30$.

D.2. Multi-index functions

Here, we follow [17]. Multi-index functions are polynomials that depend on a small number of latent directions. Following Assumptions 1 and 2 in [17] from the theoretical analysis, we construct functions of the form $f(\boldsymbol{x}) = g(\langle \boldsymbol{x}, \boldsymbol{u}_1 \rangle, \ldots, \langle \boldsymbol{x}, \boldsymbol{u}_r \rangle)$ where $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$ are linearly independent vectors spanning the principal subspace S^*, while ensuring the non-degeneracy condition that the expected Hessian $H = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[\nabla^2 f(\boldsymbol{x})]$ has rank exactly r.

Specifically, we first construct a random orthogonal projection matrix $U \in \mathbb{R}^{d \times r}$ through QR decomposition of a Gaussian random matrix, where $r \ll d$ represents the intrinsic dimension of the target function. This ensures linear independence of the latent directions. Input data is sampled from a standard normal distribution $X \sim \mathcal{N}(0, I_d)$ and projected onto this latent space via $X_{latent} = XU$. The target polynomial function is then constructed as a sum over all multi-indices $\alpha \in \mathbb{N}^r$ with total degree at most p, where each term has a random Gaussian coefficient $c_\alpha \sim \mathcal{N}(0, 1)$: $f(\boldsymbol{x}) = \sum_{\alpha : \|\alpha\|_1 \leq p} c_\alpha \prod_{i=1}^{r} (U^T \boldsymbol{x})_i^{\alpha_i}$.

To model measurement noise, we add symmetric binary noise $\epsilon \sim \mathcal{U}(\{-\sigma, \sigma\})$ to obtain the final labels $y = f(\boldsymbol{x}) + \epsilon$. For evaluation, we generate test data using the same projection matrix U and coefficients c_α but with fresh input samples. The outputs are normalized to have zero mean and unit variance based on training set statistics.

Appendix E. μP-parameterization

When training with μP parameterization, we follow the definition of the μP parameterization in [22]. For an L-layer feed-forward NN with width n_l and input dimension n_0, muP prescribes specific initialization and learning rate scaling rules:

Initialization The weights $\boldsymbol{W}^l$ at each layer are initialized as:

$$\boldsymbol{W}^1 \sim \mathcal{N}\left(0, \frac{1}{n_0}\right), \quad \boldsymbol{W}^l \sim \mathcal{N}\left(0, \frac{1}{n_{l-1}}\right) \quad 2 \leq l \leq L, \quad \boldsymbol{W}^{L+1} \sim \mathcal{N}\left(0, \frac{1}{n_L}\right) \tag{13}$$

All bias terms are initialized to zero:

$$\boldsymbol{b}^l = \mathbf{0} \quad \forall l \in 1, \ldots, L+1 \tag{14}$$

Learning Rate Scaling The learning rates η_l for each layer follow:

$$\eta_1 = \eta_{\text{base}}, \quad \eta_l = \frac{\eta_{\text{base}}}{n_{l-1}} \quad 2 \leq l \leq L, \quad \eta_{L+1} = \frac{\eta_{\text{base}}}{n_L} \tag{15}$$

where η_{base} is the base learning rate.

E.1. Figure 1

Hyperparameter	**MSP Experiment**	**Multi-index functions Experiment**
latent dimension	—	3
polynomial degree	—	5
noise std	—	0.0
P (MSP parameter)	8	—
d (input dimension)	30	20
# hidden layer	4	4
hidden layer sizes	[400]	[400]
activation	ReLU	ReLU
batch size	64	64
epochs	5000	5000
learning rate	0.05	0.001
weight decay	10^{-4}	10^{-4}
initialization mode	muP Pennington	muP Pennington
γ	1	1
test set size	1000	10000
optimizer	Adam (muP mode)	Adam (muP mode)
learning rate scheduler	CosineAnnealing	CosineAnnealing
gradient clipping	1.0	1.0
MSP sets	$\{7\}, \{2,7\}, \{0,2,7\}, \{5,7,4\}, \{1\}, \{0,4\}, \{3,7\}, \{0,1,2,3,4,6,7\}$	

Table 1: Hyperparameter settings for both MSP and multi-index functions experiments in Figure 1 (a), (b).

Hyperparameter	**Value**
Architecture	WideResNet [72]
Block size	4
Width multipliers (k)	4.0
Number of classes	10
Initial channels	16
Channel progression	[16k, 32k, 64k]
Dataset	CIFAR-10
Input normalization	divide by 255
Batch size	128
Epochs	200
Learning rate	0.001
Optimizer	Adam
Loss function	MSE with one-hot targets
Learning rate scheduler	CosineAnnealing
Number of test samples	10000

Table 2: Hyperparameter settings for training on CIFAR-10 Figure 1 (c).

E.2. Figure 2

Hyperparameter	**Value**
Architecture	FFNN
Hidden sizes	[400, 1000]
Depth	4
Weight initialization	He ($1/\sqrt{N}$)
Input dimension (d)	30
Training Parameters	
Test set size	5000
Training set sizes	[10, 100, 250, 500, 750, 1000, 2500, 5000, 7500, 10000, 20000]
Batch size	64
Epochs	3000
Learning rate	0.005
Weight decay	10^{-4}
Optimizer	AdamW
LR scheduler	Cosine decay
Gradient clipping	1.0
γ scaling	[1.0,0.01]
Number of experiments	3
MSP Parameters	
d	30
MSP sets	$\{7\}, \{2,7\}, \{0,2,7\}, \{5,7,4\}, \{1\}, \{0,4\}, \{3,7\}, \{0,1,2,3,4,6,7\}$
NTK Parameters	
NTK computation	Empirical, batched
Kernel regularization	$10^{-6} \cdot \text{tr}(K)/n$

Table 3: Hyperparameter settings for the experiments with FFNNs in Figure 2.

Hyperparameter	**Value**
Architecture	WideResNet [72]
Block size	4
Width multiplier (k)	2.0
Number of classes	10
Initial channels	16
Channel progression	[16k, 32k, 64k]
Normalization	LayerNorm
Training Parameters	
Input normalization	$\frac{x-\mu}{\sigma}$ (per channel)
Batch size	64
Epochs	2500
Base learning rate	0.0001
Weight decay	10^{-4}
Optimizer	AdamW
Loss function	Cross-entropy
LR scheduler	Cosine decay
Gradient clipping	10.0
Training set sizes	[10, 100, 500, 1000, 2000, 4000, 8192, 16384, 32768]
Test set size	10000
Number of experiments	3
NTK Parameters	
NTK computation	Empirical, batched
Kernel regularization	10^{-2} if $n > 8000$ else 10^{-4}

Table 4: Hyperparameter settings for experiments with the WideResNet in Figure 2.

E.3. Figures 3 and 4

Figure 3 uses the settings from Figure 1.

Hyperparameter	MSP Experiment
P (MSP parameter)	8
d (input dimension)	30
# hidden layer	4
hidden layer sizes	2000
activation	ReLU
batch size	64
epochs	5000
learning rate	0.001
weight decay	10^{-4}
initialization mode	muP Pennington
γ	[1, 0.0001]
test set size	1000
optimizer	Adam (muP mode)
learning rate scheduler	CosineAnnealing
gradient clipping	1.0
MSP sets	$\{7\}, \{2, 7\}, \{0, 2, 7\}, \{5, 7, 4\}, \{1\}, \{0, 4\}, \{3, 7\}, \{0, 1, 2, 3, 4, 6, 7\}$

Table 5: Hyperparameter settings for for Figure 4.

E.4. Addition to: Section 2

Source	Data	NN Type	Kernel	NN Scaling	Kernel Scaling
[17], [20], [29], [43], [63] [44]	$x \sim \mathcal{N}(0, I_d)$, $y = g(Ux)$, $U \in \mathbb{R}^{r \times d}$, deg p poly	1-hidden layer $N = O(r^p)$	NTK	$m = \Omega(dr^p + \frac{d^2 r}{\varepsilon^2})$	$m = \Omega(\frac{d^{p/2}}{\varepsilon^2})$
[1]	MSP function, input dim. d max. poly. degree. P	1-hidden layer NN with $N = e^{e^P}$	Any	$m = O(d \cdot 2^{2^{O(P)}}/\varepsilon^5)$	$n = \Omega(d^P)$
[18], [61]	Sparse parity on k bits	1-hidden layer $N = \text{poly}(k)$	Any	$m = \Omega(\text{poly}(k, \frac{1}{\varepsilon}))$	$m = \Omega(\frac{2^k}{\varepsilon^2})$
[54]	d-dim Gaussian mixture 4 clusters XOR config	1-hidden layer $O(1)$ width	ReLU RFK $O(d)$ feats	$\epsilon_{NN} = \Theta(1)$ $m = \Omega(d)$	$\epsilon_{RF} = \Omega(1)$ $m = O(d)$
[24], [68]	Noisy 2-XOR cluster d-dim distribution	1-hidden layer	NTK	$m = O(\frac{1}{\varepsilon^2})$	$m = \Omega(\frac{d^2}{\varepsilon^2})$
[60]	Spiked d-dim cumulant model (≥ 4 cumulants)	1-hidden layer $N \geq 5d$	ReLU RFK $O(d)$	$\epsilon_{NN} = O(1)$ $m = \Omega(d^2)$	$\epsilon_{RF} = \frac{1}{2}$ $m = O(d^2)$
[3], [4], [5], [39], [71]	Uniform on S^{d-1}, Two-layer ReLU teacher network width N	1-hidden layer width N	Any	$m = O(\frac{N^5}{\varepsilon})$ fixed width	$m = \Omega(\varepsilon^{-\frac{d+2}{2d+2}})$

Table 6: Comparison of NN and kernel method scaling for different target functions.

E.5. Addition to: Section 3

E.6. 2. Family: Spectra

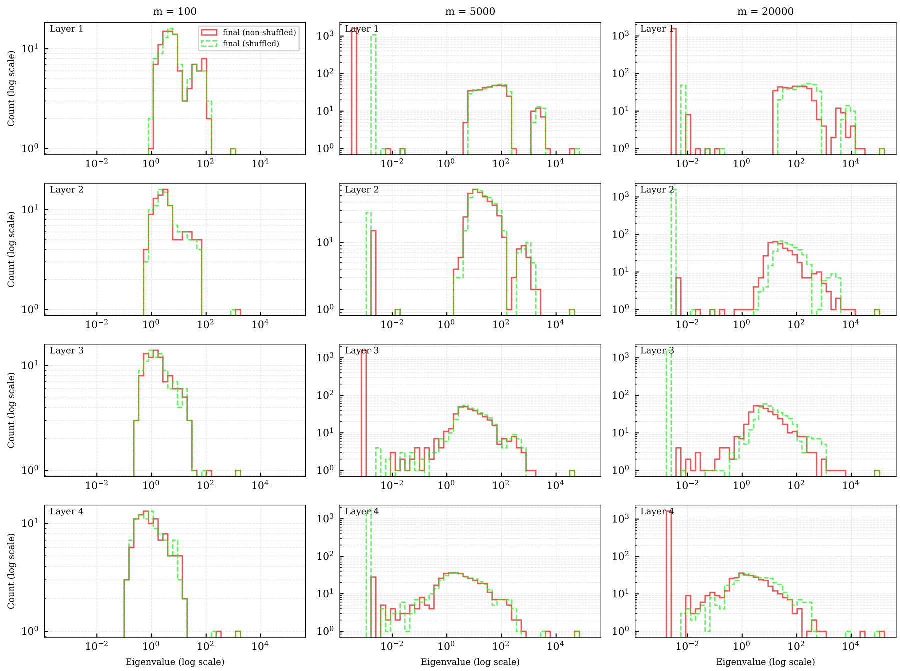

Figure 6: CK spectrum for a NN trained with standard parameterization and $N = 400$.

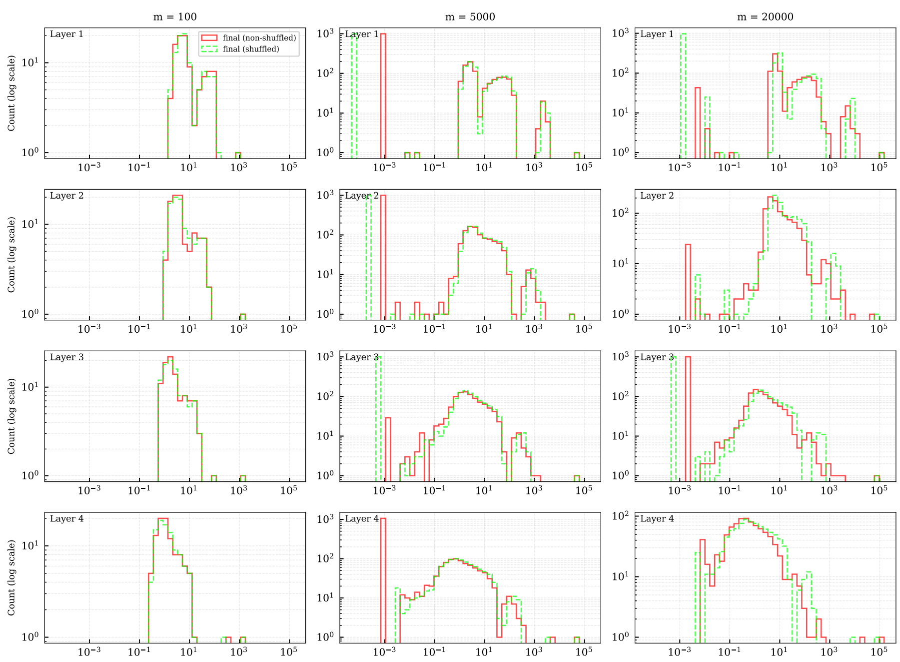

Figure 7: CK spectrum for a NN trained with standard parameterization and $N = 1000$.

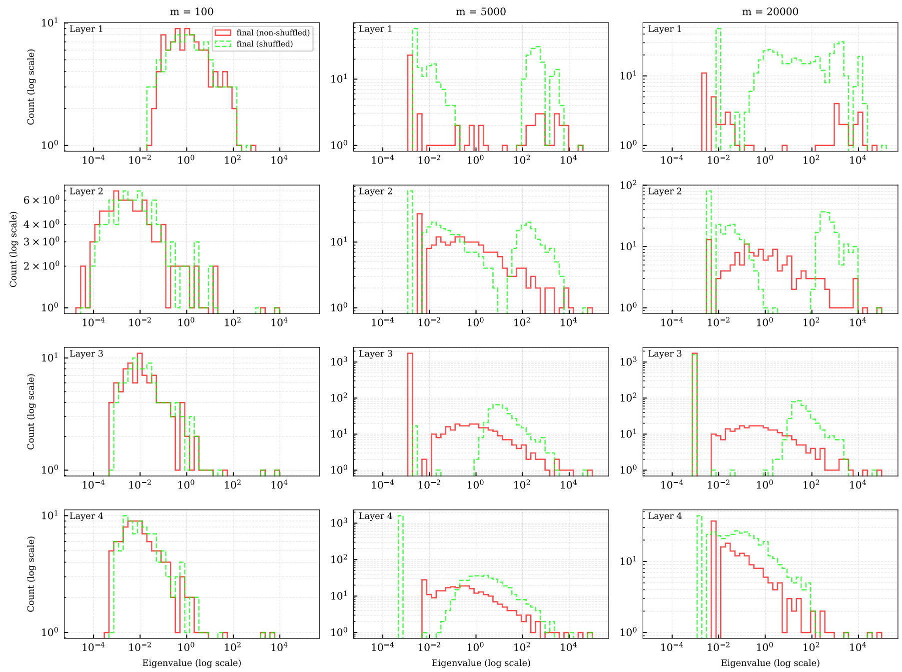

Figure 8: CK spectrum for a NN trained with μP-parameterization and $N = 400$.

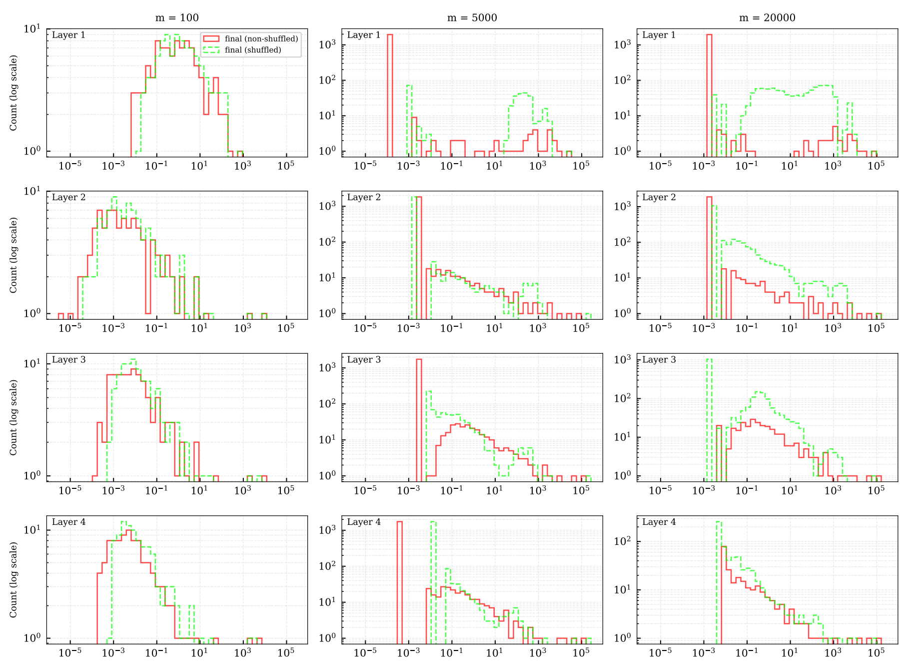

Figure 9: CK spectrum for a NN trained with μP-parameterization and $N = 1000$.

E.7. 3. Family: Distribution plots

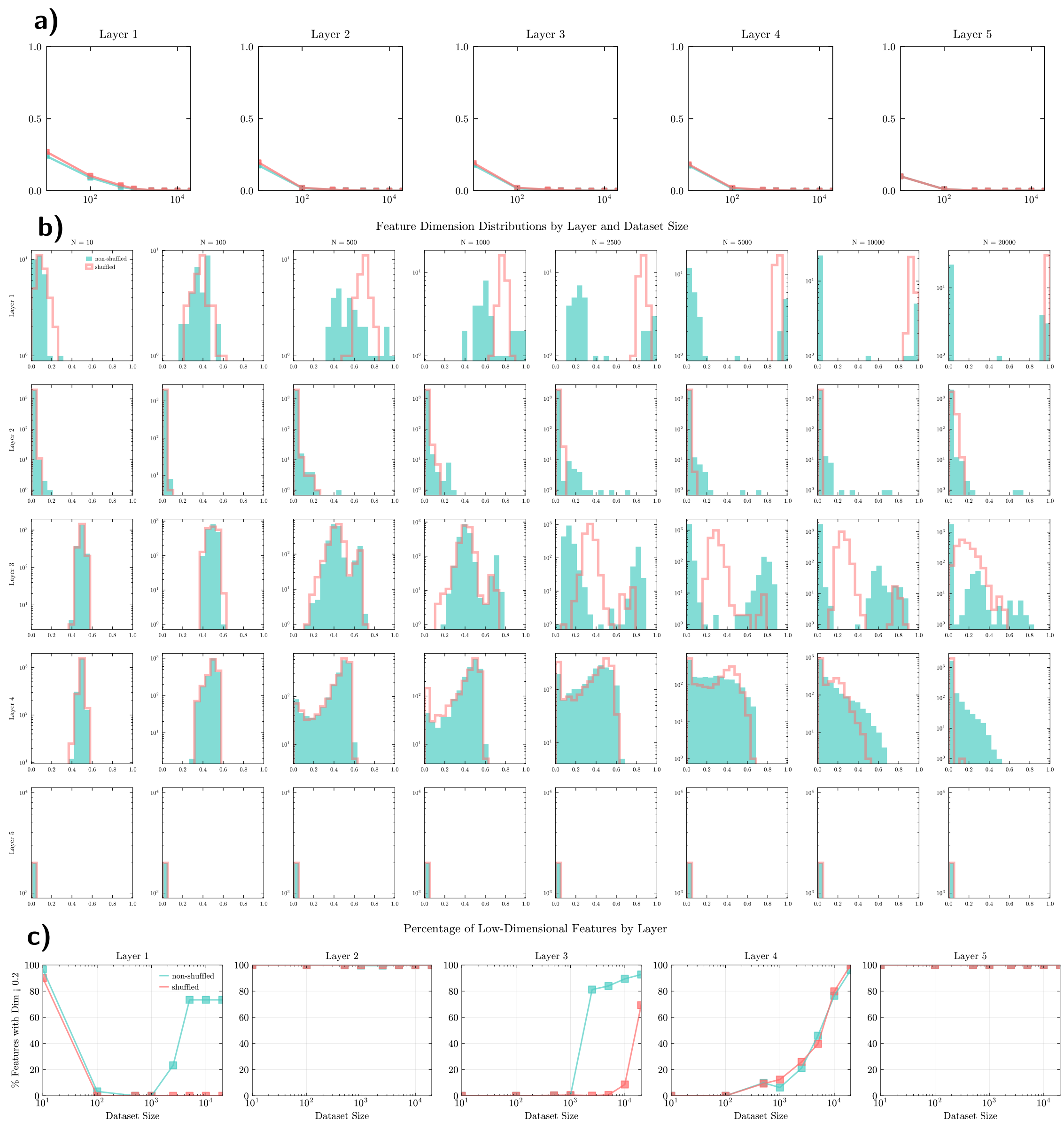

Figure 10: NNs (width 2000) trained with μP parameterization on MSP functions across varying training set sizes m with $\gamma = 1$. (a) Sample complexity analysis fails to differentiate between shuffled and non-shuffled data. (b) Per-layer feature dimensionality comparison between shuffled and non-shuffled datasets reveals diffuse patterns across training set sizes. (c) Relative number of dimensions with $D_{f_i} = 0$. The proportion for the first layer rises around m^*. This is not observed in later layers.

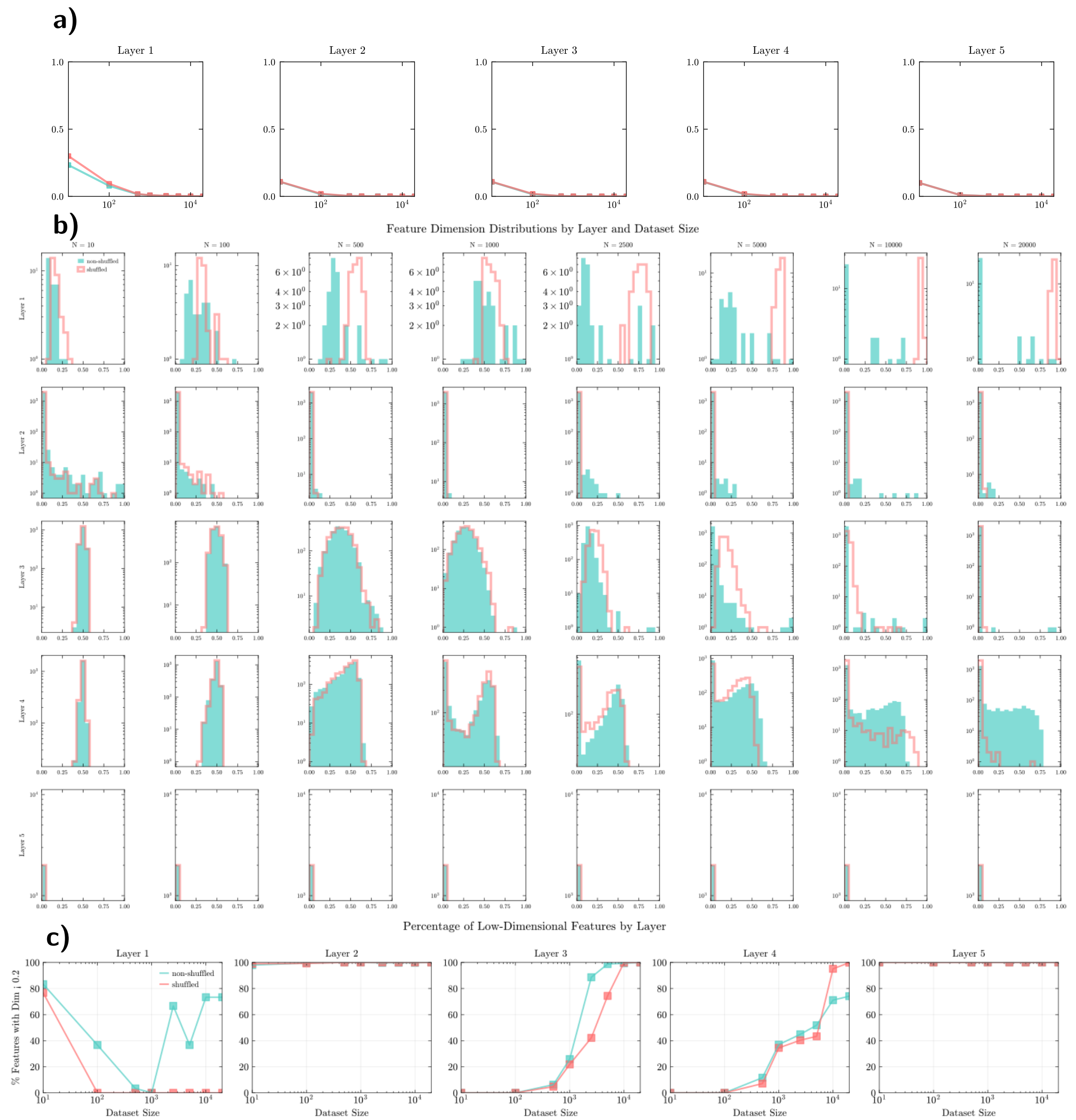

Figure 11: Same as Figure 10 but with $\gamma = 0.0001$. $\gamma = 0.0001$ moves the weight of the D_{f_i} distributions closer to 0.

E.8. Additional information on Δ_{NT}

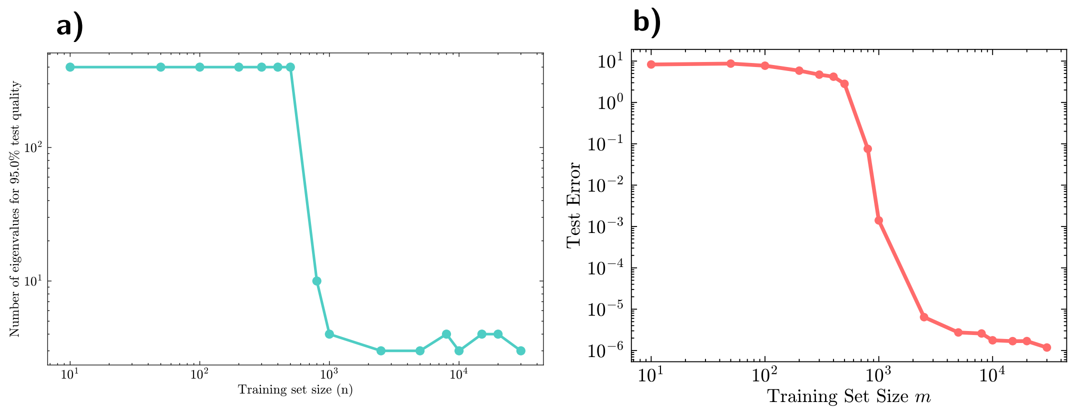

Figure 12: NNs (width 400, depth 4) trained with μP on MSP functions. (a) S_{CK} computed via projection onto the target function using $Q_k = \langle e_k | f^* \rangle$ instead of the learned function, quantifying how well the top k eigenfunctions approximate the target function. (b) Generalization error versus training set size. This demonstrates that S_{CK} can predict the generalization error as both curves strongly correlate.

In the following we will define the critical dataset size m^* more formally.

Definition 16 *Given a data generating model $p(\boldsymbol{x}, \boldsymbol{y})$ and an i.i.d. dataset $\mathcal{D}$ of size m, a FFNN f_θ of width N and depth D with μP-parameterization and base learning rate η_0, we define the critical training set size m^* as the smallest dataset size such that*

$$\exists N^* : \quad \forall N \geq N^*, \forall m \geq m^* : \quad \frac{\mathcal{E}(f_\theta^*; m)}{\mathcal{E}(f_{NTK}^*; m)} < \varepsilon \tag{16}$$

where the generalization error of the NN is smaller than the one of the NTK by a factor of ε, typically taken to be $\varepsilon \approx 1/10$ or smaller.

Dependence on N, D For the μP-parameterization, we find that m^* exhibits weak dependence on the initial learning rate η_0 and depth D, while remaining independent of width N. Base learning rates that are too small lead to very slow training. As depth increases, the error $\mathcal{E}(f_\theta^*; m)$ decreases for fixed m, generally resulting in smaller values of m^*. In contrast, under standard parameterization, m^* shows width dependence—an artifact of suboptimal hyperparameter selection rather than an intrinsic property.

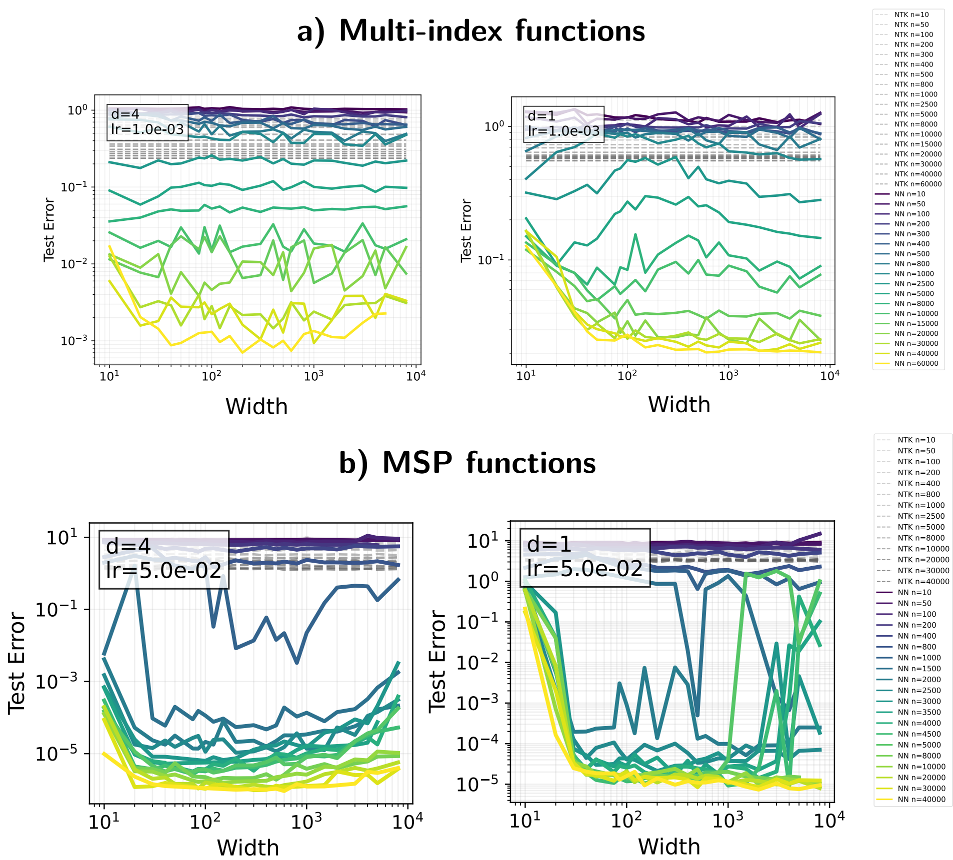

Figure 13: Generalization error of NNs with varying widths and depths ($d = 1, 4$) trained with μP on (a) multi-index functions and (b) MSP functions. The results demonstrate a width-independence threshold: beyond a critical width where the network achieves sufficient expressivity, the generalization error remains consistent regardless of further width increases.

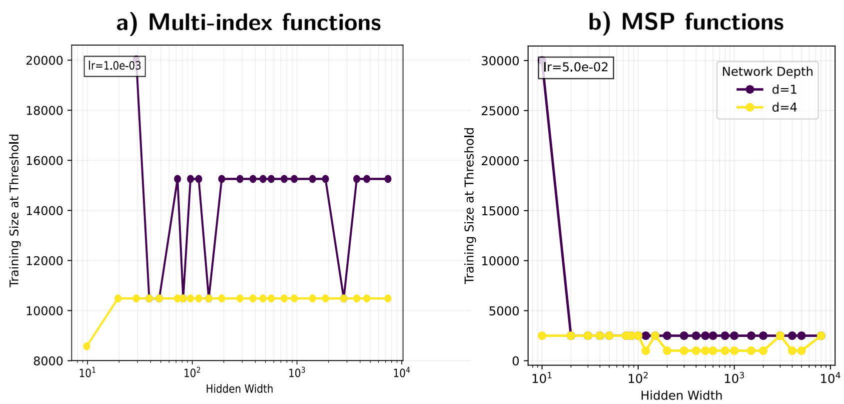

Figure 14: Critical dataset size m^* as a function of network width for (a) multi-index functions and (b) MSP functions, demonstrating that m^* beyond a certain width threshold, exhibits width independence.

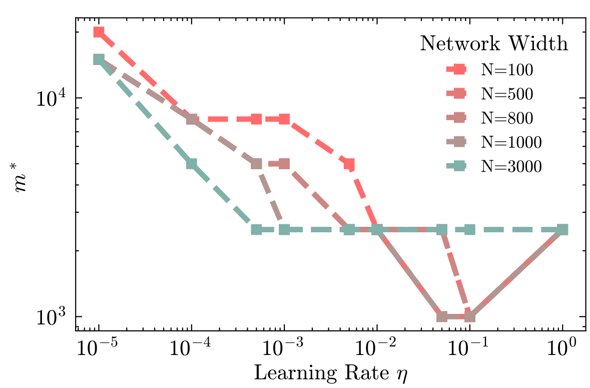

Figure 15: Critical dataset size m^* plotted against base learning rate for MSP functions, revealing a stable region where m^* remains constant across a specific range of learning rates.