Feature Learning in Deep Classifiers through Intermediate Neural Collapse

Akshay Rangamani [*1] **Marius Lindegaard** [*1] **Tomer Galanti** [1] **Tomaso Poggio** [1]

Abstract

In this paper, we conduct an empirical study of the feature learning process in deep classifiers. Recent research has identified a training phenomenon called Neural Collapse (NC), in which the top-layer feature embeddings of samples from the same class tend to concentrate around their means, and the top layer's weights align with those features. Our study aims to investigate if these properties extend to intermediate layers. We empirically study the evolution of the covariance and mean of representations across different layers and show that as we move deeper into a trained neural network, the within-class covariance decreases relative to the between-class covariance. Additionally, we find that in the top layers, where the between-class covariance is dominant, the subspace spanned by the class means aligns with the subspace spanned by the most significant singular vector components of the weight matrix in the corresponding layer. Finally, we discuss the relationship between NC and Associative Memories (Willshaw et al., 1969).

1. Introduction

Deep learning has emerged as a powerful technique for solving various problems in diverse domains such as computer vision (He et al., 2016; Simonyan & Zisserman, 2014), natural language processing (Vaswani et al., 2017; Brown et al., 2020), and decision making in novel environments (Silver et al., 2016). Despite its successes, there remains a significant gap between its empirical performance and our theoretical understanding, even for simple supervised learning problems in classification or regression.

A major line of work (e.g., (Jacot et al., 2018; Du et al., 2019; 2018; Arora et al., 2019; Yang, 2020; Yang & Littwin, 2021; Littwin et al., 2020)) aims to understand neural networks at their infinite-width limit. In this framework, it is shown that infinitely wide neural networks converge to a solution of a kernel least squares problem with a kernel associated with the network's architecture, known as the Neural Tangent Kernel (NTK). While this approach provides valuable insights into the solutions of optimization problems in the "kernel regime", it has limitations when it comes to understanding the representations learned by finite neural networks. For instance, Chen et al. (2020); Allen-Zhu & Li (2019); Malach et al. (2021) identified classes of functions that can be learned efficiently by deep architectures, but not with kernel methods. In (Allen-Zhu & Li, 2020), they characterized a "backward feature correction" process in which features are learned hierarchically by SGD. Additionally, Woodworth et al. (2020) studied how the scale of initialization controls the transition between the "kernel" and "feature learning" regimes.

In a recent study, Papyan et al. (2020) empirically observed that when training overparameterized deep neural networks for Cross-Entropy loss minimization on a given classification task, several structural properties tend to emerge in the last layer. These properties, known as Neural Collapse (NC) list four conditions: (NC1) the features of examples within the same class collapse to their mean, (NC2) class means of the features spread out to form an equiangular tight frame, (NC3) the weights of the classifier converge to the class means of the features, and (NC4) the deep network becomes a nearest class center classifier. These observations raise the question of whether similar phenomena also occur in the intermediate layers of the neural network.

Contributions. In this paper, we investigate whether similar phenomena to Neural Collapse (NC) occur in intermediate layers of deep classifiers. We study the process of feature learning in terms of the first and second-order statistics of representations at different layers. Our results show that when deep networks exhibit NC at the last layer, they also display signs of collapse in intermediate layers. We identify layers where the within-class covariance of representations is dominated by the between-class covariance (NC1) and observe that in these layers, class means to form a simplex ETF (NC2), the subspaces spanned by class means are aligned with the input subspace of linear transformations (NC3), and nearest class center

[*]Equal contribution [1]Center for Brains, Minds, and Machines, Massachusetts Institute of Technology. Correspondence to: Akshay Rangamani <arangam@mit.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

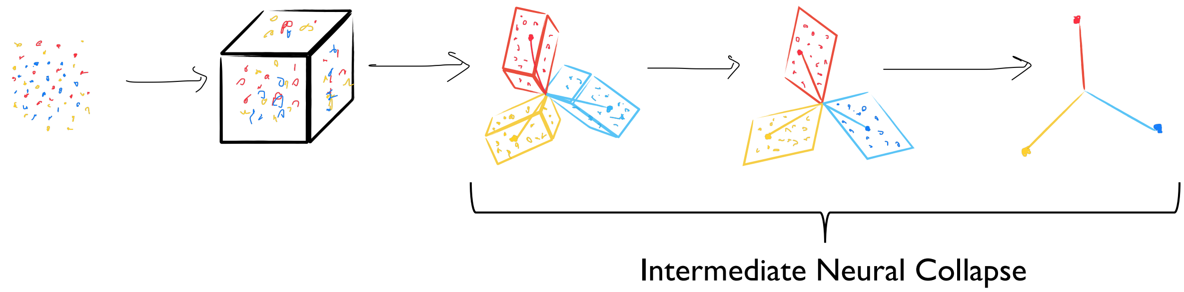

Figure 1. A cartoon illustration of intermediate neural collapse as data is fed through the layers of an overparameterized deep classifier. The classifier first projects the data into a high dimensional space in which linear decision boundaries are found. Intermediate neural collapse in the deeper layers indicates that the role of these layers is to reduce the within class covariance of the representations and increase the between class covariance as a fraction of the total covariance. In this paper we provide definitions and evidence of intermediate neural collapse.

classification using the layer's representations align with the decision of the deep network (NC4). This is the first study to provide a comprehensive description of NC in intermediate layers, and we also measure the rank of weight matrices and covariances of representations to understand how features are transformed in the NC regime. A cartoon illustration of the representations learned by deep classifiers in collapsed layers is given in Fig. 1. The code for running our experiments is available at `https://github.com/mariuslindegaard/Intermediate_Neural_Collapse/tree/ICML2023`

2. Related Work

Neural collapse. The phenomenon of Neural Collapse (NC) was first described in full in (Papyan et al., 2020), although the observation of a certain geometric clustering of features within the same class had been made in earlier papers, such as Goldfeld et al. (2019). Since the initial NC paper, which showed the phenomenon occurring with the cross-entropy loss, there has been a surge of research into theoretical and empirical descriptions of NC. Han et al. (2022) demonstrated NC using the Mean Squared Error (MSE) loss, while papers such as Xu et al. (2023); Ergen & Pilanci (2020) have shown that different optimization algorithms can lead to NC solutions when trained to zero MSE loss. Rangamani & Banburski-Fahey (2022) show that weight decay is necessary for the emergence of NC with the MSE loss. The emergence of NC solutions using cross-entropy was also shown in other papers (Wojtowytsch et al., 2020; Fang et al., 2021; Lu & Steinerberger, 2020). Several papers such as (Zhu et al., 2021; Zhou et al., 2022; Mixon et al., 2020; Tirer & Bruna, 2022; Ji et al., 2021) have also explored the Unconstrained Features Model (UFM), which analyzes the last layer features and classifier as optimization variables. The abstraction of the UFM has provided a simplified model for deriving the emergence of NC theoretically.

Feature learning in deep networks. Since deep networks are organized as hierarchical layers, the structure of the representations learned at intermediate layers has also been an object of study in order to understand how deep networks work. In Recanatesi et al. (2019) and Ansuini et al. (2019), the authors study how different measures of the dimension of intermediate representations progresses through the network. Both papers show that the dimension of the representations first blows up and later reduces as one goes through deeper layers of the classifier. We later show that networks that exhibit neural collapse also show this behavior. Attempts to understand the evolution of deep network representations through the lens of information theory were made by Shwartz-Ziv & Tishby (2017). They described the representations learned by intermediate layers through the mechanism of the information bottleneck. Their observations on the dynamics of the representations and their connections to generalization were later shown to be highly dependent on the architectures and non-linearities used (Saxe et al., 2019), as well as the type of binning (Goldfeld et al., 2019) used in the estimation of mutual information.

Clustering properties of intermediate layers of deep networks. In the literature on feature learning, a particular focus is placed on clustering properties that emerge in intermediate layers of the network, as they indicate that samples can be easily classified at early stages of the network. For instance, in (Alain & Bengio, 2017), it was demonstrated that linear probing of intermediate layers in a trained network becomes more accurate as we move deeper into the network. This finding was also supported in (Cohen et al., 2018), where the authors demonstrated that a k-nearest neighbors classifier using intermediate representations performed well, particularly using the final layer of the deep network.

Following the work of Papyan et al. (2020), several papers (Ben-Shaul & Dekel, 2022; Galanti et al., 2022a; He & Su, 2022) investigated the applicability of the nearest class center (NCC) classification rule (NC4) to intermediate layers in neural networks. While these papers demonstrate that the accuracy of the NCC classifier improves across the layers, they do not explore the entire set of NC properties. While Tirer & Bruna (2022) explores a two-layer unconstrained features model, they only present experimental evidence for NC1 and NC2 in the layer prior to the last hidden layer. Their theoretical results show the emergence of NC in the classifier and the features at the penultimate layer in the case of ReLU non-linearities, but do not describe the emergence of NC in intermediate layers.

Deep networks as associative memories. Associative memories have been a popular topic in neural networks for over half a century, starting with the work of Kohonen (1989) who proposed a mathematical model for a non-hierarchical pattern storage system. This work inspired many subsequent studies, including the Self-Organizing Map algorithm by Kohonen (1989) and the Simple Recurrent Network by Anderson (1972). Hopfield (1982) later proposed the Hopfield network, a recurrent neural network that can store and recall multiple patterns. Kanerva (1992) proposed the sparse distributed memory, which uses high-dimensional binary vectors for efficient pattern storage. Associative memories have also been used in signal processing applications such as holography prior to their being studied as neural networks (Willshaw et al., 1969).

The notion of associative memories can be used to interpret and understand the layers of a deep neural network, and in some cases, describe the entire network. This approach, known as the dual form of neural networks (Irie et al., 2022; Aizerman et al., 1964), allows for interesting practical applications such as editing generative models (Bau et al., 2020) and classifier rules (Santurkar et al., 2021). Recent research (Dai et al., 2021; Geva et al., 2020; Meng et al., 2022) has also focused on exploring the connection between associative memories and transformer architectures.

3. Problem Setup

We consider the problem of training deep neural networks to solve multi-class classification problems between an input space $\mathcal{X} \subset \mathbb{R}^d$ and a label space $\mathcal{Y}_C$ with cardinality C. We use a one-hot encoding for the label space. The deep neural network classifiers $f_{\boldsymbol{W}} : \mathcal{X} \to \mathbb{R}^C$ that we study consist of compositions of parametric transformations and can be defined as:

$$f_{\boldsymbol{W}}(\boldsymbol{x}) = T_L \circ \ldots \circ T_1(\boldsymbol{x}),$$

where $T_l : \mathbb{R}^{p_l} \to \mathbb{R}^{p_{l+1}}$ is a parametric transformation with parameters $\boldsymbol{W}_\ell$. For instance, T_ℓ could be a fully-connected layer with a nonlinearity, $T_\ell(\boldsymbol{z}) = \sigma(\boldsymbol{W}_\ell \boldsymbol{z})$, or a residual block $T_\ell(\boldsymbol{z}) = \sigma(\boldsymbol{z} + \boldsymbol{W}_\ell^2 \sigma(\boldsymbol{W}_\ell^1 \boldsymbol{z}))$ or a convolutional layer. Here, $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-linear function that is applied coordinate-wise, such as the ReLU activation function $\sigma(x) = \max(0, x)$. We use $\boldsymbol{W} = \{\boldsymbol{W}_L, \boldsymbol{W}_{L-1}, \ldots, \boldsymbol{W}_1\}$ to denote the parameters of each one of the layers. In this paper, we will be interested in the characteristics of the *features* computed by the deep network at each layer. We define features at layer ℓ for the input $\boldsymbol{x}_{i,c}$ as $\boldsymbol{h}^\ell(\boldsymbol{x}_{i,c}) = T_\ell \circ \ldots \circ T_1(\boldsymbol{x}_{i,c})$.

In this setting, we aim to learn the classifier from a balanced training dataset $S := \{(\boldsymbol{x}_{i,c}, y_{i,c})\}_{i=1,c=1}^{N,C}$ of CN samples consisting of N independent and identically distributed (i.i.d.) samples drawn from each of the C classes. To train the classifier, we typically minimize the regularized empirical loss function

$$L_S^\lambda(f_{\boldsymbol{W}}) := \frac{1}{CN} \sum_{c=1}^{C} \sum_{i=1}^{N} \mathcal{L}(f_{\boldsymbol{W}}(\boldsymbol{x}_{i,c}), y_{i,c}) + \lambda \mathcal{R}(\boldsymbol{W})$$

where $\mathcal{L} : \mathbb{R}^C \times \mathcal{Y}_C \to [0, \infty)$ is a non-negative loss function (e.g., squared error or cross-entropy losses) and regularizer $\mathcal{R}(\boldsymbol{W})$ (such as L_2 regularization) controls the complexity of the function $f_{\boldsymbol{W}}$ and typically improves generalization.

4. Intermediate Neural Collapse

In a recent paper, Papyan et al. (2020) described four properties of the terminal phase of training (TPT) in deep networks using the cross-entropy loss function. TPT starts at the point where the training error becomes zero and continues until training is stopped. During TPT, the training error remains effectively zero while the training loss continues to decrease. Direct empirical measurements expose an inductive bias they call Neural Collapse (NC), involving four interconnected properties. In this paper, we extend the characterization of Neural Collapse (NC) by examining its presence in intermediate layers, in addition to its previously studied presence at the last layer features and weights.

Before mathematically describing the conditions of Intermediate Neural Collapse, we first define the following first and second-order statistics of features in deep networks. The mean class features and the global mean features for layer ℓ are computed as follows:

$$\mu_c^\ell := \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{h}_{i,c}^\ell \qquad \mu_G^\ell := \frac{1}{C} \sum_{c=1}^{C} \mu_c^\ell$$

The within-class, between-class, and total covariance matrices for layer ℓ are computed as:

$$\Sigma_W^\ell = \frac{1}{NC}\sum_{c=1}^{C}\sum_{i=1}^{N}(\boldsymbol{h}_{i,c}^\ell - \mu_c^\ell)(\boldsymbol{h}_{i,c}^\ell - \mu_c^\ell)^\top$$

$$\Sigma_B^\ell = \frac{1}{C}\sum_{c=1}^{C}(\mu_c^\ell - \mu_G^\ell)(\mu_c^\ell - \mu_G^\ell)^\top$$

$$\Sigma_T^\ell = \frac{1}{NC}\sum_{c=1}^{C}\sum_{i=1}^{N}(\boldsymbol{h}_{i,c}^\ell - \mu_G^\ell)(\boldsymbol{h}_{i,c}^\ell - \mu_G^\ell)^\top$$

We note that the total covariance can be decomposed into the within and between class covariances $\Sigma_T^\ell = \Sigma_W^\ell + \Sigma_B^\ell$. We now characterize Intermediate Neural Collapse through the following conditions:

(NC1) Feature variability suppression. Most of the total covariance of the features in a layer is contained in the between-class covariance. We compare the normalized within-class variance $\operatorname{Tr}(\Sigma_W^\ell)/\operatorname{Tr}(\Sigma_T^\ell)$ and the normalized between-class variance $\operatorname{Tr}(\Sigma_B^\ell)/\operatorname{Tr}(\Sigma_T^\ell)$. An intermediate layer shows feature variability suppression if the normalized within-class variance is smaller than a threshold, $\operatorname{Tr}(\Sigma_W^\ell)/\operatorname{Tr}(\Sigma_T^\ell) < \epsilon$. From our experiments, we observe that $\epsilon \approx 0.2$ is a reasonable choice. Since $\Sigma_W^\ell + \Sigma_B^\ell = \Sigma_T^\ell$, this means that most of the variability in the features comes from the distance between the between-class covariance and the within-class variability is suppressed. This is a weaker definition than the original definition of NC1 in the last layer, which claims that NC1 is achieved when $\operatorname{Tr}(\Sigma_W^L \cdot (\Sigma_B^L)^\dagger) \to 0$.

(NC2) Simplex ETF structure. The class means at layer ℓ show a simplex ETF structure if the following two conditions are satisfied: 1) $\left|\|\mu_c^\ell - \mu_G^\ell\|_2 - \|\mu_{c'}^\ell - \mu_G^\ell\|_2\right| \to 0$, or the centered class means of the layer features become equinorm; and 2) if we define $\tilde{\mu}_c^\ell = \frac{\mu_c^\ell - \mu_G^\ell}{\|\mu_c^\ell - \mu_G^\ell\|_2}$, then we have $\langle \tilde{\mu}_c^\ell, \tilde{\mu}_{c'}^\ell\rangle = -\frac{1}{C-1}$ for $c \neq c'$, or the centered class means are also equiangular. This condition is the same as the original simplex ETF definition for the last layer class means.

(NC3) Alignment between features and weights. Let us consider the matrix of centered class means at layer ℓ given by $\boldsymbol{M}_\ell = [\mu_c^\ell - \mu_G^\ell]_{c=1}^C \in \mathbb{R}^{p_\ell \times C}$ and its alignment with $\boldsymbol{W}_\ell \in \mathbb{R}^{p_{\ell+1}\times p_\ell}$. At the last layer, these matrices have the same dimension and hence we say the last layer features and classifier are aligned when $\left\|\frac{\boldsymbol{W}_L^\top}{\|\boldsymbol{W}_L\|_F} - \frac{\boldsymbol{M}_L}{\|\boldsymbol{M}_L\|_F}\right\| \to 0$, since each row of the weight matrix corresponds to the relevant class mean column in $\boldsymbol{M}_\ell$.

At intermediate layers we find the Principal Angles Between Subspaces (PABS) (Jordan; Björck & Golub, 1973) $\theta_1, \ldots, \theta_C$ between the range space of $\boldsymbol{M}_\ell$ and the top C rank input space of $\boldsymbol{W}_\ell$. An intermediate layer shows feature-weight alignment if $\frac{1}{C}\sum_{k=1}^{C}\cos(\theta_k) \to 1$, and the top C singular values of $\boldsymbol{W}_\ell$ are equal to each other. At the last layer, the alignment and distance-based definitions of NC3 are equivalent.

(NC4) Behavioral equivalence to nearest center classification. For a given layer, NC4 is satisfied if the decision of the deep classifier and that of the nearest-class-center (NCC) decision rule using the features at layer ℓ converge to each other: $\arg\max_c \langle \boldsymbol{W}_L^c, \boldsymbol{h}^L(\boldsymbol{x})\rangle \to \arg\min_c \|\boldsymbol{h}^\ell(\boldsymbol{x}) - \mu_c^\ell\|_2$.

In the next section we will see that for deep networks which show NC in the last layer, there exists a hidden layer in the network beyond which all subsequent layers show the above four conditions of intermediate NC.

5. Results

In this section, we will present and analyze the results of our experiments that demonstrate the existence of intermediate Neural Collapse (NC). The experimental details can be found in Appendix A. For the results, we used four datasets - MNIST, FashionMNIST, CIFAR10, and SVHN - and three architectures - Multilayer Perceptrons (MLPs), Convolutional Neural Networks, and Residual Networks.

5.1. Intermediate Neural Collapse

We present a list of figures that support our claim that intermediate Neural Collapse (NC) occurs in deep networks. These figures demonstrate results from the MNIST and CIFAR10 datasets on three different networks. Results from additional datasets can be found in appendix C. Each figure is divided into two rows, with the top row showing results from the MNIST dataset and the bottom row showing results from the CIFAR10 dataset. The two figures in each column display results from the same type of network. A vertical green line is used to indicate the layer at which intermediate collapse begins in all figures.

In Fig. 2, we investigate the suppression of feature variability through the layers of the network. In the top half of each subfigure, we plot the within-class covariance $\operatorname{Tr}(\Sigma_W^\ell)$ (dotted), between-class covariance $\operatorname{Tr}(\Sigma_B^\ell)$ (dashed), and total covariance $\operatorname{Tr}(\Sigma_T^\ell)$ (solid). In the bottom half, we plot the normalized within-class covariance $\operatorname{Tr}(\Sigma_W^\ell)/\operatorname{Tr}(\Sigma_T^\ell)$ (dotted) and normalized between-class covariance $\operatorname{Tr}(\Sigma_B^\ell)/\operatorname{Tr}(\Sigma_T^\ell)$. From the normalized plots, we can observe that at a certain layer in the deep classifier, the between-class covariance becomes much more significant than the within-class covariance. In all subsequent layers, the within-class covariance remains a small fraction of the total covariance. These layers can be referred to as the "collapsed" layers. In Fig. 5, we see that the accuracy of the nearest class center classifier (NCC) matches the accuracy of the classifier in the collapsed layer.

In Fig. 3, we present results showing the convergence of class means to a simplex equiangular tight frame (ETF) in collapsed layers. Specifically, we plot the average value of $\cos(\angle(\mu_c^\ell - \mu_G^\ell, \mu_{c'}^\ell - \mu_G^\ell)) + \frac{1}{C-1}$, its normalized (by the mean) standard deviation, and the normalized (by the mean) standard deviation of $\|\mu_c^\ell - \mu_G^\ell\|_2$ in the top, middle, and bottom panels of each subfigure. We can observe that the class means approach a simplex ETF in the deepest layers, while in earlier collapsed layers, there may still be some variability, especially in the case of convolutional neural networks.

In Fig. 4, we investigate the alignment between features and weights across layers. We plot the average of the cosines of the principal angles between the subspaces spanned by the centered class means $\boldsymbol{M}_\ell$ and the input subspace of the weight matrix $\boldsymbol{W}_\ell$. We can observe that the alignment between class means and weight matrices is strongest in the collapsed layers, and that this alignment is much higher than at initialization, where the features and weights are essentially random. Moreover, in Fig. 6 we see that in the collapsed layers, the top C singular values of the weights are nearly equal. These two observations establish NC3. In the case of residual neural networks, it is interesting to note that the alignment is strongest at layers just before a residual connection, and that the features within a residual block are not as well aligned with their weights.

5.2. Stable Rank of intermediate features and weights

Having established the conditions of intermediate NC, we further investigate the structure of the weights and features that are learned in deep networks.

Low rank and near orthogonal weights. In the top row of Fig. 6, we present the singular value spectrum of the weight matrices/kernels through the layers. We observe that in the collapsed layers of MLPs and resnets, the top C singular values are significantly larger than the remaining singular values, indicating that the weights have a low-rank structure. Additionally, these top C singular values are highly concentrated, indicating that the weights are nearly orthogonal. This structure is less pronounced in the convnet, but we can still see a concentration of the top singular values. These observations align with the conclusions in (Papyan, 2020), which found that the feature class means at different layers are also near orthogonal.

Stable rank of intermediate features. In the bottom row of Fig. 6, we present the results of the stable rank analysis of the matrix of within-class features centered around their class means $\boldsymbol{H}_c^\ell = [\boldsymbol{h}_{i,c}^\ell - \mu_c^\ell]_{c=1}^C$. The stable rank, which is a lower bound of the actual rank and can be computed without storing the entire matrix, is defined as $\|\boldsymbol{H}_c^\ell\|_F^2 / \|\boldsymbol{H}_c^\ell\|_2^2$. We can see that the rank of the class features decreases in the collapsed layers, which is consistent with our observation that in those layers the within-class covariance becomes a smaller fraction of the total covariance. This is also expected with low-rank weight matrices in these layers, as we can see in the top row of Fig. 6. In the layers below the top layer, we can see that the rank of the features is very high. This suggests that the deep network first projects the samples into a high-dimensional space, where it is easier to find a classification boundary, and then extracts the most discriminative features to classify the samples. This "hunchback" structure in the dimensionality of the features was also observed in previous studies such as (Recanatesi et al., 2019; Ansuini et al., 2019), though both of these papers used a nonlinear measure of dimension to establish this observation.

5.3. Fixing all Collapsed Layers with Simplex ETFs

One implication of neural collapse is that the last layer of a deep network can be fixed to a simplex ETF, without negatively impacting performance (Zhu et al., 2021). In a similar fashion, we test whether one can fix all of the collapsed layers to be simplex ETFs and still maintain good performance. In this experiment, we train the bottom L layers and fix the rest of the $10 - L$ layers to be canonical simplex ETFs (Fig. 7). Specifically, the last layer is set to be a rank $C-1$ simplex ETF (for a C class problem), while the layers below are set to be rank $H - 1$ simplex ETFs (where H is the width of the network). Namely, the rank K canonical simplex ETF is $\sqrt{\frac{K}{K-1}}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)$. At the last layer we set $\boldsymbol{W}_L^\top = \sqrt{\frac{C}{C-1}}\boldsymbol{P}\left(\boldsymbol{I}_C - \frac{1}{C}\mathbf{1}_C\mathbf{1}_C^\top\right)$ where $\boldsymbol{P} \in \mathbb{R}^{d\times C}$ contains the first C columns of a $d \times d$ identity matrix, which lifts a $C\times C$ ETF to a $d\times K$ matrix (Zhu et al., 2021). In Fig. 7 we present the results of this experiment using MLPs on MNIST, FashionMNIST, and CIFAR10. We observe that replacing the collapsed layers (layers 7-10 or 8-10 in the case of CIFAR10) with fixed simplex ETFs does not negatively impact performance, but replacing non-collapsed layers (layers 2-6 or 2-7 in the case of CIFAR10) does. This observation suggests that the features learned in the bottom half of the network are most crucial.

5.4. Solutions without Intermediate Neural Collapse

In the process of hyperparameter search, there were solutions showing varying degrees of intermediate NC. Some examples that displayed NC in the final hidden layer, as described in (Papyan et al., 2020), did not display intermediate NC. We highlight this to encourage investigations into the mechanics that produce intermediate NC, as well as the effects of this intermediate collapse on performance. Examples of two convolutional networks trained of CIFAR10 and Fashion MNIST that do not demonstrate intermediate NC, but do show NC in the final hidden layer, are given in appendix B.

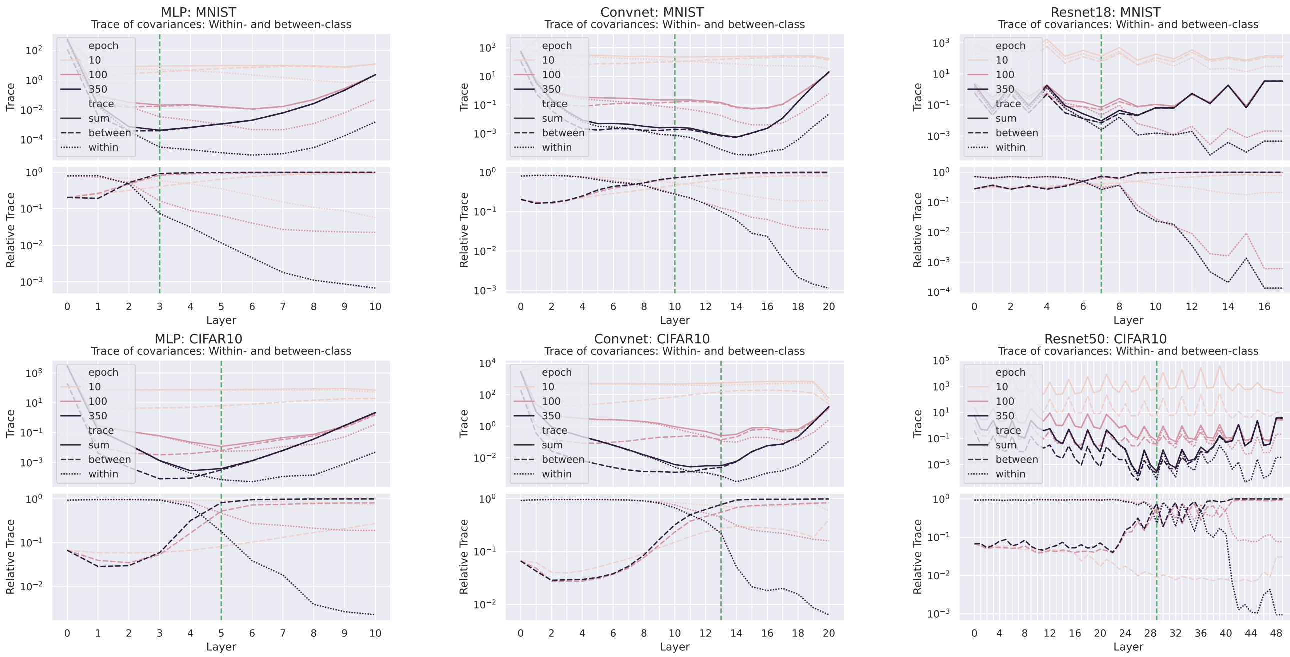

Figure 2. **(NC1) Feature variability suppression:** There is a layer in a deep classifier (vertical green line) where $\mathrm{Tr}(\Sigma_W^\ell)$ (dotted) contributes a significantly smaller fraction to $\mathrm{Tr}(\Sigma_T^\ell)$ (solid) than $\mathrm{Tr}(\Sigma_B^\ell)$ (dashed). In all subsequent layers this fraction generally remains below this threshold, showing feature variability suppression.

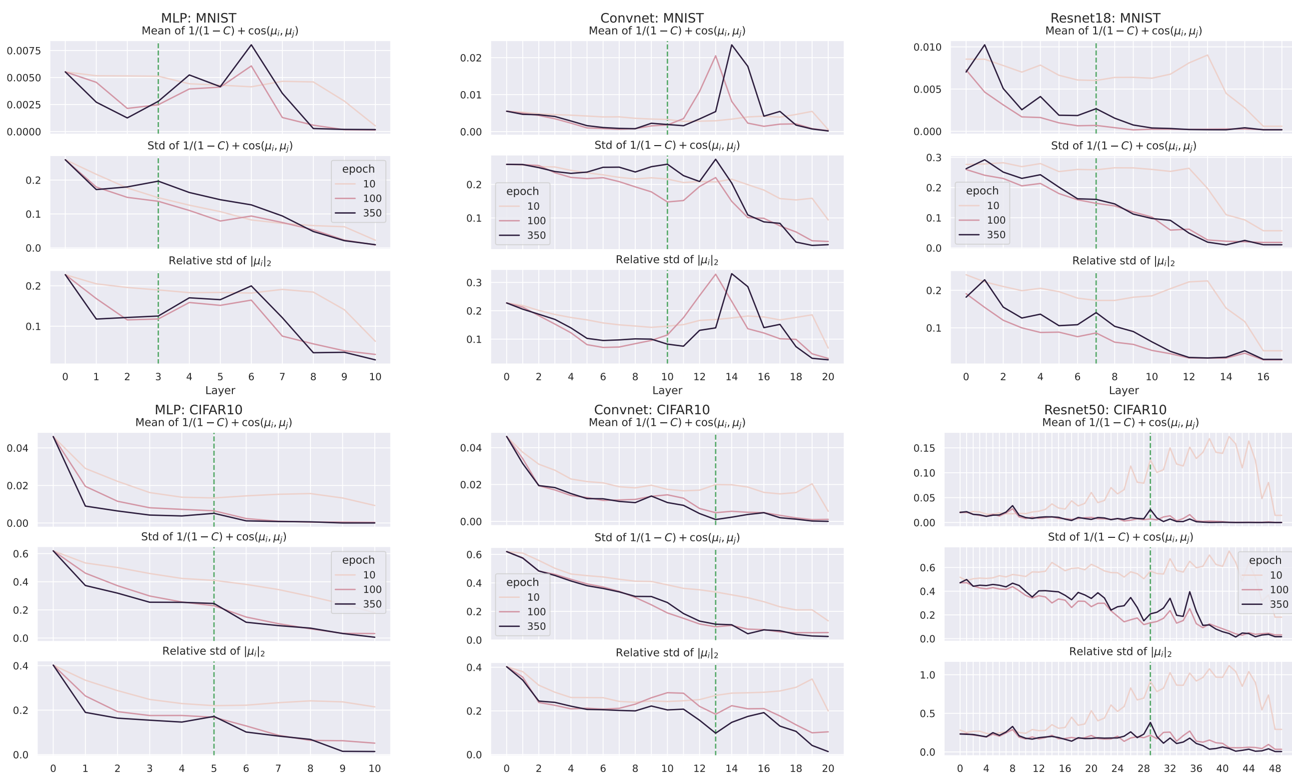

Figure 3. **(NC2) Convergence of class means to a Simplex ETF:** We see that as training progresses $\{\mu_c^\ell - \mu_G^\ell\}$ approach equinorm and maximal equiangularity in the collapsed layers, though this is most clearly achieved in the layers closest to the output.

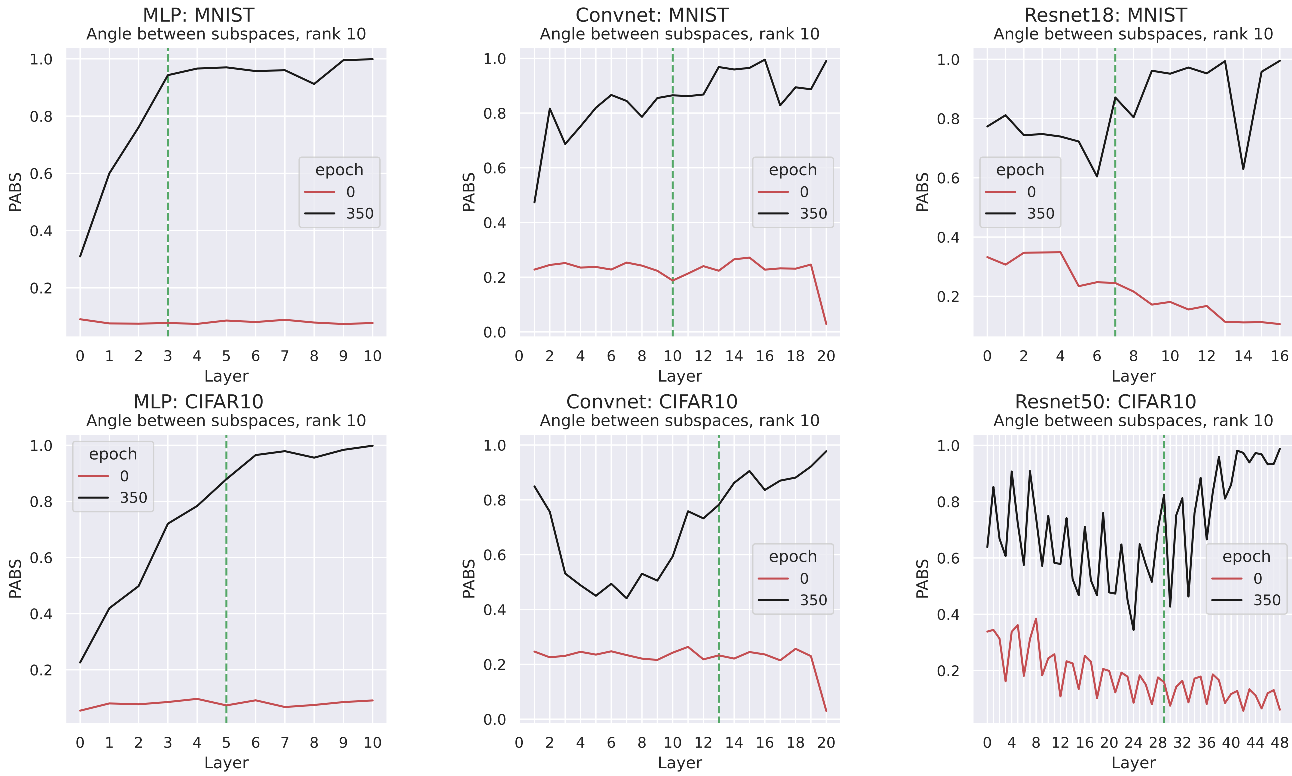

Figure 4. **(NC3) Feature-Weight Alignment:** In collapsed layers we see feature weight alignment measured as the average of the cosines of the principal angles between the subspaces. This is significantly above the alignment between random subspaces (at initialization).

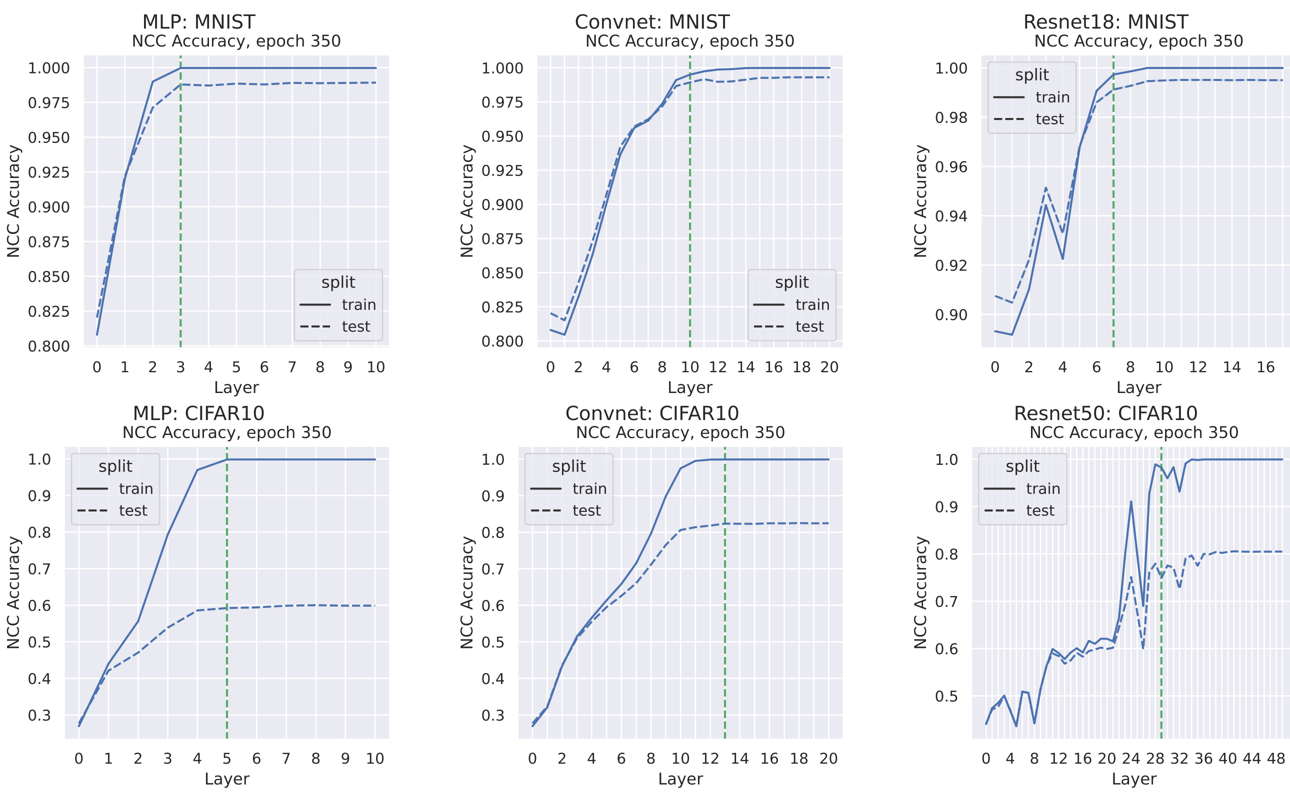

Figure 5. **(NC4) Equivalence to Nearest Class Center (NCC) classification** NCC classifier agrees with f_W in the collapsed layers.

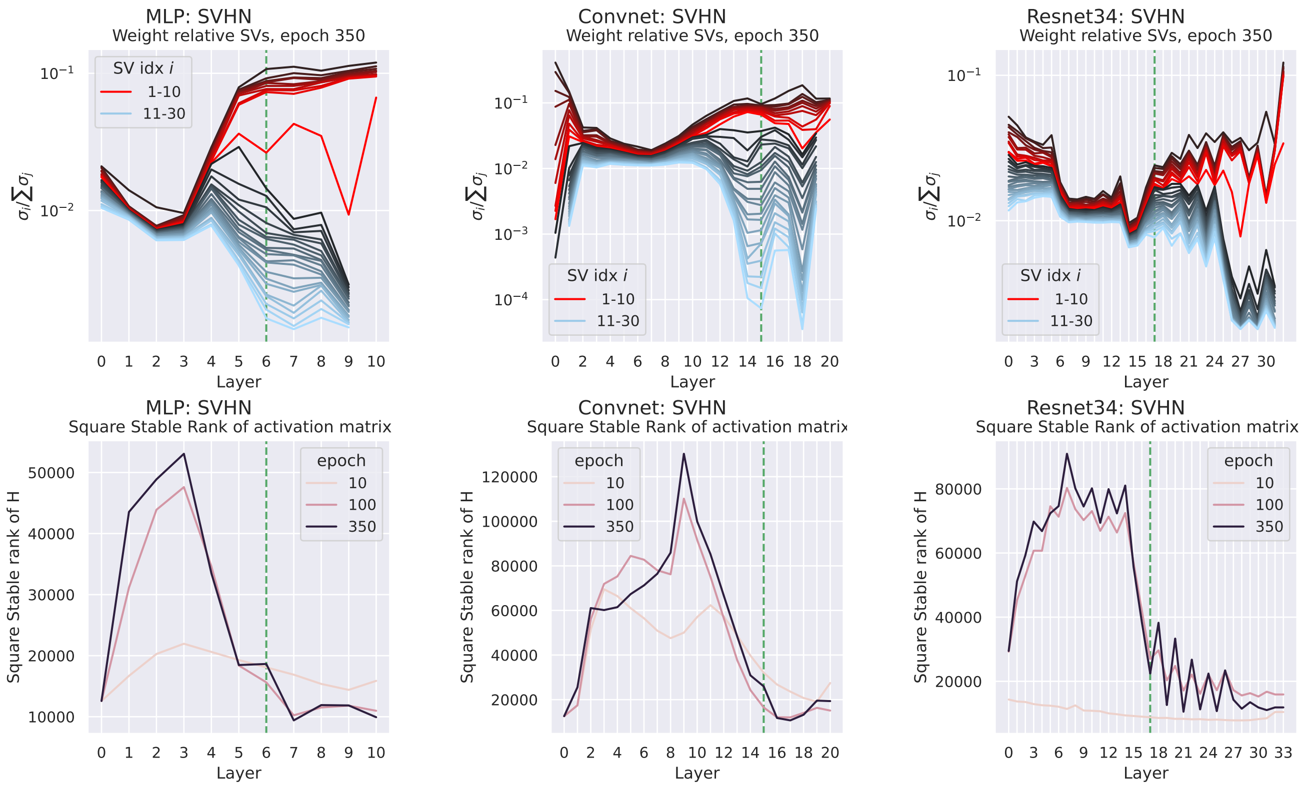

Figure 6. **Estimates of Ranks of Weights and (Within Class) Features** We see a low rank structure in the weights in collapsed layers, while previous layers are nearly full rank. This is also reflected in the stable rank of the features which first increases and then decreases in the collapsed layers.

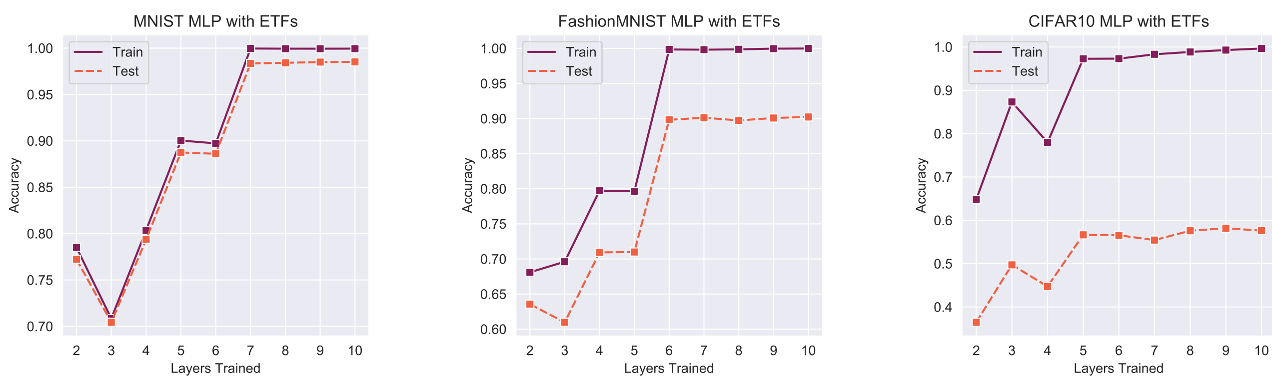

Figure 7. **Comparison of performance of fixing collapsed layers to be ETFs** We compare the accuracy achieved by fixing a certain number of layers to be simplex ETFs. We see that fixing uncollapsed layers (lower than layer 6 or 7 in this instance) to be ETFs results in lower accuracy, while fixing collapsed layers (7 or 8 and above) does not hurt accuracy.

6. Neural Collapse and Associative Memories

Associative memories (Kohonen, 1989; Anderson, 1972) are systems that store associations between stimuli and responses. If we have a number of stimuli response pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, an associative memory $\boldsymbol{A}$ when probed with a stimulus $\boldsymbol{x}_i$ will return the response $\boldsymbol{y}_i$. If the stimuli $\boldsymbol{x}_i$ are orthogonal, or near orthogonal, we can construct the associative memory from the data as $\boldsymbol{A} = \sum_{i=1}^{N} \boldsymbol{y}_i \boldsymbol{x}_i^\top$. We have the following relationship between Associative Memories and deep networks that exhibit NC:

Remark 6.1. An associative memory constructed from the last layer features and one-hot labels of a deep network at neural collapse (NC) is equivalent to the classifier weight matrix. To see this, let us collect the centered last layer features into a matrix $\boldsymbol{H}_L \in \mathbb{R}^{p \times NC}$ and the one-hot labels into $\boldsymbol{Y} \in \mathbb{R}^{C \times NC}$. If the condition of variability collapse is achieved, we have $\boldsymbol{H}_L = \boldsymbol{M}_L \boldsymbol{Y}$, where $\boldsymbol{M}_L \in \mathbb{R}^{p \times C}$ is the matrix of centered class mean features at the last layer. Then constructing an associative memory, we get $\hat{\boldsymbol{W}}_L = \boldsymbol{Y}\boldsymbol{H}_L^\top = \boldsymbol{Y}\boldsymbol{Y}^\top \boldsymbol{M}_L^\top = N \times \boldsymbol{I}_C \boldsymbol{M}_L^\top$. This is also the weight matrix predicted by NC3 at the last layer. Moreover, since the $\boldsymbol{M}_L$ is a simplex ETF the keys for the associative memory are nearly orthogonal, which is a desirable property for robust recall.

While this does not immediately translate to intermediate layers due to the non-linearities involved, intermediate NC suggests that intermediate layers in deep networks may also be viewed as associative memories. This interpretation has already led to interesting applications such as introducing novel concepts to a generative model (Bau et al., 2020), and editing the prediction rules of a classifier to include new concepts (Santurkar et al., 2021). A thorough understanding of NC could help make this connection more concrete.

7. Conclusion and Future Work

In this paper, we identified several extensions of neural collapse for intermediate layers and empirically investigated these conditions in various neural network architectures. We empirically showed that several properties appear in intermediate layers during training: 1) feature variability suppression, 2) emergence of a simplex ETF in the layer class means, 3) alignment between features and weights, and 4) nearest class center classification with layer features.

As future work, it would be interesting to develop a theoretical foundation for intermediate NC in deep networks. Specifically, it would be worthwhile to study the connections between low-rank and orthogonal weight matrices, simplex ETF features, and optimization algorithms like stochastic gradient descent. The inductive bias towards simplex ETFs may help accelerate optimization, and make systems more efficient. Understanding intermediate NC may also help us choose features that better transfer across tasks (Galanti et al., 2022c;b;d). Whether intermediate NC is desirable for better generalization is also an important question for future work.

Acknowledgements

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. This work was also supported by an NSF grant (DMS-2134108). We would like to thank Vardan Papyan for his valuable comments and feedback.

References

Aizerman, M. A., Braverman, E. M., and Rozonoer, L. I. Theoretical foundation of potential functions method in pattern recognition. *Avtomatika i Telemekhanika*, 25(6): 917–936, 1964.

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644, 2017.

Allen-Zhu, Z. and Li, Y. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

Allen-Zhu, Z. and Li, Y. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

Anderson, J. A. A simple neural network generating an interactive memory. *Mathematical biosciences*, 14(3-4): 197–220, 1972.

Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dbc4d84bfcfe2284ba11beffb853a8c4-Paper.pdf.

Bau, D., Liu, S., Wang, T., Zhu, J.-Y., and Torralba, A. Rewriting a deep generative model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Ben-Shaul, I. and Dekel, S. Nearest class-center simplification through intermediate layers. *arXiv preprint arXiv:2201.08924*, 2022.

Björck, A. and Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Chen, M., Bai, Y., Lee, J. D., Zhao, T., Wang, H., Xiong, C., and Socher, R. Towards understanding hierarchical learning: Benefits of neural representations. *Advances in Neural Information Processing Systems*, 33:22134–22145, 2020.

Cohen, G., Sapiro, G., and Giryes, R. Dnn or k-nn: That is the generalize vs. memorize question. *arXiv preprint arXiv:1805.06822*, 2018.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/du19c.html.

Du, S. S., Zhai, X., Póczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *ArXiv*, abs/1810.02054, 2018.

Ergen, T. and Pilanci, M. Revealing the structure of deep neural networks via convex duality. *arXiv preprint arXiv:2002.09773*, 2020.

Fang, C., He, H., Long, Q., and Su, W. J. Layer-peeled model: Toward understanding well-trained deep neural networks. *CoRR*, abs/2101.12699, 2021. URL https://arxiv.org/abs/2101.12699.

Galanti, T., Galanti, L., and Ben-Shaul, I. On the implicit bias towards minimal depth of deep neural networks. *arXiv preprint arXiv:2202.09028*, 2022a.

Galanti, T., György, A., and Hutter, M. Improved generalization bounds for transfer learning via neural collapse. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*, 2022b. URL https://openreview.net/forum?id=VrK7pKwOhT_.

Galanti, T., György, A., and Hutter, M. On the role of neural collapse in transfer learning. In *International Conference on Learning Representations*, 2022c. URL https://openreview.net/forum?id=SwIp410B6aQ.

Galanti, T., György, A., and Hutter, M. Generalization bounds for transfer learning with pretrained classifiers, 2022d. URL https://arxiv.org/abs/2212.12532.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

Goldfeld, Z., Van Den Berg, E., Greenewald, K., Melnyk, I., Nguyen, N., Kingsbury, B., and Polyanskiy, Y. Estimating information flow in deep neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2299–2308. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/goldfeld19a.html.

Han, X., Papyan, V., and Donoho, D. L. Neural collapse under MSE loss: Proximity to and dynamics on the central path. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=w1UbdvWH_R3.

He, H. and Su, W. J. A law of data separation in deep learning. *arXiv preprint arXiv:2210.17020*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Irie, K., Csordás, R., and Schmidhuber, J. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. *arXiv preprint arXiv:2202.05798*, 2022.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS, pp. 8580–8589, Red Hook, NY, USA, 2018. Curran Associates Inc.

Ji, W., Lu, Y., Zhang, Y., Deng, Z., and Su, W. J. An unconstrained layer-peeled perspective on neural collapse. *arXiv preprint arXiv:2110.02796*, 2021.

Jordan, C. Essai sur la géométrie à n dimensions. *Bulletin de la Société Mathématique de France*, 3:103–174.

Kanerva, P. Sparse distributed memory and related models. Technical report, 1992.

Kohonen, T. Self-organization and associative memory, 1989.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Littwin, E., Galanti, T., Wolf, L., and Yang, G. On infinite-width hypernetworks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13226–13237. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/999df4ce78b966de17aee1dc87111044-Paper.pdf.

Lu, J. and Steinerberger, S. Neural collapse with cross-entropy loss. *CoRR*, abs/2012.08465, 2020. URL https://arxiv.org/abs/2012.08465.

Malach, E., Kamath, P., Abbe, E., and Srebro, N. Quantifying the benefit of using differentiable learning over tangent kernels. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7379–7389. PMLR, 18–24 Jul 2021.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual knowledge in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

Mixon, D. G., Parshall, H., and Pi, J. Neural collapse with unconstrained features. *CoRR*, abs/2011.11619, 2020. URL https://arxiv.org/abs/2011.11619.

Netzer, Y., Wang, T., Coates, A., Bissacco, B., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *Advances in neural information processing systems*, pp. 557–565, 2011.

Papyan, V. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020. URL http://jmlr.org/papers/v21/20-933.html.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Rangamani, A. and Banburski-Fahey, A. Neural collapse in deep homogeneous classifiers and the role of weight decay. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4243–4247. IEEE, 2022.

Recanatesi, S., Farrell, M., Advani, M., Moore, T., Lajoie, G., and Shea-Brown, E. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

Santurkar, S., Tsipras, D., Elango, M., Bau, D., Torralba, A., and Madry, A. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Shwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. ISSN 0028-0836. doi: 10.1038/nature16961.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Tirer, T. and Bruna, J. Extended unconstrained features model for exploring deep neural collapse. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21478–21505. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/tirer22a.html.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C. Non-holographic associative memory. *Nature*, 222(5197): 960–962, 1969.

Wojtowytsch, S. et al. On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers. *arXiv preprint arXiv:2012.05420*, 2020.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xu, M., Rangamani, A., Liao, Q., Galanti, T., and Poggio, T. Dynamics in deep classifiers trained with the square loss: normalization, low rank, neural collapse and generalization bounds. *Research*, 2023.

Yang, G. Tensor programs ii: Neural tangent kernel for any architecture, 2020. URL https://arxiv.org/abs/2006.14548.

Yang, G. and Littwin, E. Tensor programs iib: Architectural universality of neural tangent kernel training dynamics. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11762–11772. PMLR, 18–24 Jul 2021.

Zhou, J., Li, X., Ding, T., You, C., Qu, Q., and Zhu, Z. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. *arXiv preprint arXiv:2203.01238*, 2022.

Zhu, Z., Ding, T., Zhou, J., Li, X., You, C., Sulam, J., and Qu, Q. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

A. Experimental details

In this section, we describe the details of the experiments in the main text.

Datasets. We consider the MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), CIFAR10 (Krizhevsky & Hinton, 2009), and SVHN (Netzer et al., 2011) datasets. The images were preprocessed by centering and normalization using the pixel-wise mean and standard deviation as per the method described in (Han et al., 2022). No data-augmentation techniques were applied during training.

Network architectures. We conduct experiments with three deep network architectures. The first architecture is a multilayer perceptron (MLP) consisting of $L = 10$ hidden layers, where each layer contains a linear layer of width $H = 1024$, followed by batch normalization and ReLU. The last layer is linear. The second architecture is a deep convolutional network. This network starts with a stack of a 2×2 convolutional layer with stride 2, batch normalization, a convolution of the same structure, batch normalization, and ReLU. Following that we have a set of $L = 20$ stacks of 3×3 convolutional layers with $H = 128$ channels, stride 1 and padding 1, batch normalization, and ReLU. The last layer is linear. The third architecture type is a Resnet (He et al., 2016). We use Resnet architectures of different sizes for different datasets. We stick to the prescriptions of (Han et al., 2022) and use Resnet-18 for MNIST and FashionMNIST, Resnet-34 for SVHN, and Resnet-50 for CIFAR10.

Training details. For each combination of network and dataset, we trained the network to minimize mean squared error (MSE) loss using an SGD optimizer with momentum and weight decay. The hyperparameter settings were based on those used by Han et al. (2022). This includes a logarithmic learning rate sweep between 0.0001 and 0.25 decayed twice by a factor of 0.1,a momentum of 0.9, a weight decay of 5e-4, for 350 training epochs, and batch-size of 128. In the first epoch we do "warmup": Linearly increasing the learning rate from 0 to the specified learning rate through the batches. For our new models, the MLP and ConvNet, we used a learning rate decay of 0.2. We run our measurements on the model with the best test accuracy in the final epoch. The specific hyperparameters used can be found in the config files accompanying our code. All experiments were conducted on a cluster with NVIDIA Tesla V100, GeForce GTX 1080 TI, and A100 GPUs.

Intermediate neural collapse measurements. To make measurements, we compute the class means of the layer features over the entire training dataset. To measure NC1 (Fig. 2), we only need the trace of the within and between class covariances and do not need to store these matrices. For NC2 (Fig. 3), we measure the relative standard deviation of the norms of the class feature means, and the mean and the relative standard deviation of the pairwise inner products. For NC4 we construct an NCC classifier using the features of each layer (Fig. 5). For NC3, we perform Singular Value Decompositions on $\boldsymbol{W}_\ell = \boldsymbol{U_W} \boldsymbol{S_W} \boldsymbol{V_W}^\top$ and $\boldsymbol{M}_\ell = \boldsymbol{U_M} \boldsymbol{S_M} \boldsymbol{V_M}^\top$. We then measure the PABS between $\boldsymbol{V_W}$ - the basis for the input subspace of $\boldsymbol{W}_\ell$ - and $\boldsymbol{U_M}$ - the basis for the range space of $\boldsymbol{M}_\ell$. The cosines of the PABS can be obtained by computing the singular values of $\boldsymbol{V_W}^\top \boldsymbol{U_M}$ and their average is our NC3 measure (Fig. 4). For convolutional layers that transform features $\boldsymbol{h}^\ell \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ through kernels that are of dimension $\boldsymbol{W}_\ell \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_H \times k_W}$, we compute the alignment between the features and kernel along the C_{in} dimension and reshape the tensors accordingly.

B. Solutions without intermediate Neural Collapse

As noted in 5.4, we came across solutions that did not show NC during hyperparameter searches. We present two convolutional networks in fig. 8 trained on CIFAR10 and Fashion MNIST that did not demonstrate NC. Further examination is required to identify the conditions under which NC is achieved and to compare the capabilities of networks that exhibit NC and those that do not. This data highlights that NC solutions are not the only outcome of deep network training.

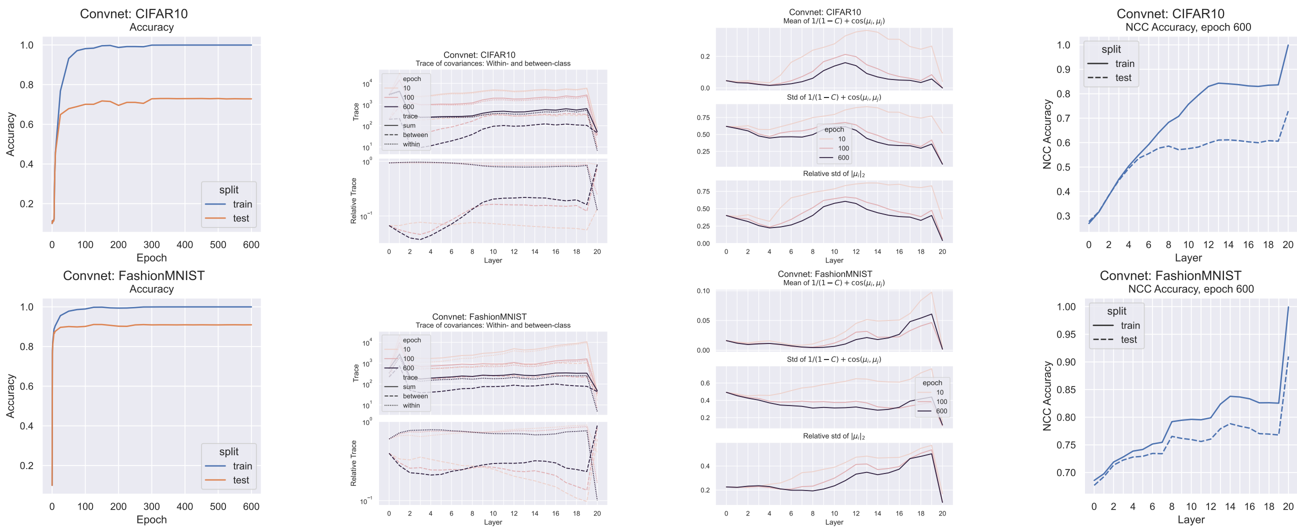

Figure 8. **Deep Networks without intermediate Neural Collapse** These convolutional networks trained on CIFAR10 (top row) and FashionMNIST (bottom row) do not show intermediate NC. The within class covariance stays high until the last hidden layer, the class means are not in a simplex ETF and the intermediate layers do not show NCC separability. This demonstrates that NC can happen in the final hidden layer, as described in (Papyan et al., 2020), while not appearing in the intermediate layers.

C. Additional Figures establishing Intermediate Neural Collapse

In this section we display Figures 9, 10, 11, 12 showing intermediate NC for all architectures on the FashionMNIST and SVHN datasets. The takeaways from these figures is largely the same as that from the figures for MNIST and CIFAR10 in the main text. We provide these figures here for completeness. We also display the rank estimates for weights and within class features across the layers of different deep networks for the remaining datasets. Figures 13, 14, 15 contain these plots.

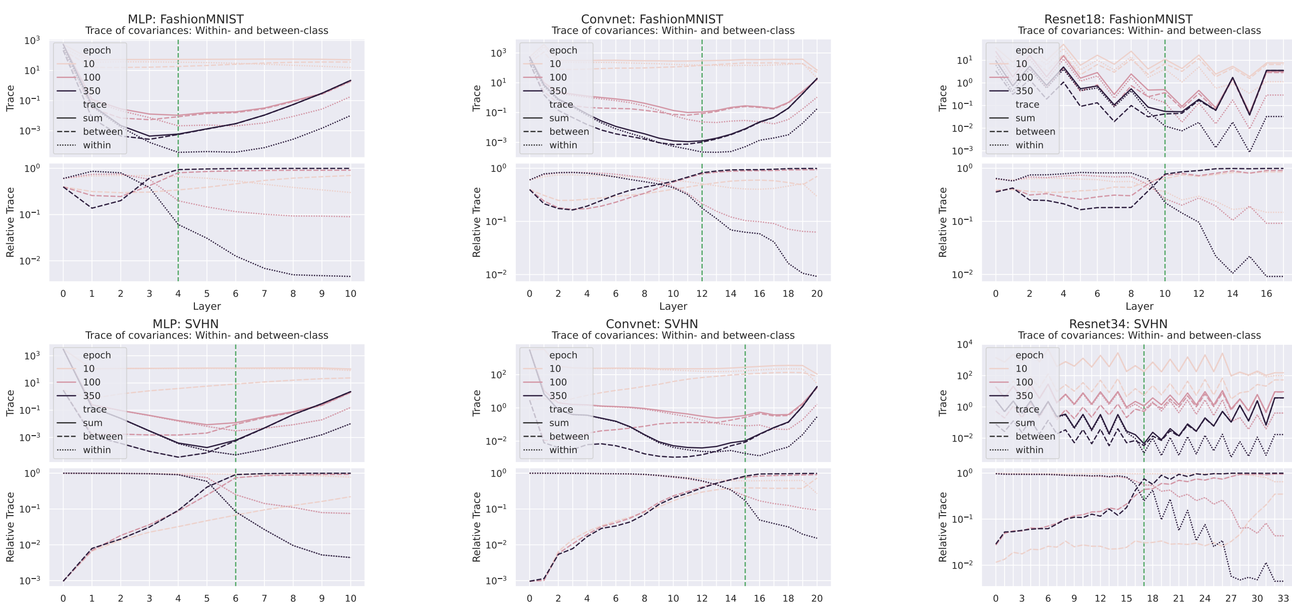

Figure 9. **(NC1) Feature variability suppression:** There is a layer in a deep classifier (vertical green line) where $\mathrm{Tr}(\Sigma_W^\ell)$ (dotted) contributes a smaller fraction to $\mathrm{Tr}(\Sigma_T^\ell)$ (solid) than $\mathrm{Tr}(\Sigma_B^\ell)$ (dashed). In all subsequent layers this fraction remains below this threshold, showing feature variability suppression

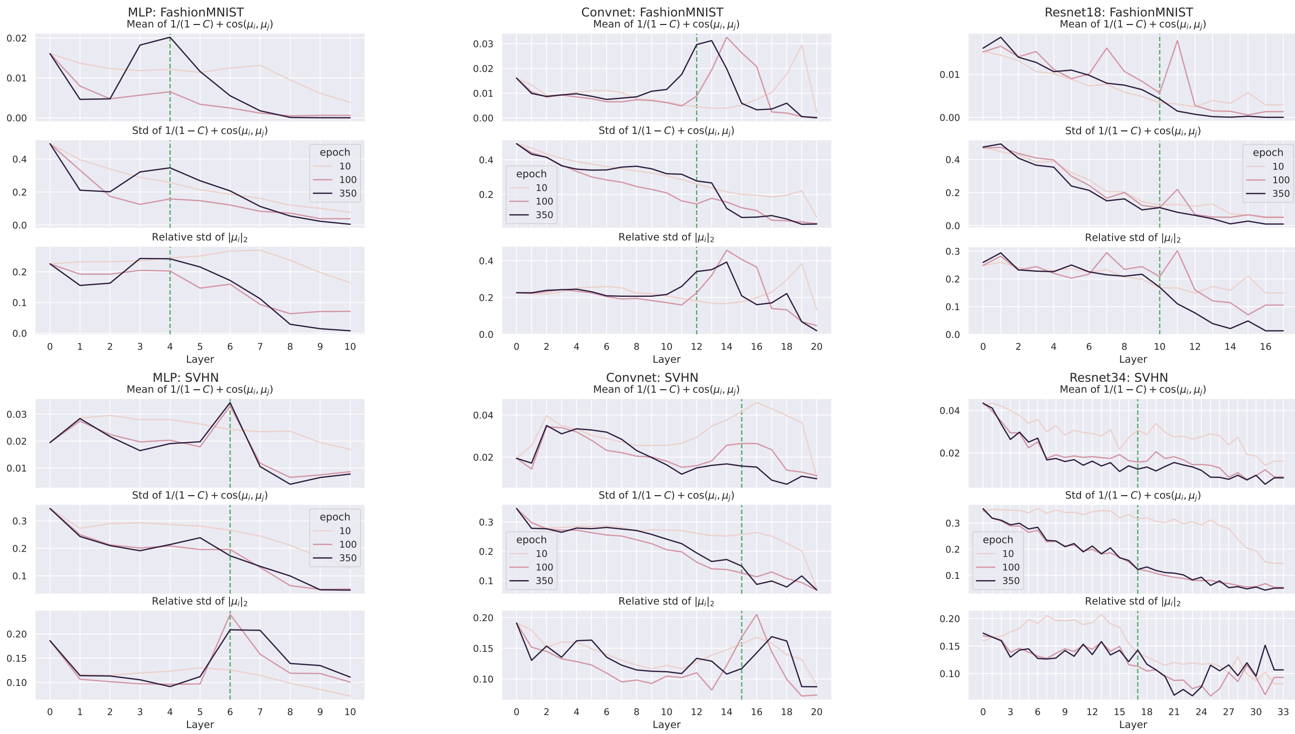

Figure 10. **(NC2) Convergence of class means to a Simplex ETF:** We see that as training progresses $\{\mu_c^\ell - \mu_G^\ell\}$ approach equinorm and maximal equiangularity in the collapsed layers, though this is most clearly achieved in the layers closest to the output.

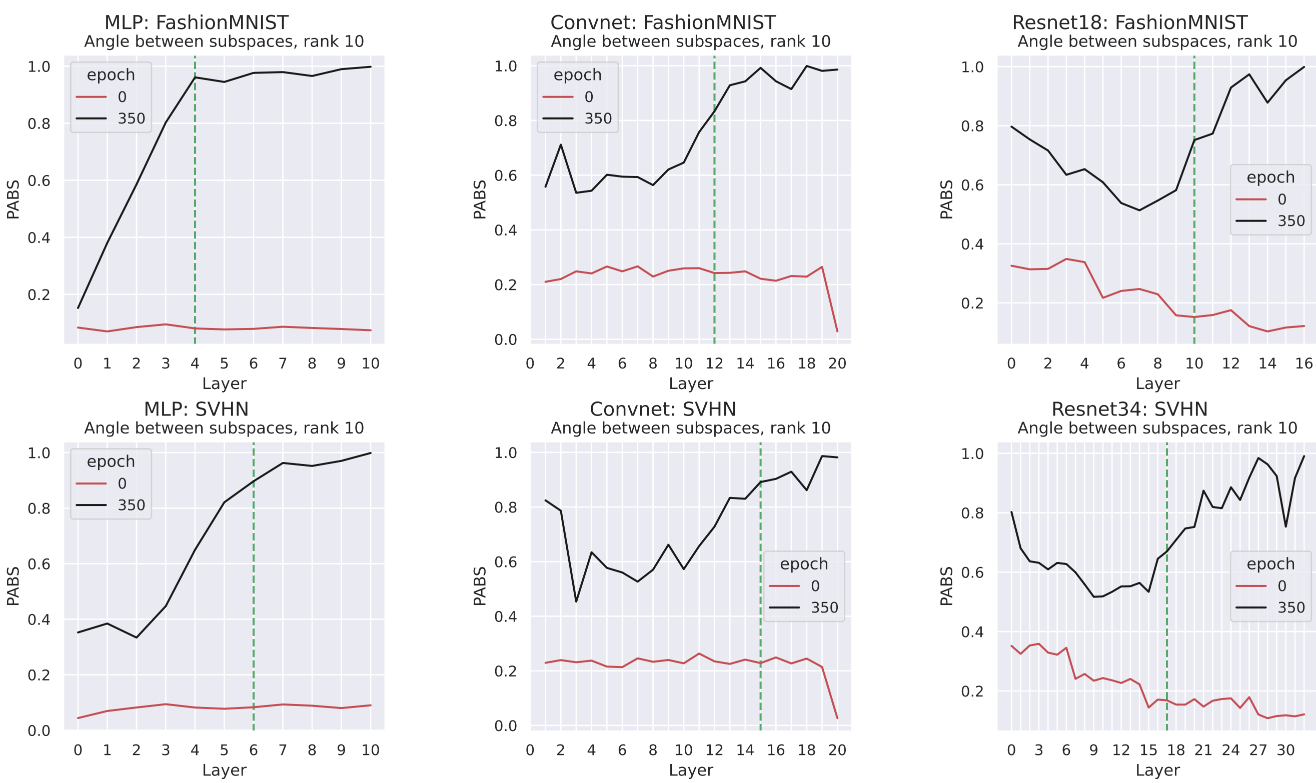

Figure 11. **(NC3) Feature-Weight Alignment:** In collapsed layers we see feature weight alignment measured as the average of the cosines of the principal angles between the subspaces. This is significantly above the alignment between random subspaces (at initialization).

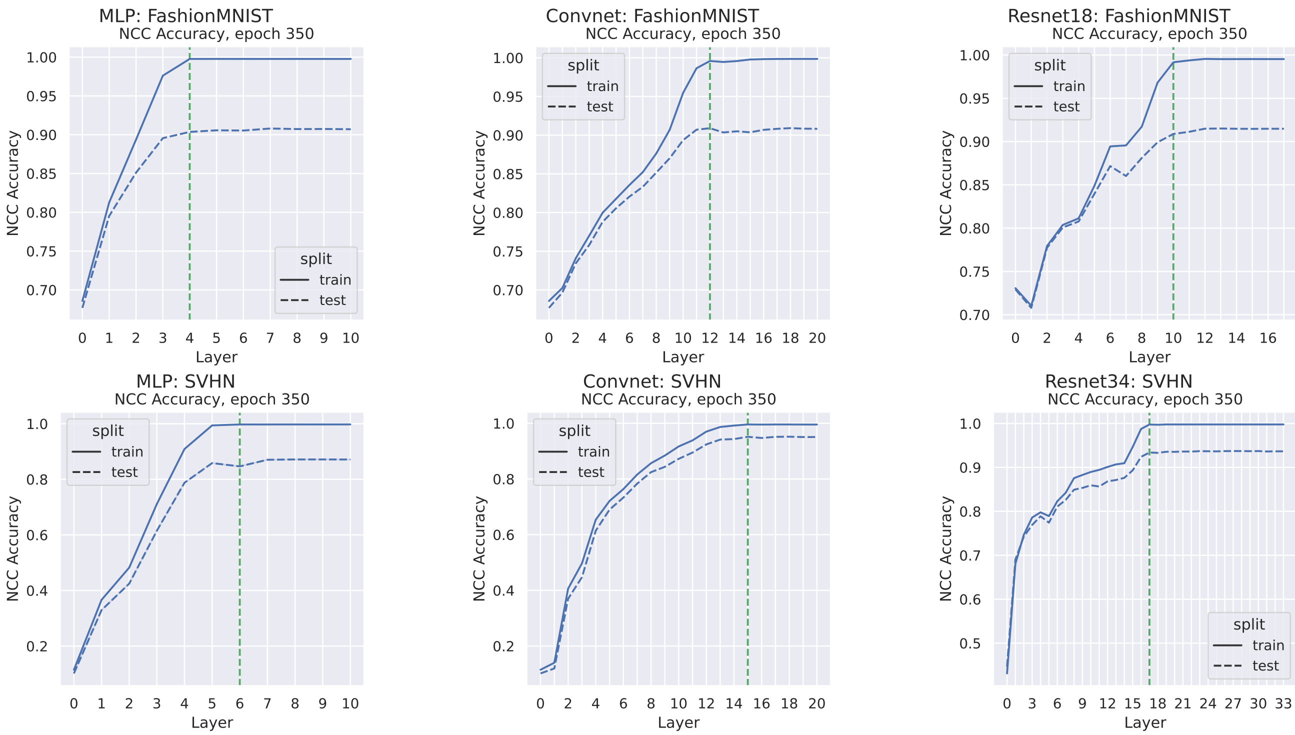

Figure 12. **(NC4) Equivalence to Nearest Class Center (NCC) classification** NCC classifier agrees with f_W in the collapsed layers.

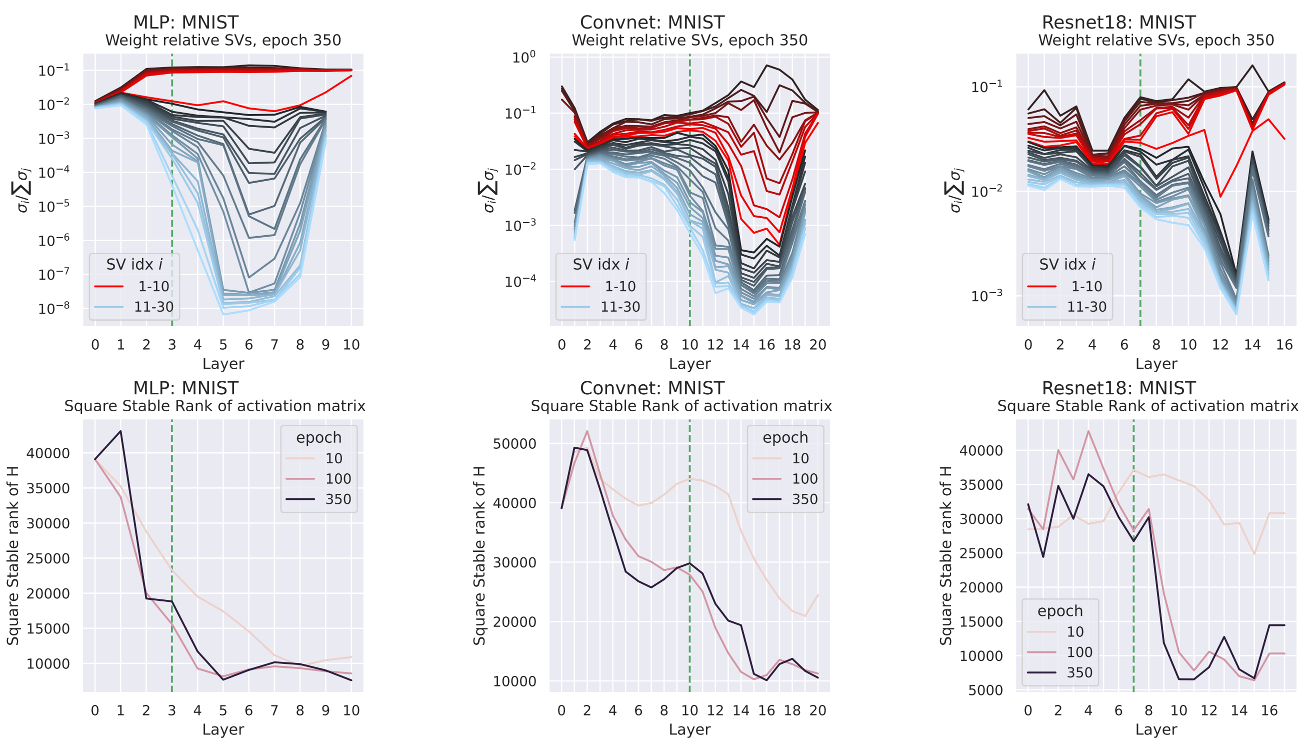

Figure 13. **Estimates of Ranks of Weights and (Within Class) Features** We see a low rank structure in the weights in collapsed layers, while previous layers are nearly full rank. This is also reflected in the stable rank of the features which first increases and then decreases in the collapsed layers.

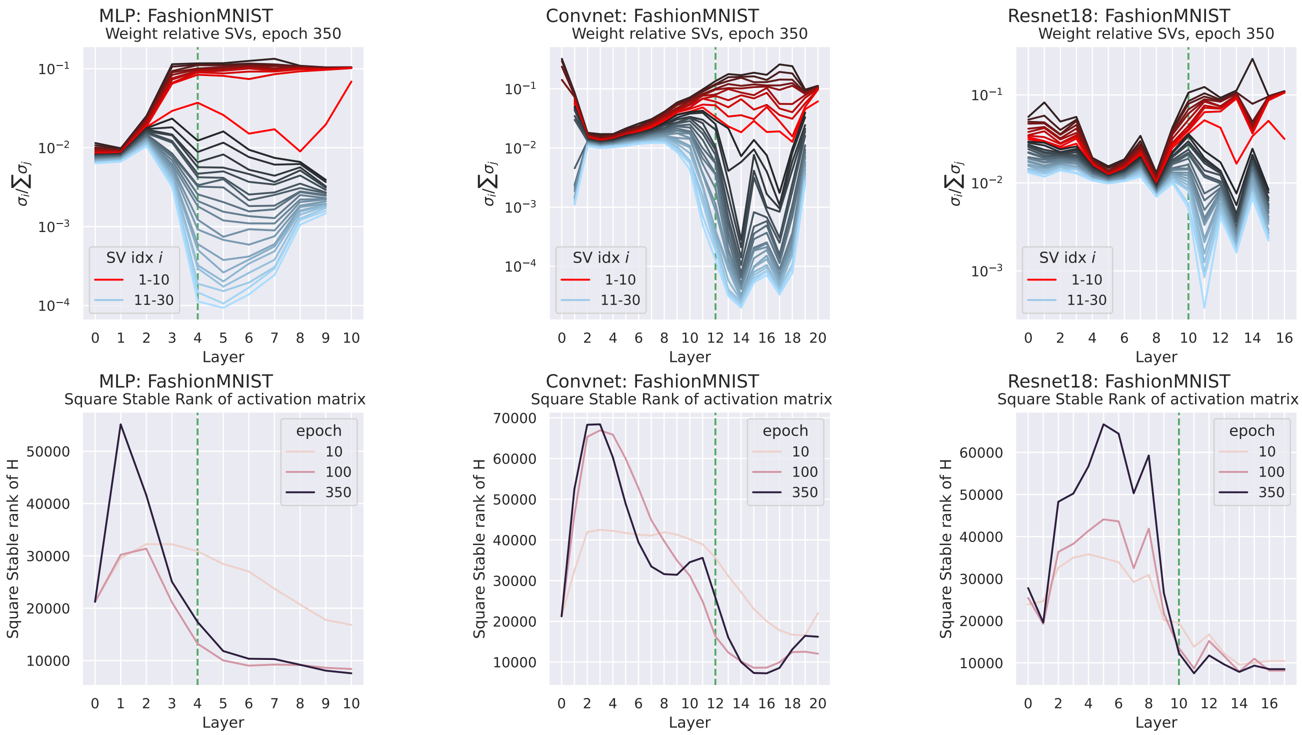

Figure 14. **Estimates of Ranks of Weights and (Within Class) Features** We see a low rank structure in the weights in collapsed layers, while previous layers are nearly full rank. This is also reflected in the feature stable rank of the features which first increases and then decreases in the collapsed layers.

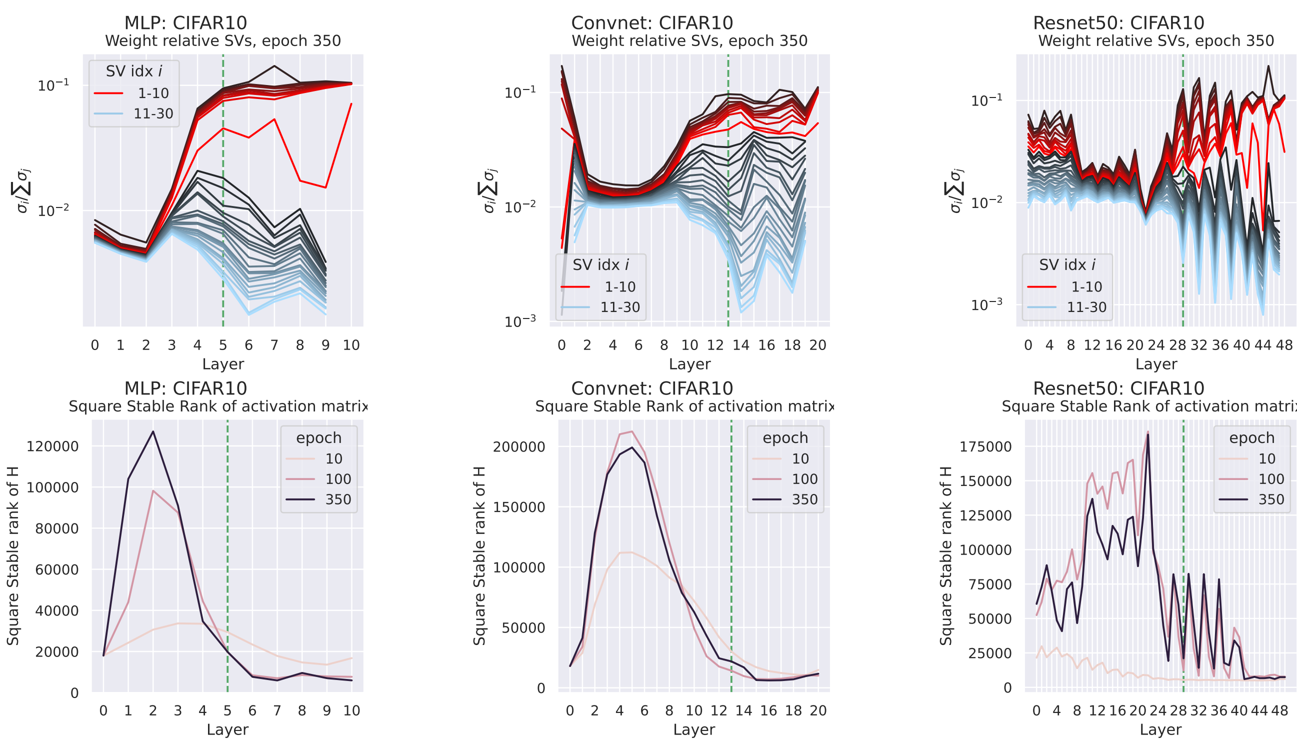

Figure 15. **Estimates of Ranks of Weights and (Within Class) Features** We see a low rank structure in the weights in collapsed layers, while previous layers are nearly full rank. This is also reflected in the stable rank of the features which first increases and then decreases in the collapsed layers.