Gradient descent induces alignment between weights and the pre-activation tangents for deep non-linear networks

Daniel Beaglehole DBEAGLEHOLE@UCSD.EDU
UC San Diego, La Jolla, United States

Ioannis Mitliagkas IOANNISM@GOOGLE.COM
Google DeepMind, Mountain View, United States
Mila, Montreal, Canada

Atish Agarwala THETISH@GOOGLE.COM
Google DeepMind, Mountain View, United States

Abstract

Understanding the mechanisms through which neural networks extract statistics from input-label pairs is one of the most important unsolved problems in supervised learning. Prior works have identified that the gram matrices of the weights in trained neural networks of general architectures are proportional to the average gradient outer product of the model, in a statement known as the *Neural Feature Ansatz* (NFA). However, the reason these quantities become correlated during training is poorly understood. In this work, we clarify the nature of this correlation and explain its emergence at early training times. We identify that the NFA is equivalent to alignment between the left singular structure of the weight matrices and the newly defined pre-activation tangent kernel. We identify a centering of the NFA that isolates this alignment and is robust to initialization scale. We show that, through this centering, the speed of NFA development can be predicted analytically in terms of simple statistics of the inputs and labels.

1. Introduction

Neural networks have emerged as the state-of-the-art machine learning methods for seemingly complex tasks, such as language generation [5], image classification [7], and visual rendering [8]. The precise reasons why neural networks generalize well have been the subject of intensive exploration, beginning with the observation that standard generalization bounds from statistical learning theory fall short of explaining their performance [16].

Instead, the success of neural networks has been largely attributed to *feature learning* - the ability of neural networks to learn statistics, measurements, and representations of data which are useful for downstream tasks. However, the specific mechanism through which features are learned is an important unsolved problem in deep learning theory. A number of works have studied the abilities of neural networks to learn features in structured settings [1–3, 6, 9, 11, 12]. Some of that work proves strict separation in terms of sample complexity between neural networks trained with stochastic gradient descent and kernels [10].

The work above studies simple structure, such as learning from low-rank data or functions that are hierarchical compositions of simple elements. Recent work makes a big step towards generalizing these assumptions by proposing the *neural feature ansatz* (NFA) [4, 13], a general structure that emerges in the weights of trained neural networks. The NFA states that the gram matrix of the

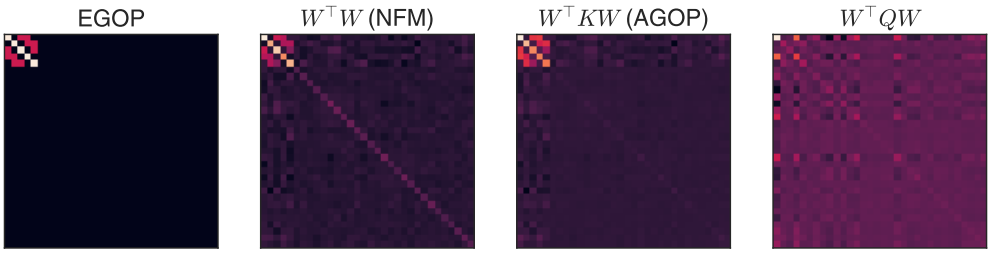


Figure 1: Various feature learning measures for target function $y(x) = \sum_{k=1}^{r} x_{k \bmod r} \cdot x_{(k+1) \bmod r}$ with $r = 5$ and inputs drawn from standard normal. The EGOP $\mathbb{E}_{x\sim\mu}\left[\frac{\partial y}{\partial x}\frac{\partial y}{\partial x}^\top\right]$ (first plot) captures the low-rank structure of the task. The NFM $\left(W^\top W\right)$ (second plot) and AGOP $\left(W^\top KW\right)$ (third plot) of a fully-connected network are similar to each other and the EGOP. Replacing K with a symmetric matrix Q with the same spectrum but independent eigenvectors obscures the low rank structure (fourth plot), and reduces the correlation from $\rho\left(F, \bar{G}\right) = 0.93$ to $\rho\left(F, W^\top QW\right) = 0.53$.

weights at a given layer (known as the *neural feature matrix* (NFM)) is aligned with the *average gradient outer product* (AGOP) of the network with respect to the input to that layer. In particular, the NFM and AGOP are highly correlated in all layers of trained neural networks of general architectures, including practical models such as AlexNet [7] and VGG [14].

A major missing element of this theory is the reason the AGOP and NFM become correlated during training. In this paper, we clarify the nature of this correlation and explain its emergence at early training times. We establish that the NFA is equivalent to alignment between the left singular structure of the weight matrices and the *pre-activation neural tangent kernel* (Section 2). We introduce a centering of the NFA that isolates this particular alignment and is robust to initialization scale (Section 2). Further, we show that, through this centering, the alignment speed can be understood analytically through the centered NFA in terms of the statistics of the data and labels, and can be manipulated theoretically (Section 3).

2. Alignment between the weight matrices and the pre-activation tangent kernel

2.1. Preliminaries

We consider fully-connected neural networks with a single output of depth $L \geq 1$, where L is the number of hidden layers, written $f : \mathbb{R}^d \to \mathbb{R}$. We write the input to layer $\ell \in \{0, \ldots, L-1\}$ as x_ℓ, where $x_0 \equiv x$ is the original datapoint, and the pre-activation as $h_\ell(x)$. Then,

$$h_\ell(x) = W^{(\ell)} x_\ell, \qquad x_{\ell+1} = \phi(h_\ell(x)) \,,$$

where ϕ is an element-wise nonlinearity, $W^{(\ell)} \in \mathbb{R}^{k_{\ell+1} \times k_\ell}$ is a weight matrix, and k_ℓ is the hidden dimension at layer ℓ. We restrict k_{L+1} to be the number of output logits, and set $k_0 = d$, where d is the input dimension of the data. Note that $f(x) = h_{L+1}(x)$. We train f by gradient descent on a loss function $\mathcal{L}(\theta, X)$, where X is an input dataset, and θ is the collection of weights.

We consider a supervised learning setup where we are provided n input-label pairs $(x^{(1)}, y^{(1)})$ $, \ldots, (x^{(n)}, y^{(n)}) \in \mathbb{R}^d \times \mathbb{R}$. We denote the inputs $X \in \mathbb{R}^{n \times d}$ and the labels $y \in \mathbb{R}^{n \times 1}$. We train a

fully-connected neural network to learn the mapping from inputs to labels by minimizing a standard loss function, such as mean-squared-error or cross-entropy, on the dataset.

One can define two objects associated with neural networks that capture learned structure. For a given layer ℓ, the *neural feature matrix* (NFM) F_ℓ is the gram matrix of the columns of the weight matrix $W^{(\ell)}$, i.e. $F_\ell \equiv (W^{(\ell)})^\top W^{(\ell)}$. The second fundamental object we consider is the *average gradient outer product* (AGOP) $\bar{G}_\ell$, defined as $\bar{G}_\ell \equiv \frac{1}{n}\sum_{\alpha=1}^{n} \frac{\partial f(x_\ell^{(\alpha)})}{\partial x_\ell} \frac{\partial f(x_\ell^{(\alpha)})}{\partial x_\ell}^\top$. To understand the structure of these objects, consider the following *chain-monomial* low-rank task, where the target features have a closed form:

$$y(x) = \sum_{k=1}^{r} x_{k \bmod r} \cdot x_{(k+1) \bmod r} \ , \tag{1}$$

where the data inputs are sampled from an isotropic Gaussian distribution $\mu = \mathcal{N}(0, I)$. In this case, both objects capture the coordinates on which the target function depends (Figure 1). Prior work has shown that in trained neural networks, these objects will be approximately correlated to each other. This notion is formalized in the *Neural Feature Ansatz* (NFA):

Ansatz 1 (Neural Feature Ansatz [13]) *The Neural Feature Ansatz states that, for all layers $\ell \in [L]$ of a fully-connected neural network with L hidden layers trained on input data $x^{(1)}, \ldots, x^{(n)}$, the following correlation holds,*

$$\rho\left(\bar{G}_\ell, F_\ell\right) \approx 1 \ ,$$

Here, ρ is the cosine similarity, or *correlation*, with range $[-1, 1]$, and is defined as

$$\rho\left(A, B\right) = \text{tr}\left(A^\top B\right) \cdot \text{tr}\left(A^\top A\right)^{-1/2} \cdot \text{tr}\left(B^\top B\right)^{-1/2} \ .$$

for any two matrices $A, B \in \mathbb{R}^{d_1 \times d_2}$, for any d_1 and d_2.

The first and second arguments to the correlation are the NFM and AGOP with respect to the input of layer ℓ. Here, $\frac{\partial f(x)}{\partial x_\ell} \in \mathbb{R}^{k_\ell \times 1}$ denotes the gradient of the function f with respect to the intermediate representation x_ℓ. For simplicity, we may concatenate these gradients into a single matrix $\frac{\partial f(X)}{\partial x_\ell} \in \mathbb{R}^{n \times k_\ell}$. Note we consider scalar outputs in this work, though the NFA relation is identical when there are $c \geq 1$ outputs, where in this case $\frac{\partial f(x)}{\partial x_\ell} \in \mathbb{R}^{k_\ell \times c}$ is the input-output Jacobian of the model f.

We note that while the NFA states the correlation is approximately 1, in practice, the NFM and AGOP are highly correlated with correlation less than 1 (see e.g. Figure 2), where the final correlation depends on many aspects of training and architecture choice. For example, the final value of the NFA can be sensitive to the magnitude of the initial weights (e.g. first column of Figure 2). We parameterize this magnitude by the *initialization scale*, s_ℓ, for layer ℓ, where the initial weights are sampled i.i.d. as $W_{ij}^{(\ell)} \sim \mathcal{N}\left(0, \frac{s_\ell}{k_\ell}\right)$.

The relation between the NFM and the AGOP is significant, in part, because the AGOP of a model with respect to the first-layer inputs will approximate the *expected gradient outer product* (EGOP) of the target function [15] for networks that well-approximate the target function. In particular, as we will see later with the example of a low rank polynomial, the EGOP of the target

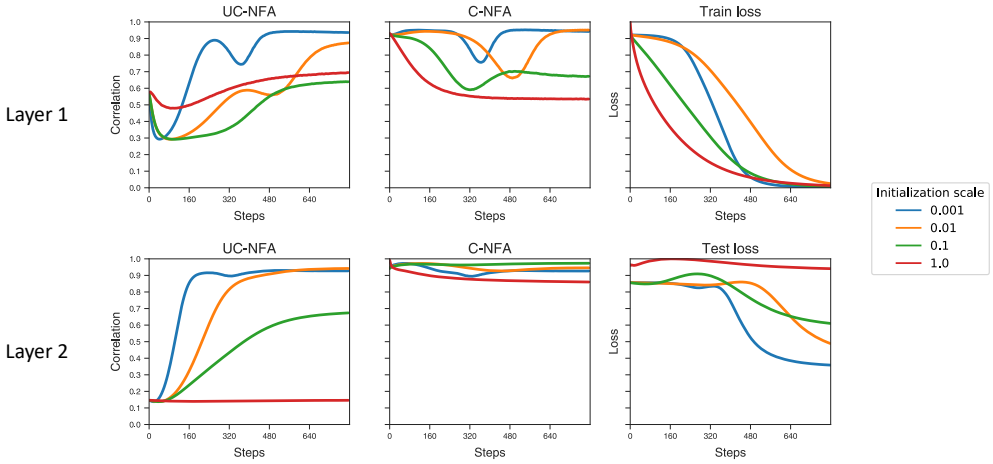


Figure 2: Uncentered and centered NFA correlations for a two hidden layer MLP trained on data drawn from a standard Normal distribution on the chain monomial task of rank $r = 5$. Only, the initialization scale of the first layer weights is varied, while $s_\ell = 1$ for $\ell > 0$. The top row shows the values for layer 1, while the bottom row are the values for layer 2. Train (test) losses are scaled by the maximum train (test) loss achieved so that they are between 0 and 1. Here width is 256, the input dimension is 32, and the dataset contains 256 points.

function contains task-specific structure that is completely independent of the model used to estimate it. Where the labels are generated from a particular target function $y(x) : \mathbb{R}^d \to \mathbb{R}$ on data sampled from a distribution μ, the EGOP is equal to,

$$\text{EGOP}(y, \mu) = \mathbb{E}_{x\sim\mu}\left[\frac{\partial y}{\partial x}\frac{\partial y}{\partial x}^\top\right] . \tag{2}$$

If the NFA holds, the correlation of the EGOP and the AGOP at the end of training also implies high correlation between the NFM of the first layer and the EGOP, so that the NFM has encoded this task-specific structure.

To demonstrate the significance of the approximate proportionality between the NFM and the AGOP in successfully trained networks, we return to the chain-monomial low rank task. In this case, $\text{EGOP}(y, \mu)$ will be rank r, where r is much less than the ambient dimension (Figure 1). Similarly, the AGOP of the trained model resembles the EGOP. In this case the NFA implies that the NFM also resembles the EGOP. Therefore, the neural network has learned the model-independent and task-specific structure of the chain-monomial task in the right singular values and vectors of the first layer weight matrix, as these are determined by the NFM. As a demonstration of this fact, the NFM of a trained neural network can be recovered up to high correlation from the AGOP of a fixed kernel method on real datasets [13].

2.2. Alignment decomposition

In order to understand the NFA, it is useful to decompose the AGOP. Doing so will allow us to show that the NFA can be interpreted as an alignment between weight matrices and the *pre-activation tangent kernel* (PTK).

For any layer ℓ, we can re-write the AGOP as,

$$\bar{G}_\ell = (W^{(\ell)})^\top K^{(\ell)} W^{(\ell)},\ K^{(\ell)} \equiv \frac{\partial f(X)}{\partial h_\ell}^\top \frac{\partial f(X)}{\partial h_\ell}$$

This gives us the following proposition:

Proposition 2 (Alignment decomposition of NFA)

$$\rho\left(F_\ell, \bar{G}_\ell\right) = \rho\left((W^{(\ell)})^\top W^{(\ell)}, (W^{(\ell)})^\top K^{(\ell)} W^{(\ell)}\right) .$$

This alignment holds trivially and exactly if $K^{(\ell)}$ is the identity. However, the correlation can be high in trained networks even with non-trivial $K^{(\ell)}$. For example, in the chain monomial task (Figure 1), $K^{(0)}$ is far from identity (standard deviation of its eigenvalues is 5.9 times its average eigenvalue), but the NFA correlation is 0.93 at the end of training. We also note that if $K^{(\ell)}$ is independent of $W^{(\ell)}$, the alignment is lower than in trained networks; in the same example, replacing $K^{(0)}$ with a matrix Q with equal spectrum but random eigenvectors greatly reduces the correlation to 0.53 and qualitatively disrupts the structure relative to the NFM (Figure 1, second row). We show the same result for neural networks trained on the CelebA dataset (see Appendix I). Therefore, the NFA has to do with the alignment of the left eigenvectors of $W^{(\ell)}$ with $K^{(\ell)}$ in addition to spectral considerations.

2.3. Centering the NFA isolates weight-PTK alignment

We showed that the NFA is equivalent to PTK-weight alignment (Proposition 2), and that it emerges during training (Figure 2 and Appendix G). We now ask: is the development of the NFA due to weights aligning with the current PTK, or the alignment of the PTK to the current weights?

In practice, both effects matter, but numerical evidence suggests that changes in the PTK do not drive the early dynamics of the NFA (Appendix B). Instead, we focus on the alignment of the weights to the PTK at early times. We can measure this alignment by considering the change in weights from their initialization.

Let $W_0^{(\ell)}$, $W^{(\ell)}$, and $K^{(\ell)}$ be the initial weight matrix, trained weight matrix, and interior feature matrix, respectively, at layer ℓ, and let $\bar{W}^{(\ell)} \equiv W^{(\ell)} - W_0^{(\ell)}$. We observe that the covariance of the NFM and AGOP can be decomposed into quantities that depend on the centered weights. In particular, substituting the definition of $\bar{W}^{(\ell)}$, we can see that the numerator of the NFA correlation $(W^{(\ell)})^\top W^{(\ell)} (W^{(\ell)})^\top K^{(\ell)} W^{(\ell)}$ contains the term $(\bar{W}^{(\ell)})^\top \bar{W}^{(\ell)} (\bar{W}^{(\ell)})^\top K^{(\ell)} \bar{W}^{(\ell)}$. We can perform a similar decomposition for both terms in the denominator. Hence, the centered NFM, $(\bar{W}^{(\ell)})^\top \bar{W}^{(\ell)}$, and the centered AGOP, $(\bar{W}^{(\ell)})^\top K^{(\ell)} \bar{W}^{(\ell)}$, contribute to the NFA correlation, while also isolating how the change in weights are aligned with the PTK. We define the correlation between the centered quantities, $\rho\left((\bar{W}^{(\ell)})^\top \bar{W}^{(\ell)}, (\bar{W}^{(\ell)})^\top K^{(\ell)} \bar{W}^{(\ell)}\right)$ as the *centered NFA* (C-NFA), while referring to the original correlation as the *uncentered NFA* (UC-NFA) to distinguish them.

Since $\bar{W} = 0$ at initialization, the early dynamics of the C-NFA are dominated by W aligning with the initial K:

Proposition 3 (Centered NFA dynamics) *At initialization, $\bar{W}^\top \bar{W} = \bar{W}^\top K \bar{W} = 0$, and,*

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top \bar{W}) = 0, \quad \frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top \bar{W}) = 2\dot{W}^\top \dot{W},$$
$$\frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top K \bar{W}) = 0, \quad \frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top K \bar{W}) = 2\dot{W}^\top K \dot{W} .$$

The first non-zero derivatives of the two quantities in the C-NFA give us the change in the gram matrix of W, as well as the change in the AGOP for fixed PTK K. This makes it a useful object to study weight-PTK alignment.

We note that the C-NFA correlation remains high throughout training across initialization scale (Figure 2, second column, and Appendix G), but is especially high and invariant to scale early on in training. This is in contrast to the UC-NFA, whose final value can be extremely sensitive to the initialization scale (e.g. second row, Figure 2).

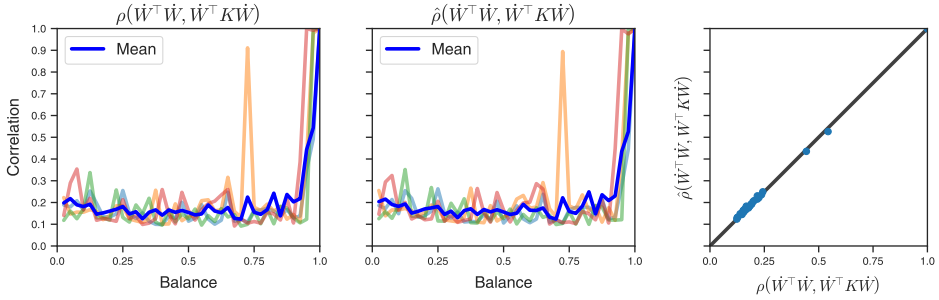


Figure 3: Predicted versus observed correlation of the second derivatives of centered F and $\bar{G}$ on the alignment reversing dataset. Different shaded color curves correspond to four different seeds for the dataset. The solid blue curve is the average over all data seeds. The rightmost sub-figure is a scatter plot of the predicted versus observed correlations of these second derivatives, with one point for each balance value. We instantiate the dataset in the proportional regime where width, input dimension, and dataset size are all equal to 1024.

3. Theoretically predicting the centered NFA

3.1. Early time C-NFA dynamics

To understand the dynamics of the weights aligning to the PTK, we compute the time derivatives of the C-NFA at initialization (Appendix A):

Proposition 4 (Centered NFA decomposition) *For a fully-connected neural network at initialization, whose weights are trained with gradient descent on a loss function $\mathcal{L}$,*

$$\dot{W}^\top\dot{W} = X^\top\dot{\mathcal{L}}\mathcal{K}\dot{\mathcal{L}}X, \quad \dot{W}^\top K\dot{W} = X^\top\dot{\mathcal{L}}\mathcal{K}^2\dot{\mathcal{L}}X \tag{3}$$

where $\dot{\mathcal{L}}$ is the $n \times n$ diagonal matrix of logit derivatives

$$\dot{\mathcal{L}} \equiv \mathrm{diag}\left(\partial\mathcal{L}/\partial f\right)$$

We immediately see from these equations how gradient-based training can drive the NFA. Even though the NFA doesn't explicitly involve the loss, the gradient descent dynamics of the NFM and AGOP depend on the labels in a similar way, and differ by a factor of the kernel $\mathcal{K}$ - similar to how the AGOP and NFM differ by a factor of K. The derivatives are perfectly correlated if $\mathcal{K} \propto \mathcal{K}^2$, however even in the general case they have positive correlation.

In order to understand the correlation more quantitatively, we will focus on the case of mean-squared error (MSE) loss in the rest of this work. Here $\dot{\mathcal{L}}$ corresponds to the diagonal matrix of the

residuals $y - f(x)$. If the outputs of the network is 0 on the training data, then $\dot{\mathcal{L}} = Y \equiv \text{diag}(y)$, the labels themselves. In that case, the correlation of the time derivatives at initialization is a function of the input-label covariances as well as the PTK matrix:

$$\begin{aligned} \rho\left(\dot{W}^\top \dot{W}, \dot{W}^\top K \dot{W}\right) &= \text{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right) \\ &\cdot \text{tr}\left((X^\top Y \mathcal{K} Y X)^2\right)^{-1/2} \cdot \text{tr}\left((X^\top Y \mathcal{K}^2 Y X)^2\right)^{-1/2} \end{aligned} \tag{4}$$

In general we expect that these quantities can self-average under the appropriate high-dimensional limits. It is helpful then to write down the average (over initializations) of the covariance. Taking expectation of the first factor,

$$\begin{aligned} &\mathbb{E}\left[\text{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right)\right] = \\ &\text{tr}\left(X^\top Y \mathbb{E}\left[\mathcal{K}\right] Y X X^\top Y \mathbb{E}\left[\mathcal{K}^2\right] Y X\right) \\ &+ \text{tr}\left(\text{Cov}\left(X^\top Y \mathcal{K} Y X, X^\top Y \mathcal{K}^2 Y X\right)\right) \end{aligned} \tag{5}$$

Similar decompositions exist for the denominator terms.

The simplest limit is the NTK regime where the width tends to infinity with fixed number of data points and input dimension. Here the PTK matrix will approach its expectation, and the second, covariance term above will tend to 0. This suggests that the first (mean) term encodes much of the interesting phenomenology near initialization for large networks. We use this fact to design a dataset that interpolates between small and large initial derivatives of the C-NFA, even in the non-NTK regime, which will help us demonstrate the validity of our random matrix theory approach.

We focus most of our random matrix theory analysis on a one-hidden layer network, as the calculations quickly become complicated with depth. However, we also show that the mean term can be used to approximately predict the NFA value for Gaussian data with different spectral decay rates for more complicated networks (see Appendix C).

3.2. Exact predictions with one hidden layer, quadratic activation

We can capture the behavior of both the mean and covariance terms from Equation (5) in certain high dimensional settings. In particular for one hidden layer neural networks with quadratic activations, we can exactly predict the value of Equation (5) in the high dimensional limit. We make the assumption that X and Y are (asymptotically) freely independent of the parameters at initialization, and that the resulting average is close to the value of any specific network initialization (self-averaging). For more details on the assumptions, see Appendix E.

To illustrate our analysis, we derive here the numerator of Equation (4). Let $M^{(4)}_{X|Y} = (X^\top Y X)^2$, and $M^{(2)}_X = X^\top X$, and $F_a = W^\top \text{diag}\left(a^2\right) W$. Then,

$$\begin{aligned} &\text{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right) \\ &= \text{tr}\left(M^{(4)}_{X|Y} F_a M^{(4)}_{X|Y} F_a M^{(2)}_X F_a\right) . \end{aligned}$$

We can simplify this alternating product using standard results from random matrix theory, and factor the parameter and data contributions at the cost of increasing the number of terms in the expression (Appendix E).

In the simplified case of isotropic data $X^\top X = I$, we have:

$$\begin{aligned} &\text{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right) \\ &= \text{tr}\left(M^{(4)}_{X|Y}\right)^2 \text{tr}\left(F_a^3\right) + \text{tr}\left(F_a\right) \text{tr}\left(F_a^2\right) \text{tr}\left((\bar{M}^{(4)}_{X|Y})^2\right) . \end{aligned} \tag{6}$$

From these random matrix calculations, we see that the correlations of the derivatives are determined by traces of powers of $M^{(4)}_{X|Y}$ (and $M^{(2)}_X$ by the calculations in Section E), which is specific to the dataset, and F_a, which is specific to the architecture and initialization.

Manipulating the C-NFA To numerically explore the validity of the random matrix theory calculations, we need a way to generate datasets with different values of $\rho\left(\dot{W}^\top \dot{W}, \dot{W}^\top K \dot{W}\right)$. We construct a random dataset called the *alignment reversing* dataset, parameterized by a *balance* parameter $\gamma \in (0, 1]$ to adversarially disrupt the NFA near initialization in the regime that width k, input dimension d, and dataset size n are all equal ($n = k = d = 1024$). By Proposition 10, for the aforementioned neural architecture, the expected second derivative of the centered NFM satisfies, $\mathbb{E}\left[\dot{W}^\top \dot{W}\right] = X^\top Y \mathbb{E}\left[\mathcal{K}\right] Y X = (X^\top Y X)^2$, while the expected second derivative of the centered AGOP, $\mathbb{E}\left[\dot{W}^\top K \dot{W}\right] = X^\top Y \mathbb{E}\left[\mathcal{K}^2\right] Y X$, has an additional component $X^\top Y X \cdot X^\top X \cdot X^\top Y X$. Our construction exploits this difference in that $X^\top X$ becomes adversarially unaligned to $X^\top Y X$ as the balance parameter decreases. In our experiment, we sample multiple random datasets with this construction and compute the predicted and observed correlation of the second derivatives of the centered NFA at initialization.

The construction exploits that we can manipulate $X^\top Y X \cdot X^\top X \cdot X^\top Y X$ freely of the NFM using a certain choice of Y (see Appendix F for details of the construction). We design the dataset such that this AGOP-unique term is close to identity, while the NFM second derivative has many large off-diagonal entries, leading to low correlation between the second derivatives of the NFM and AGOP.

The centered NFA correlations predicted with random matrix theory closely match the observed values (Figure 3), across both individual random seeds as well as the average values across them. Crucially, a single neural network is used across the datasets, confirming the validity of the self-averaging assumption. The variation in the plot across seeds come from randomness in the sample of the data, which cause deviations from the adversarial construction.

4. Acknowledgements

We thank Lechao Xiao for detailed feedback on the manuscript. We also thank Jeffrey Pennington for helpful discussions.

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. In *Conference on Learning Theory*, 2022.

[2] Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation. *arXiv preprint arXiv:2205.01445*, 2022.

[3] Boaz Barak, Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *arXiv preprint arXiv:2207.08799*, 2022.

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

[5] 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.

[6] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pages 5413–5452. PMLR, 2022.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[9] Behrad Moniri, Donghwan Lee, Hamed Hassani, and Edgar Dobriban. A theory of non-linear feature learning with one gradient step in two-layer neural networks. *arXiv preprint arXiv:2310.07891*, 2023.

[10] Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu. Neural networks efficiently learn low-dimensional representations with sgd. *arXiv preprint arXiv:2209.14863*, 2022.

[11] Eshaan Nichani, Alex Damian, and Jason D Lee. Provable guarantees for nonlinear feature learning in three-layer neural networks. *arXiv preprint arXiv:2305.06986*, 2023.

[12] Suzanna Parkinson, Greg Ongie, and Rebecca Willett. Linear neural network layers promote learning single-and multiple-index models. *arXiv preprint arXiv:2305.15598*, 2023.

[13] 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.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Gan Yuan, Mingyue Xu, Samory Kpotufe, and Daniel Hsu. Efficient estimation of the central mean subspace via smoothed gradient outer products. *arXiv preprint arXiv:2312.15469*, 2023.

[16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 2021.

Appendix A. Omitted proofs of propositions in the main text

Proof [Proof of Proposition 4] At initialization, we have, $\dot{W} = \frac{\partial f(X)}{\partial h_1}^\top \dot{\mathcal{L}} X$. Therefore, using that $\mathcal{K} \equiv \frac{\partial f(X)}{\partial h_1} \frac{\partial f(X)}{\partial h_1}^\top$,

$$\dot{W}^\top \dot{W} = X^\top \dot{\mathcal{L}} \frac{\partial f(X)}{\partial h_1} \frac{\partial f(X)}{\partial h_1}^\top \dot{\mathcal{L}} X = X^\top \dot{\mathcal{L}} \mathcal{K} \dot{\mathcal{L}} X \, ,$$

and,

$$\begin{aligned} \dot{W}^\top K \dot{W} &= X^\top \dot{\mathcal{L}} \frac{\partial f(X)}{\partial h_1} K \frac{\partial f(X)}{\partial h_1}^\top \dot{\mathcal{L}} X \\ &= X^\top \dot{\mathcal{L}} \frac{\partial f(X)}{\partial h_1} \frac{\partial f(X)}{\partial h_1}^\top \frac{\partial f(X)}{\partial h_1} \frac{\partial f(X)}{\partial h_1}^\top \dot{\mathcal{L}} X \\ &= X^\top \dot{\mathcal{L}} \mathcal{K}^2 \dot{\mathcal{L}} X \, . \end{aligned}$$

■

Appendix B. Additional centerings of the NFA

Double-centered NFA One may additionally center the PTK feature map to understand the co-evolution of the PTK feature covariance and the weight matrices. In this work, we consider such a centering that we refer to as the *double-centered* NFA.

Ansatz 5 (Double-centered NFA)

$$(\bar{W}^{(\ell)})^\top \bar{W}^{(\ell)} \propto (\bar{W}^{(\ell)})^\top \bar{K}^{(\ell)} \bar{W}^{(\ell)} \, ,$$

where $\bar{K}^{(\ell)} = \left(\frac{\partial f(X)}{\partial h_\ell} - \frac{\partial f_0(X)}{\partial h_\ell} \right)^\top \left(\frac{\partial f(X)}{\partial h_\ell} - \frac{\partial f_0(X)}{\partial h_\ell} \right)$*, and* f_0 *is the neural network at initialization.*

However, the double-centered NFA term corresponds to higher-order dynamics that do not significantly contribute the centered and uncentered NFA (Figure 4) when initialization is large or for early periods of training. Note however this term becomes relevant over longer periods of training.

Isolating alignment of the PTK to the initial weight matrix One may also center just the PTK feature map, while substituting the initial weights for W to isolate how the PTK feature covariance aligns to the weight matrices. To measure this alignment, we consider the *PTK-centered* NFA.

Ansatz 6 (PTK-centered NFA)

$$(W_0^{(\ell)})^\top W_0^{(\ell)} \propto (W_0^{(\ell)})^\top \bar{K}^{(\ell)} W_0^{(\ell)} \, ,$$

where $W_0^{(\ell)}$ *is the initial weight matrix at layer* ℓ*.*

However, this correlation decreases through training, indicating that the correlation of these quantities does not drive alignment between the uncentered NFM and AGOP (Figure 5).

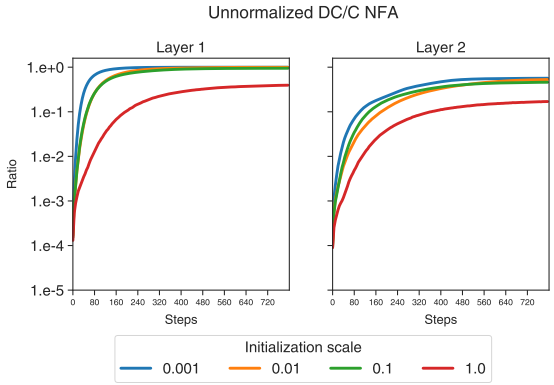


Figure 4: Ratio of the unnormalized double-centered NFA correlation to the centered NFA correlation throughout neural network training. In particular, we plot $\operatorname{tr}\left(\bar{W}^\top \bar{W} \bar{W}^\top \bar{K} \bar{W}\right) \cdot \operatorname{tr}\left(\bar{W}^\top \bar{W} \bar{W}^\top K \bar{W}\right)^{-1}$ throughout training for both layers of a two-hidden layer MLP with ReLU activations.

Appendix C. Extending our theoretical predictions to depth and general activations

Precise predictions of the C-NFA become more complicated with additional depth and general activation functions. However, we note that the deep C-NFA will remain sensitive to a first-order approximation in which K is replaced by its expectation. We demonstrate that this term qualitatively captures the behavior of the C-NFA for 2 hidden layer architectures with quadratic and, to a lesser extent, ReLU activation functions in Figure 6. In this experiment, we sample Gaussian data with mean 0 and covariance with a random eigenbasis. We parameterize the eigenvalue decay of the covariance matrix by a parameter α, called the data decay rate, so that the eigenvalues have values $\lambda_k = \frac{1}{1+k^\alpha}$. As α approaches 0 or ∞ the data covariance approaches a projector matrix.

In this experiment, we see that the data covariance spectrum will also parameterize the eigenvalue decay of $\mathbb{E}[K]$, allowing us to vary how close the expected PTK matrix (and its dual, the PTK feature covariance) is from a projector, where the NFA holds exactly. We see that for intermediate values of α, both the observed and the predicted derivatives of the C-NFA decreases in value.

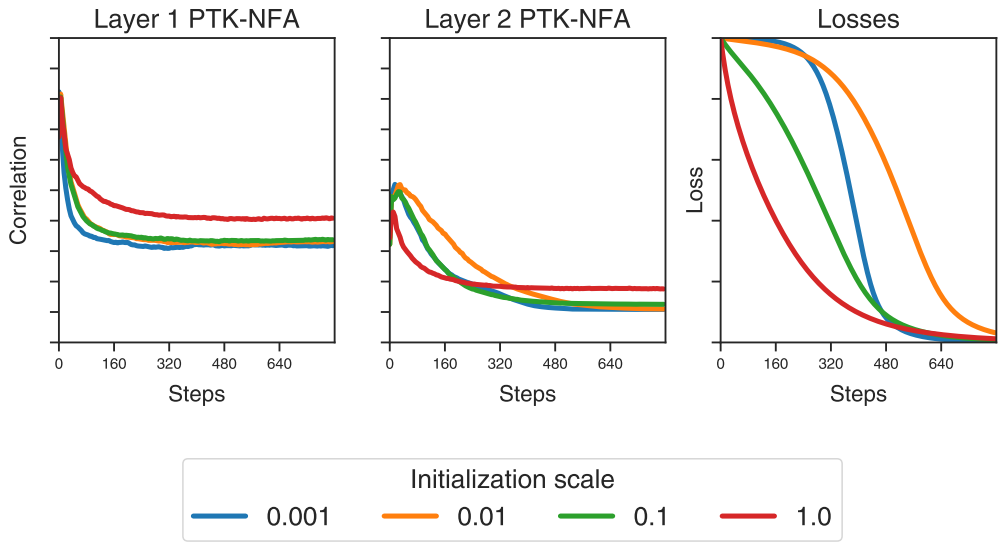
We plot the observed values in two settings corresponding to different asymptotic regimes. One setting is the proportional regime, where $n = k = d = 128$. The other is the NTK regime where $n = d = 128$ and $k = 1024$. For the quadratic case, as the network approaches infinite width, the prediction more closely matches the observed values. Additional terms corresponding to the nonlinear part of ϕ' in ReLU networks, the derivative of the activation function, are required to capture the correlation more accurately in this case.

Appendix D. Proofs and derivations

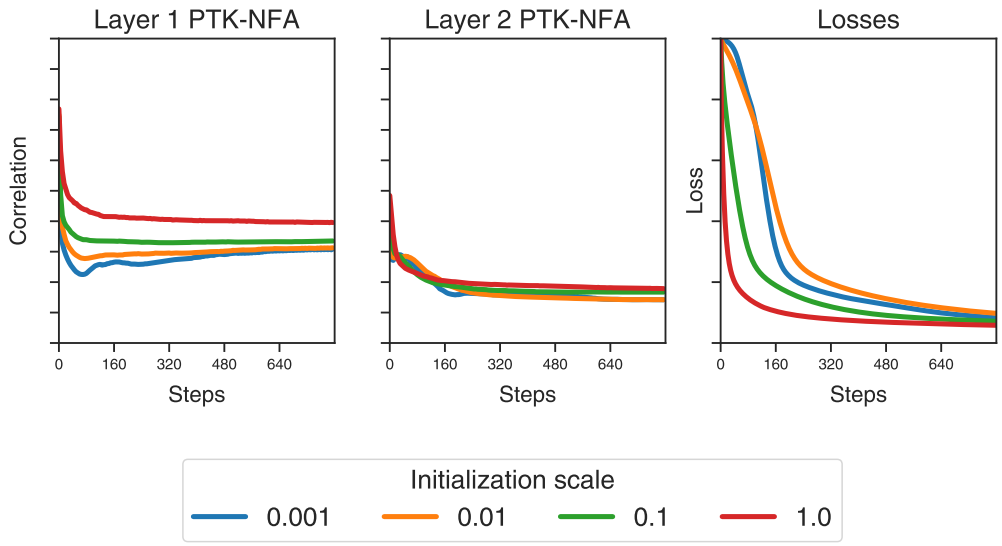
Proof [Proof of Proposition 3] Consider the second derivatives of the NFM and the AGOP.

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top \bar{W}) = \dot{W}^\top \bar{W} + \bar{W}^\top \dot{W} \,.$$

Then,

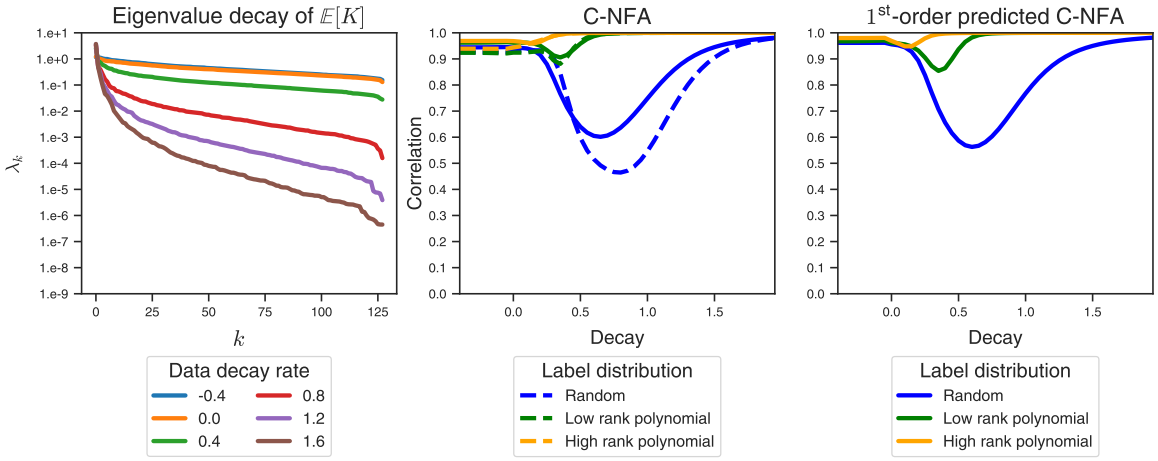


(*a*) Isotropic data.

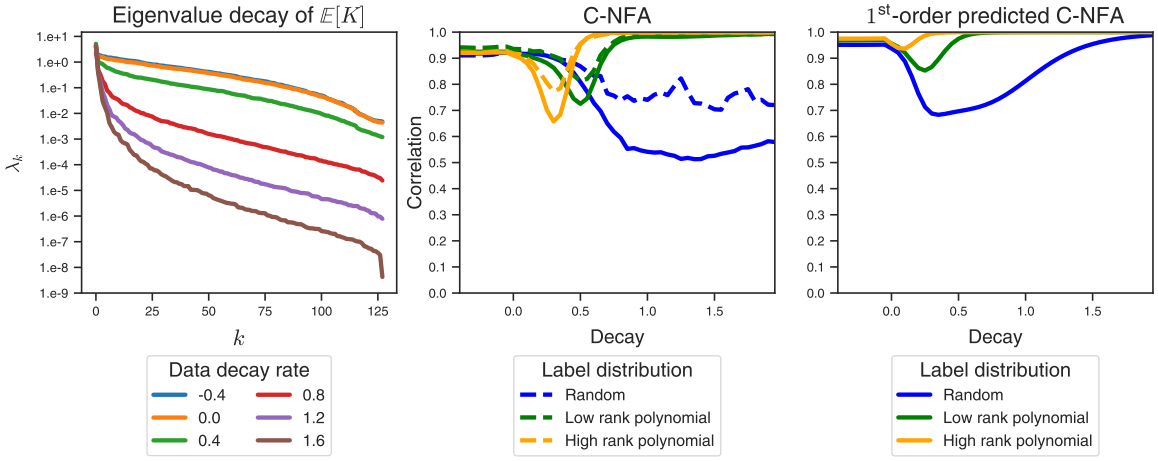


(*b*) Data spectrum with decay $\lambda_k \sim \frac{1}{1+k^2}$.

Figure 5: PTK-centered NFA correlation throughout training for both layers of a two-hidden layer MLP with ReLU activations on Gaussian data with two different spectra.



(a) Quadratic.



(b) ReLU.

Figure 6: Observed versus the first-order predicted C-NFA for the input to the first layer of a two hidden layer MLP. The dashed line is neural network width $k = n = d = 128$, where n and d are the number of data point and data dimension, respectively, while the solid line uses $n = d = 128$ and $k = 1024$.

$$\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top \bar{W}) = 2\dot{W}^\top \dot{W} + \ddot{W}^\top \bar{W} + \bar{W}^\top \ddot{W}\,.$$

At initialization $\bar{W} = 0$, therefore,

$$\bar{W}^\top \bar{W} = 0, \quad \frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top \bar{W}) = 0, \quad \frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top \bar{W}) = 2\dot{W}^\top \dot{W}\,. \tag{7}$$

Meanwhile, for the (centered) AGOP,

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top K\bar{W}) = \dot{W}^\top K\bar{W} + \bar{W}^\top \dot{K}_z\bar{W} + \bar{W}^\top K\dot{W} \,.$$

Then,

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top K\bar{W}) = \ddot{W}^\top K\bar{W} + \dot{W}^\top \dot{K}\bar{W} + \dot{W}^\top K_z\dot{W} \\ + \dot{W}^\top K\dot{W} + \bar{W}^\top \dot{K}\dot{W} + \bar{W}^\top K\ddot{W} \\ + \dot{W}^\top \dot{K}\bar{W} + \bar{W}^\top \ddot{K}\bar{W} + \bar{W}^\top \dot{K}\dot{W} \,.\end{aligned}$$

At initialization,

$$\bar{W}^\top K\bar{W} = 0, \quad \frac{\mathrm{d}}{\mathrm{d}\,t}(\bar{W}^\top K\bar{W}) = 0, \tag{8}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(\bar{W}^\top K\bar{W}) = 2\dot{W}^\top K\dot{W} \,. \tag{9}$$

Therefore, the correlation between the centered neural feature matrix and the centered AGOP at initialization is determined by,

$$\rho\left(\dot{W}^\top\dot{W}, \dot{W}^\top K\dot{W}\right) \,.$$

Both sides simplify at initialization:

$$\dot{W}^\top\dot{W} = X^\top YKYX, \quad \dot{W}^\top K\dot{W} = X^\top YK^2YX \,,$$

where K is its value at initialization. ■

Appendix E. Free probability calculations of C-NFA

In order to understand the development of the NFA, we analyze the centered NFA in the limit that learning rate is much smaller than the initialization for a one hidden layer MLP with quadratic activations. We write this particular network as,

$$f(x) = a^\top (Wx)^2 \,,$$

where $a \in \mathbb{R}^{1\times k}$ and $W \in \mathbb{R}^{k\times d}$, where d is the input dimension and k is the width. In this case, the NFA has the following form,

$$\rho\left(F, \bar{G}\right) = \frac{\mathrm{tr}\left(X^\top Y\mathcal{K}YXX^\top Y\mathcal{K}^2YX\right)}{\mathrm{tr}\left((X^\top Y\mathcal{K}YX)^2\right)^{-1/2}\mathrm{tr}\left((X^\top Y\mathcal{K}^2YX)^2\right)^{-1/2}} \,, \tag{10}$$

where $\mathcal{K} = XW^\top \mathrm{diag}\,(a)^2\, WX^\top$.

We assume three properties hold in the finite dimensional case we consider, that will hold asymptotically in the infinite dimensional limit.

Assumption 7 (Self-averaging) *We assume that computing the average of the NFA quantities across initializations is equal to the quantities themselves in the high-dimensional limit.*

Assumption 8 (Asymptotic freeness) *We assume that the collections $\{X, Y\}$ and $\{W, a\}$ are asymptotically free with respect to the operator $\mathbb{E}\left[\mathrm{tr}(\cdot)\right]$, where $\mathrm{tr}[M] = \frac{1}{n}\sum_{i=1}^{n} M_{ii}$.*

Assumption 9 (Commutativity of expectation) *We will also make the approximation that the expectation commutes with ratio and square root.*

$$\mathbb{E}\left[\rho(A,B)\right] = \mathbb{E}\left[\frac{\mathrm{tr}(A^\top B)}{\sqrt{\mathrm{tr}(A^\top A)\mathrm{tr}(B^\top B)}}\right] \approx \frac{\mathbb{E}\left[\mathrm{tr}(A^\top B)\right]}{\sqrt{\mathbb{E}\left[\mathrm{tr}(A^\top A)\right]\mathrm{E}[\mathrm{tr}(B^\top B)]}} \tag{11}$$

We will compute the expected values of the centered NFA under these assumptions. In the remainder of the section we will drop the $\mathbb{E}\left[\cdot\right]$ in the trace for ease of notation.

E.1. Free probability identities

The following lemmas will be useful: let $\{\bar{C}_i\}$ and $\{R_i\}$ be freely independent of each other with respect to tr, with $\mathrm{tr}[\bar{C}_i] = 0$. Alternating words have the following products:

$$\mathrm{tr}[\bar{C}_1 R_1] = 0 \tag{12}$$

$$\mathrm{tr}[\bar{C}_1 R_1 \bar{C}_2 R_2] = \mathrm{tr}[R_1]\mathrm{tr}[R_2]\mathrm{tr}[\bar{C}_1\bar{C}_2] \tag{13}$$

$$\mathrm{tr}[\bar{C}_1 R_1 \bar{C}_2 R_2 \bar{C}_3 R_3] = \mathrm{tr}[R_1]\mathrm{tr}[R_2]\mathrm{tr}[R_3]\mathrm{tr}[\bar{C}_1\bar{C}_2\bar{C}_3] \tag{14}$$

$$\begin{aligned}\mathrm{tr}[\bar{C}_1 R_1 \bar{C}_2 R_2 \bar{C}_3 R_3 \bar{C}_4 R_4] &= \mathrm{tr}[R_1]\mathrm{tr}[R_2]\mathrm{tr}[R_3]\mathrm{tr}[R_4]\mathrm{tr}[\bar{C}_1\bar{C}_2\bar{C}_3\bar{C}_4] + \\ &\mathrm{tr}[R_1]\mathrm{tr}[R_3]\mathrm{tr}[\bar{R}_2\bar{R}_4]\mathrm{tr}[\bar{C}_1\bar{C}_2]\mathrm{tr}[\bar{C}_3\bar{C}_4] + \mathrm{tr}[R_2]\mathrm{tr}[R_4]\mathrm{tr}[\bar{R}_1\bar{R}_3]\mathrm{tr}[\bar{C}_2\bar{C}_3]\mathrm{tr}[\bar{C}_1\bar{C}_4]\end{aligned} \tag{15}$$

where $\bar{R}_i \equiv R_i - \mathrm{tr}[R_i]$.

Applying these identities to the one hidden layer quadratic case, we use the following definitions:

$$R = W^\top \mathrm{diag}\left(a^2\right) W,\ A = (X^\top Y X)^2,\ B = X^\top X \tag{16}$$

Crucially, R is freely independent of the set $\{A, B\}$. We will also use the notation $\bar{M}$ to indicated the centered version of M, $\bar{M} = M - \mathrm{tr}[M]$.

E.2. Numerator term of NFA

The numerator in Equation (10) is

$$\mathrm{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right) = \mathrm{tr}\left(ARARBR\right) \tag{17}$$

Re-writing $A = \bar{A} + \mathrm{tr}[A]$ and $B = \bar{B} + \mathrm{tr}[B]$ we have:

$$\mathrm{tr}\left(X^\top Y \mathcal{K} Y X X^\top Y \mathcal{K}^2 Y X\right) = \mathrm{tr}\left((\bar{A} + \mathrm{tr}[A])R(\bar{A} + \mathrm{tr}[A])R(\bar{B} + \mathrm{tr}[B])R\right) \tag{18}$$

This expands to

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}\left(\bar{A}R\bar{A}R\bar{B}R\right) + 2\text{tr}\left(A\right)\text{tr}\left(\bar{A}R\bar{B}R^2\right) + \text{tr}\left(B\right)\text{tr}\left(\bar{A}R\bar{A}R^2\right) \\ \text{tr}\left(A\right)^2\text{tr}\left(\bar{B}R^3\right) + 2\text{tr}\left(A\right)\text{tr}\left(B\right)\text{tr}\left(\bar{A}R^3\right) + \text{tr}\left(A\right)^2\text{tr}\left(B\right)\text{tr}\left(R\right)^3\end{aligned} \tag{19}$$

Using the identities we arrive at:

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}[A]^2\text{tr}[B]\text{tr}\left(R^3\right) \\ + 2\text{tr}[A]\text{tr}[R^2]\text{tr}[R]\text{tr}\left(\bar{A}\bar{B}\right) \\ + \text{tr}[B]\text{tr}[R]\text{tr}[R^2]\text{tr}\left(\bar{A}^2\right) \\ + \text{tr}[R]^3\text{tr}\left(\bar{A}^2\bar{B}\right)\end{aligned} \tag{20}$$

E.3. First denominator term of NFA

The first denominator term in Equation (10) is

$$\text{tr}\left(X^\top Y\mathcal{K}YXX^\top Y\mathcal{K}YX\right) = \text{tr}\left(ARAR\right) \tag{21}$$

This is a classic free probability product:

$$\text{tr}\left(X^\top Y\mathcal{K}YXX^\top Y\mathcal{K}YX\right) = \text{tr}[A^2]\text{tr}[R]^2 + \text{tr}[A]^2\text{tr}\left(R^2\right) - \text{tr}[A]^2\text{tr}[R]^2 \tag{22}$$

which can be derived from the lemmas.

E.4. Second denominator term of NFA

For the second denominator term of Equation (10) we have

$$\text{tr}\left(X^\top Y\mathcal{K}^2YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}\left(ARBRARBR\right) \tag{23}$$

Expanding the first A we have

$$\text{tr}\left(X^\top Y\mathcal{K}^2YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}\left(\bar{A}RBRARBR\right) + \text{tr}[A]\text{tr}\left(R^2BRARB\right) \tag{24}$$

Next we expand the first B:

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}^2YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}\left(\bar{A}R\bar{B}RARBR\right) + \text{tr}[B]\text{tr}\left(\bar{A}R^2ARBR\right) \\ + \text{tr}[A]\text{tr}[B]\text{tr}\left(R^3ARB\right) + \text{tr}[A]\text{tr}\left(R^2\bar{B}RARB\right)\end{aligned} \tag{25}$$

The next A gives us

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}^2YXX^\top Y\mathcal{K}^2YX\right) = \text{tr}\left(\bar{A}R\bar{B}R\bar{A}RBR\right) + 2\text{tr}[A]\text{tr}\left(\bar{A}R\bar{B}R^2BR\right) + \text{tr}[B]\text{tr}\left(\bar{A}R^2\bar{A}RBR\right) \\ + 2\text{tr}[A]\text{tr}[B]\text{tr}\left(R^3\bar{A}RB\right) + \text{tr}[A]^2\text{tr}[B]\text{tr}\left(R^4B\right) + \text{tr}[A]^2\text{tr}\left(R^2\bar{B}R^2B\right)\end{aligned} \tag{26}$$

Expanding the final B we have

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}^2 YXX^\top Y\mathcal{K}^2 YX\right) &= \text{tr}\left(\bar{A}R\bar{B}R\bar{A}R\bar{B}R\right) + 2\text{tr}[B]\text{tr}\left(\bar{A}R\bar{B}R\bar{A}R^2\right) + 2\text{tr}[A]\text{tr}\left(\bar{A}R\bar{B}R^2\bar{B}R\right) \\ &+ 4\text{tr}[A]\text{tr}[B]\text{tr}\left(R^3\bar{A}R\bar{B}\right) + 2\text{tr}[A]\text{tr}[B]^2\text{tr}\left(R^4\bar{A}\right) + 2\text{tr}[A]^2\text{tr}[B]\text{tr}\left(R^4\bar{B}\right) \\ &+ \text{tr}[A]^2\text{tr}[B]^2\text{tr}[R^4] + \text{tr}[A]^2\text{tr}[R^2\bar{B}R^2\bar{B}] + \text{tr}[B]^2\text{tr}[R^2\bar{A}R^2\bar{A}]\end{aligned} \tag{27}$$

Now all terms are in the form of alternating products from the lemma. This means we can factor out the non-zero traces of the other terms. Simplifying we have:

$$\begin{aligned}\text{tr}\left(X^\top Y\mathcal{K}^2 YXX^\top Y\mathcal{K}^2 YX\right) &= \text{tr}[R]^4\text{tr}\left((\bar{A}\bar{B})^2\right) + 2\text{tr}[R]^2(\text{tr}[R^2] - \text{tr}[R]^2)\text{tr}[\bar{A}\bar{B}]^2 \\ &+ 2\text{tr}[R]^2\text{tr}[R^2]\left(\text{tr}[B]\text{tr}\left(\bar{A}^2\bar{B}\right) + \text{tr}[A]\text{tr}\left(\bar{A}\bar{B}^2\right)\right) \\ &+ 4\text{tr}[A]\text{tr}[B]\text{tr}[R^3]\text{tr}[R]\text{tr}\left(\bar{A}\bar{B}\right) + \text{tr}[A]^2\text{tr}[B]^2\text{tr}[R^4] \\ &+ \text{tr}[A]^2\text{tr}[\bar{B}^2]\text{tr}[R^2]^2 + \text{tr}[\bar{A}]^2\text{tr}[B^2]\text{tr}[R^2]^2\end{aligned} \tag{28}$$

All terms of the NFA are now in terms of traces of the matrices A, B, and R and functions on each term separately. The matrices A and B are determined by the data, while the moments of the eigenvalues of R are determined by the initialization distribution of the weights in the neural network, and neither training nor the data.

Appendix F. Alignment reversing dataset

The data consists of a mixture of two distributions from which two subsets of the data X_1 and X_2 are sampled from, and is parametrized by a balance parameter $\gamma \in (0, 1]$ and two variance parameters $\epsilon_1, \epsilon_2 > 0$. The subset X_1 which has label $y_1 = 1$ and constitutes a γ fraction of the entire dataset, is sampled from a multivariate Gaussian distribution with mean 0 and covariance

$$\Sigma = \mathbf{1}\mathbf{1}^\top + \epsilon_1 \cdot I\ .$$

Then the second subset, X_2, is constructed such that $X_2^\top X_2 \approx (X_1^\top X_1)^{-2}$, and has labels $y_2 = 0$. Then, for balance parameter γ sufficiently small, the AGOP second derivative approximately satisfies,

$$\begin{aligned}\mathbb{E}\left[\dot{W}^\top K\dot{W}\right] &\sim X^\top YXX^\top XX^\top YX \\ &= X_1^\top X_1 X^\top X X_1^\top X_1 \\ &\approx X_1^\top X_1 X_2^\top X_2 X_1^\top X_1 \\ &\approx I\ ,\end{aligned}$$

In contrast, the NFM second derivative, $\mathbb{E}\left[\dot{W}^\top\dot{W}\right] = (X^\top YX)^2 = (X_1^\top X_1)^2 \approx \Sigma^2$, will be significantly far from identity.

Motivated by this derivation, we construct X_2 by the following procedure:

1. Extract singular values S_1 and right singular vectors U_1 from a singular-value decomposition (SVD) of $X_1^\top X_1$.

2. Extract the left singular vectors V_2 from a sample $\tilde{X}_2$ that is sampled from the same distribution as X_1.
3. Construct $X_2 = V_2 S_1^{-1} U_1^\top$.
4. Where $X = X_1 \oplus X_2$, Set $X \leftarrow X + \epsilon_2 Z$, where $Z \sim \mathcal{N}(0, I)$.
5. Set $y \leftarrow y + 10^{-5} \cdot \mathbf{1}$.

Note that $U_1 S_1^{-1} V_2^\top V_2 S_1^{-1} U_1^\top = U_1 S_1^{-2} U_1^\top = (X_1^\top X_1)^{-2}$, therefore, we should set $X_2 = V_2 S_1^{-1} U_1^\top$ to get $X_2^\top X_2 = (X_1^\top X_1)^{-2}$. Regarding the variance parameters, in practice we set $\epsilon_1 = 0.5$ and $\epsilon_2 = 10^{-2}$.

Proposition 10 (Expected NFM and AGOP) *For a one hidden layer quadratic network, $f(x) = a^\top (Wx)^2$, with $a \sim \mathcal{N}(0, I)$ and $W \sim \frac{1}{\sqrt{k}} \cdot \mathcal{N}(0, I)$,*

$$\mathbb{E}_{a,W}\left[\dot{W}^\top \dot{W}\right] = (X^\top Y X)^2 \,,$$

and,

$$\begin{aligned}\mathbb{E}_{a,W}\left[\dot{W}^\top K \dot{W}\right] = 3 \cdot tr\left(X^\top X\right) \cdot (X^\top Y X)^2 \\ + 6 X^\top Y X X^\top X X^\top Y X\end{aligned}$$

Proof [Proof of Proposition 10]

$$\begin{aligned}\mathbb{E}\left[\dot{W}^\top \dot{W}\right] &= X^\top Y X \mathbb{E}\left[W_0^\top \operatorname{diag}(a)^2 W_0\right] X^\top Y X \\ &= (X^\top Y X)^2 \,.\end{aligned}$$

Further,

$$\mathbb{E}\left[\dot{W}^\top K \dot{W}\right] = X^\top Y \mathbb{E}\left[K^2\right] Y X \,.$$

We note that,

$$K^2 = W_0^\top \operatorname{diag}(a)^2 W_0 X^\top X W_0^\top \operatorname{diag}(a)^2 W_0 \tag{29}$$

$$= \sum_{s_1, s_2}^{k} \sum_{\alpha}^{n} \sum_{p_1, p_2}^{d} \tag{30}$$

$$a_{s_1}^2 a_{s_2}^2 W_{s_1, p_1} X_{\alpha, p_1} X_{\alpha, p_2} W_{s_2, p_2} X W_{s_1} W_{s_2}^\top X^\top \,. \tag{31}$$

Therefore, applying Wick's theorem, element i, j of K^2 satisfies,

$$
\begin{aligned}
\mathbb{E}\left[K_{ij}^2\right] &= \sum_{s}^{k}\sum_{\alpha}^{n}\sum_{p_1,p_2}^{d}\mathbb{E}\left[a_s^4 W_{s,p_1}X_{\alpha,p_1}X_{\alpha,p_2}W_{s,p_2}X_i^\top W_s W_s^\top X_j\right] \\
&= \sum_{s}^{k}\sum_{\alpha}^{n}\sum_{p_1,p_2,q_1,q_2}^{d} \\
&\mathbb{E}\left[a_s^4 W_{s,p_1}W_{s,p_2}W_{s,q_1}W_{s,q_2}X_{\alpha,p_1}X_{\alpha,p_2}X_{i,q_1}X_{j,q_2}\right] \\
&= 3\sum_{s}^{k}\sum_{\alpha}^{n}\sum_{p_1,p_2,q_1,q_2}^{d} \\
&\Big(\mathbb{E}\left[W_{s,p_1}W_{s,p_2}\right]\mathbb{E}\left[W_{s,q_1}W_{s,q_2}\right] + \\
&\mathbb{E}\left[W_{s,p_1}W_{s,q_1}\right]\mathbb{E}\left[W_{s,p_2}W_{s,p_2}\right] + \\
&\mathbb{E}\left[W_{s,p_1}W_{s,q_2}\right]\mathbb{E}\left[W_{s,p_2}W_{s,q_1}\right]\Big) \\
&\cdot X_{\alpha,p_1}X_{\alpha,p_2}X_{i,q_1}X_{j,q_2} \\
&= 3\sum_{\alpha}^{n}\sum_{p_1,p_2,q_1,q_2}^{d} \\
&\Big(\delta_{p_1p_2}\delta_{q_1q_2} + \delta_{p_1q_1}\delta_{p_2q_2} + \delta_{p_1q_2}\delta_{p_2q_1}\Big) \\
&\cdot X_{\alpha,p_1}X_{\alpha,p_2}X_{i,q_1}X_{j,q_2} \\
&= 3\sum_{\alpha}^{n}\Big(\sum_{p_1,q_1}^{d} X_{\alpha,p_1}X_{\alpha,p_1}X_{i,q_1}X_{j,q_1} \\
&+ \sum_{p_1,p_2}^{d} X_{\alpha,p_1}X_{\alpha,p_2}X_{i,p_1}X_{j,p_2} \\
&+ \sum_{p_1,p_2}^{d} X_{\alpha,p_1}X_{\alpha,p_2}X_{i,p_2}X_{j,p_1}\Big) \\
&= 3\cdot\mathrm{tr}\left(X^\top X\right)\cdot X_i^\top X_j + 3\sum_{\alpha}^{n} X_\alpha^\top X_i X_\alpha^\top X_j \\
&+ 3\sum_{\alpha}^{n} X_\alpha^\top X_j X_\alpha^\top X_i \\
&= 3\cdot\mathrm{tr}\left(X^\top X\right)\cdot X_i^\top X_j + 3X_i X^\top X X_j + 3X_j X^\top X X_i\,.
\end{aligned}
$$

Finally, we conclude,

$$
\mathbb{E}\left[K^2\right] = 3\left(\mathrm{tr}\left(X^\top X\right)XX^\top + 2XX^\top XX^\top\right)\,,
$$

giving the second statement of the proposition. ■

Appendix G. Varying the data distribution

We verify that our observations for isotropic Gaussian data hold even when the data covariance has a significant spectral decay. (Figures 7 and 8). We again consider Gaussian data that is mean 0 and where the covariance is constructed from a random eigenbasis. In Figure 7, we substitute the eignevalue decay as $\lambda_k \sim \frac{1}{1+k}$, while in Figure 8, we use $\lambda_k \sim \frac{1}{1+k^2}$. We plot the values of the UC-NFA, C-NFA, train loss, and test loss throughout training for the first and second layer of a two hidden layer network with ReLU activations, while additionally varying initialization scale. Similar to Figure 2, we observe that the C-NFA is more robust to the initialization scale than the UC-NFA, and UC-NFA value become high through training, while being small at initialization. We see that the test loss improves for smaller initializations, where the value of the C-NFA and UC-NFA are higher.

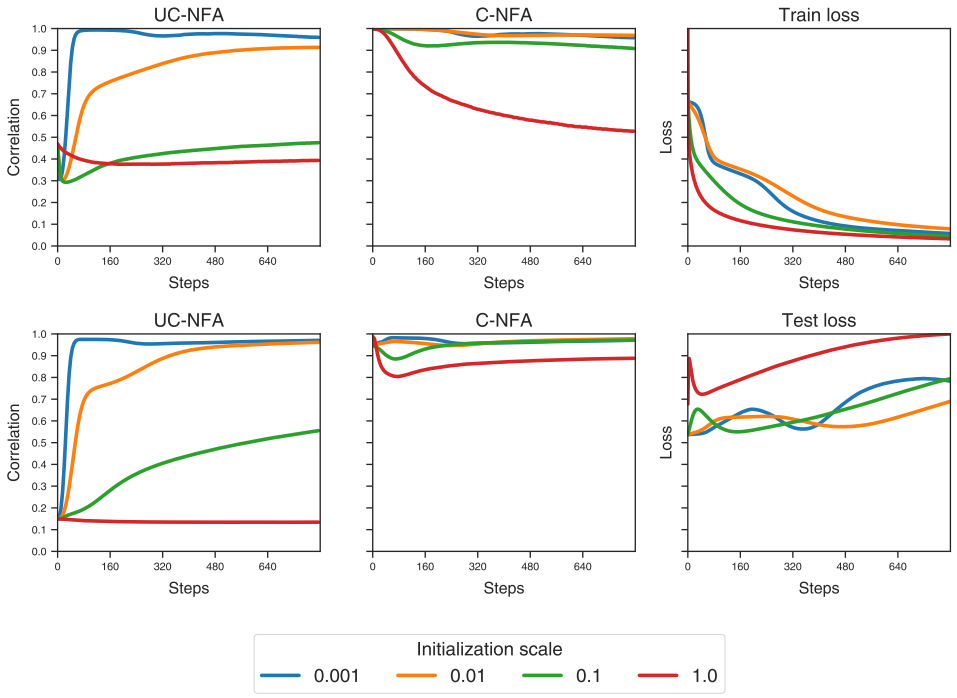


Figure 7: Centered NFA correlations. Data covariance decay rate $\lambda_k \sim \frac{1}{1+k}$. Top row is layer 1, bottom row is layer 2. Train (test) losses are scaled by the maximum train (test) loss achieved so that they are between 0 and 1.

Appendix H. Experimental details

We describe the neural network training and architectural hyperparameters in the experiments of this paper. Biases were not used for any networks. Further, in all polynomial tasks, we scaled the label vector to have standard deviation 1.

Corrupted AGOP For the experiments in Figure 1, we used $n = 384$ data points, $d = 32$, $k = 128$ as the width in all layers, isotropic Gaussian data, initialization scale 0.01 in the first layer and

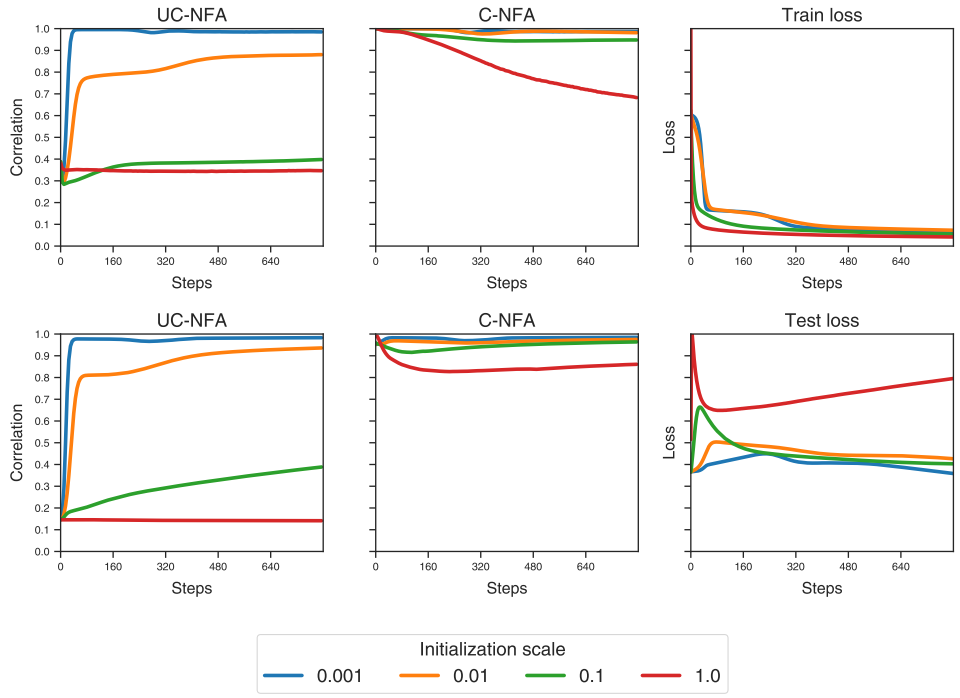


Figure 8: Centered NFA correlations. Data covariance decay rate $\lambda_k \sim \frac{1}{1+k^2}$. Top row is layer 1, bottom row is layer 2. Train (test) losses are scaled by the maximum train (test) loss achieved so that they are between 0 and 1.

default scale in the second. We used ReLU activations and two hidden layers. For the experiments in Figure 2,7,8,4, and 5, we used a two hidden layer network with ReLU activations, learning rate 0.05, 800 steps of gradient decent, and took correlation/covariance measurements every 5 steps.

Alignment reversing dataset For the experiments in Figure 3, we used $k = n = d = 1024$ for the width, dataset size, and input dimension, respectively. Further, the traces of powers of F_a are averaged over 30 neural net seeds to decouple these calculated values from the individual neural net seeds. The mean value plotted in the first two squares of figure is computed over 10 data seeds.

Predictions with depth For the Deep C-NFA predictions (Figure 6), we used $n = 128$, $d = 128$, initialization scale of 1. The low rank task is just the chain monomial of rank $r = 5$. The high rank polynomial task is $y(x) = \sum_{i=1}^{d} (Qx)_i^2$, where $Q \in \mathbb{R}^{d \times d}$ is a matrix with standard normal entries.

Real datasets For the experiments on CelebA, we train a two hidden layer network on a balanced subset of 7500 points with Adam with learning rate 0.0001 and no weight decay. We use initialization scale 0.02 in the first layer, and width 128. We train for 500 epochs. We pre-process the dataset by scaling the pixel values to be between 0 and 1.

Appendix I. Experiments on real datasets

We replicate Figures 1 on celebrity faces (CelebA). We begin by showing that one can disrupt the NFA correspondence by replacing the PTK feature covariance with a random matrix of the same spectral decay. For this example, we measure the Pearson correlation, which subtracts the mean of the image. I.e. $\bar{\rho}(A, B) \equiv \rho\left(A - m(A), B - m(B)\right)$, where $m(A), m(B)$ are the average of the elements of A and B.

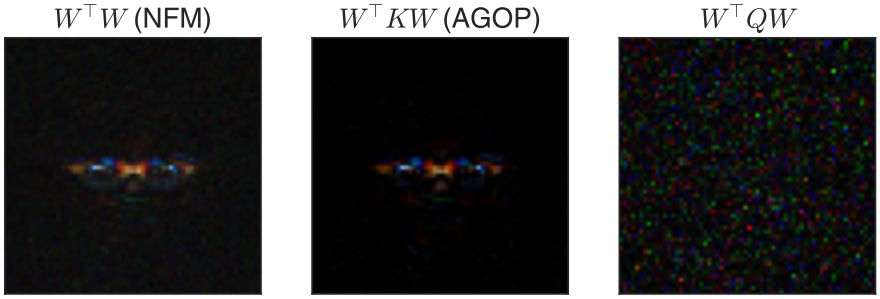


Figure 9: Various feature learning measures for the CelebA binary subtask of predicting glasses. The diagonals of the NFM $\left(W^\top W\right)$ (first plot) and AGOP $\left(W^\top KW\right)$ (second plot) of a fully-connected network are similar to each other. Replacing K with a symmetric matrix Q with the same spectrum but independent eigenvectors obscures the low rank structure (third plot), and reduces the Pearson correlation of the diagonal from $\bar{\rho}\left(\operatorname{diag}\left(F\right), \operatorname{diag}\left(\bar{G}\right)\right) = 0.91$ to $\bar{\rho}\left(\operatorname{diag}\left(F\right), \operatorname{diag}\left(W^\top QW\right)\right) = 0.04$.