Understanding How Nonlinear Networks Create Linearly Separable Features for Low-Dimensional Data

Alec S. Xu, Can Yaras, Peng Wang, Qing Qu

Department of Electrical Engineering & Computer Science, University of Michigan

January 4, 2025

Abstract

Deep neural networks have attained remarkable success across diverse classification tasks. Recent empirical studies have shown that deep networks learn features that are linearly separable across classes. However, these findings often lack rigorous justifications, even under relatively simple settings. In this work, we address this gap by examining the linear separation capabilities of shallow nonlinear networks. Specifically, inspired by the low intrinsic dimensionality of image data, we model inputs as a union of low-dimensional subspaces (UoS) and demonstrate that a single nonlinear layer can transform such data into linearly separable sets. Theoretically, we show that this transformation occurs with high probability when using random weights and quadratic activations. Notably, we prove this can be achieved when the network width scales polynomially with the intrinsic dimension of the data rather than the ambient dimension. Experimental results corroborate these theoretical findings and demonstrate that similar linear separation properties hold in practical scenarios beyond our analytical scope. This work bridges the gap between empirical observations and theoretical understanding of the separation capacity of nonlinear networks, offering deeper insights into model interpretability and generalization.

Contents

1 Introduction **2**
- 1.1 Our Contributions . . . 3
- 1.2 Notation and Paper Organization . . . 4

2 Preliminaries **5**
- 2.1 Assumptions on Input Data . . . 5
- 2.2 Linear Separability of UoS via Nonlinear Networks . . . 6

3 Theoretical Results **9**
- 3.1 Main Results . . . 9
  - 3.1.1 Extension to Multiple Subspaces . . . 11
- 3.2 Proof Sketches . . . 11

4 Experimental Results **12**
4.1 Phase Transition in Terms of Intrinsic Dimension . . . 12
4.2 Classification on Synthetic Data following a UoS Model . . . 13
4.2.1 Linear Separability of Features: Random vs. Trained Weights . . . 13
4.2.2 Effects of Nonlinear Activations . . . 13
4.3 CIFAR-10 Classification via MCR^2 Representations . . . 15

5 Discussion & Conclusion **16**
5.1 Comparison with Existing Literature . . . 16
5.2 Limitations & Future Directions . . . 17

A Supporting Results **23**
A.1 Expectation of Order Statistics: χ^2_m Random Variables . . . 23
A.2 Eigenvalues of Difference between Projection Matrices . . . 25
A.3 Expectation of Random Symmetric Rank-1 Matrices . . . 26
A.4 Matrix Bernstein's Inequality . . . 28

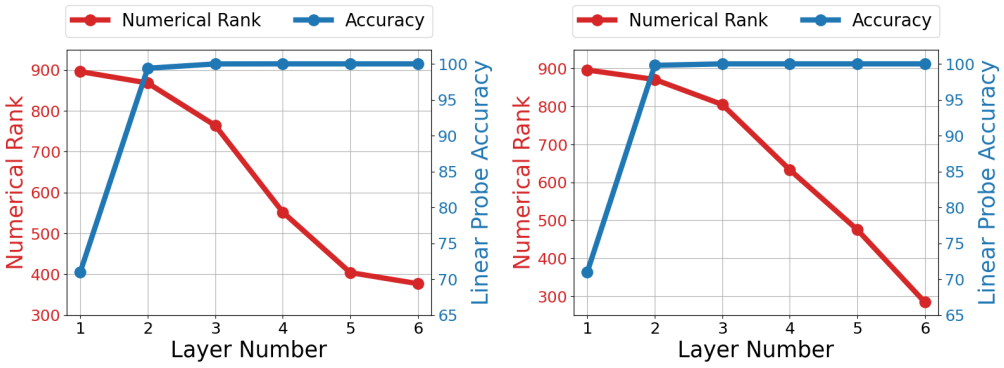
B Proof of Theorem 1 **29**
B.1 Conditions for Linear Separability . . . 29
B.2 Bounding the Failure Probability . . . 31
B.3 Final Result . . . 32

1 Introduction

Over the past decade, deep neural networks (DNNs) have achieved state-of-the-art performance in a wide range of applications, including computer vision [49, 21] and natural language processing [52, 54]. However, despite recent advances [23, 35, 24, 5, 27, 37, 64, 58, 63], the theoretical understanding of their empirical success is still primitive, even for relatively basic tasks. For example, in classification problems, the success of deep learning is often attributed to its ability to learn discriminative features that exhibit strong inter-class separation [37, 1, 43, 34, 55, 59]. Despite the remarkable ability of deep networks to achieve linear separation, the underlying mechanisms by which they accomplish this—especially when the input data are initially poorly separated—remain largely unclear. Investigating this phenomenon could significantly improve the interpretability of deep learning models and provide deeper insights into their generalization capabilities. Before presenting our main contribution, we provide a brief review of the existing results.

Empirical studies on linear separability of initial-layer features. Recent empirical studies investigated the role of the intermediate layers in deep nonlinear networks, e.g., [1, 4, 44, 20, 62, 55, 59, 34, 30]. These studies indicate that the initial layers expand the features such that they become linearly separable between classes (see Figure 1). For instance, in image classification, Alain and Bengio [1], Masarczyk et al. [34], Wang et al. [55] observed that linear probing accuracy improves significantly across the initial layers of neural networks, while the deep layers mainly compress within-class features. This implies that the initial layers play a critical role in achieving linear separability of the input data.

Theoretical works on linear separability of initial-layer features. To the best of our knowledge, theoretical studies on the linear separability of features across nonlinear layers in DNNs are quite limited. Recent works [15, 18] have studied the separability of features from initial ReLU layers.



(a) 6-layer MLP (b) 6-layer hybrid network

Figure 1: **Linear separability and compression of features across layers.** The initial layers transform the input to be linearly separable, while the deeper layers compress the features. Following the setup in [55], we trained two networks on CIFAR-10: a 6-layer multi-layer perceptron (MLP, left) and a 6-layer hybrid network (a 3-layer MLP followed by a 3-layer linear network, right), both with hidden dimensions of 1024. For each trained network, we conducted linear probing on the features from each layer. At each layer, we recorded the linear probe accuracy and the numerical rank of the feature matrix, defined as the minimum number of singular values accounting for at least 95% of the nuclear norm, and plotted these results.

These studies rigorously showed that the features extracted from a two-layer [15] and one-layer [18] random ReLU network are linearly separable for arbitrary input data. However, a key limitation of these works is that in the worst case, the required network width grows *exponentially* with respect to (w.r.t.) the ambient dimension of the data. Consequently, the network sizes required by theoretical analyses are substantially larger than those typically used in real-world applications, highlighting a significant gap between theory and practice.

Theoretical studies of representation learning in deep linear networks. Another line of research has explored how deep *linear* networks (DLNs) progressively compress within-class features and discriminate between-class features [48, 55]. Specifically, building on the empirical observation that linear layers can emulate the behavior of deeper layers in nonlinear networks, Wang et al. [55] provided a theoretical analysis of the progressive feature compression in DLNs, under the assumption that the input data are already linearly separable. However, due to this restrictive data assumption, the study cannot fully explain the structures of hierarchical representation in nonlinear networks, particularly *how* the early layers transform input features to achieve linear separability due to the nonlinear operators.

1.1 Our Contributions

In this work, we investigate the linear separability of features in *shallow* nonlinear networks for *low-dimensional data*, closing the gap between theory and practice of representation learning in the initial layers of nonlinear networks [34]. Specifically, we observe:

> *A single nonlinear layer with random weights transforms data from a union of low-dimensional subspaces into linearly separable sets.*

We rigorously prove this result with $K = 2$ subspaces and discuss how the result can be extended

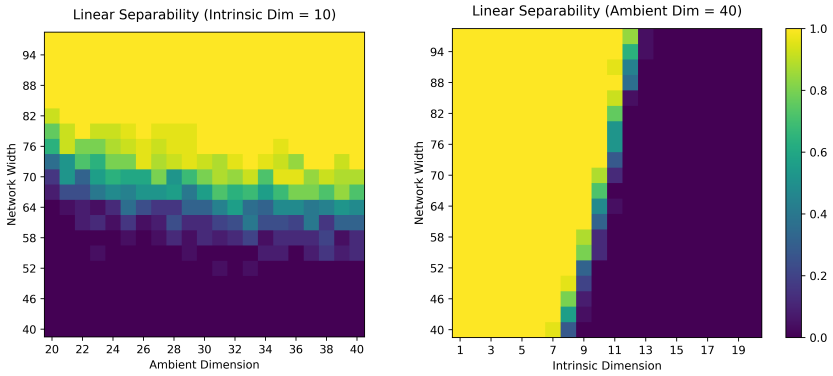


Figure 2: **Phase transition of linear separability w.r.t. dimensions** (d, r) **and network width** D**.** We demonstrate that the network width required to achieve linear separability of a union of two subspaces scales polynomially with the intrinsic dimension. See Section 4.1 for details.

to $K > 2$ subspaces. In our analysis, we assume that the activation is quadratic and the first-layer weights are random. The resulting width of the network scales *polynomially* w.r.t. the intrinsic dimension of the subspaces. Moreover, our results empirically hold under more generic settings. For example, we can replace the quadratic activation function with other nonlinear activations, such as ReLU, and still achieve linear separability (see Section 4). Additionally, the widths of ReLU and quadratic activation layers have similar dependence on the intrinsic dimension and number of subspaces to achieve linear separability (see Section 2 and Figure 6).

Our results complement previous work [55], providing a comprehensive theoretical understanding of how input data is transformed across the layers in DNNs. Our findings also offer insights into the role of overparameterization in deep representation learning and explain why learning based upon random features can lead to good in-distribution generalization.

1.2 Notation and Paper Organization

Before delving into the technical discussion, we introduce the notation used throughout the paper and outline its organization.

Notation. For a positive integer N, we use $[N]$ to denote the index set $\{1, 2, \ldots, N\}$. We use $\mathcal{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2, and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. We use $\|\cdot\|$ to denote the Euclidean norm of a vector, $\mathbf{0}_m$ to denote an m-dimensional vector of all zeros, $\lambda_i(\cdot)$ to denote the ith largest eigenvalue of a symmetric matrix, and $\sigma_i(\cdot)$ to denote the ith largest singular value of a matrix. With a slight abuse of notation, for some function ϕ and set $\mathcal{X}$, $\phi(\mathcal{X})$ denotes the set $\{\boldsymbol{x} \in \mathcal{X} : \phi(\boldsymbol{x})\}$. Unless otherwise stated, the term "subspace" implies a linear subspace embedded in Euclidean space.

Organization. The rest of this paper is organized as follows. We motivate the union of subspaces (UoS) data model, introduce our problem setting, and motivate our theoretical assumptions in Section 2. We then state our main theoretical result and provide a proof sketch in Section 3, with the full proof in Appendix B. We provide supporting results for our proof in Appendix A. In Section 4, we provide empirical evidence supporting our theoretical result, and investigate settings not considered in our analysis. Finally, we discuss related results and conclude in Section 5.

2 Preliminaries

In this section, we introduce the basic problem setup and motivations. First, we introduce the UoS model for our input data in Section 2.1, and then discuss the choices of the network in Section 2.2.

2.1 Assumptions on Input Data

Recent empirical studies indicate real-world image data typically possess a significantly lower *intrinsic* dimension than their ambient dimension. For instance, Pope et al. [39] used a nearest-neighbor approach to estimate the intrinsic dimension of many popular image datasets, including MNIST [28], CIFAR-10 [26], and ImageNet [46]. They showed the intrinsic dimension of these datasets is at most around 40, even though the images themselves contain thousands of pixels. Furthermore, Brown et al. [10] used a similar approach to show *each class* has its own low intrinsic dimensionality. These results indicate image data lie on *a union of low-dimensional manifolds* within high-dimensional space. Similar models have recently been explored for understanding generative models [57, 13].

Although low-dimensional manifolds can exhibit complex structures, each manifold can be locally approximated by its tangent space, which is a linear subspace embedded within the ambient space. This motivates us to initiate our study with a simplified model: a union of low-dimensional subspaces that capture the local structures of manifolds. For simplicity in our theoretical analysis, we focus on the case of $K = 2$ subspaces, which facilitates a clearer exposition. Nonetheless, our results extend to the case with $K > 2$ subspaces, as discussed in Section 3.1. To set the stage for our analysis, we first introduce a generic definition of a union of K subspaces.

Definition 1 (Union of K Low-Dimensional Subspaces)**.** *Let $\mathcal{S}_1, \mathcal{S}_2, \dots \mathcal{S}_K \subseteq \mathbb{R}^d$ be K linear subspaces with dimensions $r_1, r_2, \dots, r_K$, respectively. Let $\boldsymbol{U}_k \in \mathbb{R}^{d \times r_k}$ denote the orthonormal basis matrix of $\mathcal{S}_k$ for each $k \in [K]$. We say that a data point $\boldsymbol{x} \in \mathbb{R}^d$ lies on the union of subspaces $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K$ if*

$$\boldsymbol{x} \in \bigcup_{k=1}^{K} \mathcal{S}_k := \left\{ \boldsymbol{z} \in \mathbb{R}^d : \exists k \in [K] \ \text{ s.t. } \ \boldsymbol{z} = \boldsymbol{U}_k \boldsymbol{\alpha} \ \text{ for some } \ \boldsymbol{\alpha} \in \mathbb{R}^{r_k} \right\}. \tag{1}$$

The *principal angle between two subspaces* can be viewed as a generalization of the angles between two vectors (i.e., two one-dimensional subspaces). For any two subspaces $(\mathcal{S}_1, \mathcal{S}_2)$ of dimensions r_1 and r_2, there exist $\min\{r_1, r_2\}$ principal angles between them, which are formally defined as follows.

Definition 2 (Principal angles between two subspaces)**.** *Suppose that the columns of $\boldsymbol{U}_1 \in \mathbb{R}^{d \times r_1}$ and $\boldsymbol{U}_2 \in \mathbb{R}^{d \times r_2}$ are orthonormal bases for subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Let $r := \min\{r_1, r_2\}$. The ℓth principal angle $\theta_\ell \in [0, \pi/2]$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ is defined as*

$$\cos(\theta_\ell) \ := \ \sigma_\ell(\boldsymbol{U}_1^\top \boldsymbol{U}_2),$$

for all $\ell \in [r]$.

The principal angle is illustrated in Figure 3. By the above definition, since $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_r \le \pi/2$, we will sometimes use $\theta_{\min}$ to denote θ_1. If m of the r principal angles between $\mathcal{S}_1$ and $\mathcal{S}_2$ are zero (i.e., $\theta_1 = \theta_2 = \cdots = \theta_m = 0$), then the *intersection* $\mathcal{S}_1 \cap \mathcal{S}_2$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ is also

a linear subspace, but of dimension m. If $m = 0$, then the intersection between $\mathcal{S}_1$ and $\mathcal{S}_2$ is just the origin $\{\mathbf{0}_d\}$. Building on these definitions, we will make the following assumption on the UoS model for our analysis in Section 3.

Assumption 1. *We consider $K = 2$ subspaces $\mathcal{S}_1, \mathcal{S}_2$ with equal dimensions, i.e., $r_1 = r_2 := r$. Furthermore, the principal angles between $\mathcal{S}_1$ and $\mathcal{S}_2$ are strictly positive, i.e., $0 < \theta_1 \leq \theta_2 \leq \cdots \leq \theta_r \leq \pi/2$.*

Remarks. We discuss Assumption 1 in the following.

- **Number of subspaces.** Although we assume $K = 2$ subspaces to simplify both the analysis and exposition, the results can be generalized to the K-subspaces setting with $K > 2$, which we discuss in Corollary 1 of Section 3 in detail.

- **Subspace dimensions.** Here, we assumed equal dimensionality for each subspace for simplicity. In practice, each subspace of the UoS model can have different dimensions, which we believe our result can be generalized to this setting as well. We leave detailed analysis for future work.

- **Principal angles between subspaces.** We assume none of the principal angles are equal to zero to ensure $\mathcal{S}_1 \cap \mathcal{S}_2 = \{\mathbf{0}_d\}$. Otherwise, it is impossible to label the intersected points in the nonempty set $\mathcal{S}_1 \cap \mathcal{S}_2$. Additionally, it should be noted that $\theta_1 > 0$ if and only if $r < d/2$. This assumption is typically satisfied in practice, as usually $r \ll d$.

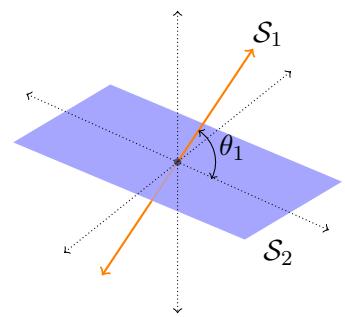


Figure 3: **The principal angle between a 1-dim subspace $\mathcal{S}_1$ and 2-dim subspace $\mathcal{S}_2$.**

2.2 Linear Separability of UoS via Nonlinear Networks

In this work, we investigate how nonlinear neural networks separate the data that follows the UoS model. Specifically, we consider a shallow neural network $f_{\boldsymbol{W}}(\boldsymbol{x}) : \mathbb{R}^d \mapsto \mathbb{R}^D$, which can be viewed as a feature mapping from the input space $\mathbb{R}^d$ to a feature space $\mathbb{R}^D$:

$$f_{\boldsymbol{W}}(\boldsymbol{x}) \;=\; \sigma(g_{\boldsymbol{W}}(\boldsymbol{x})) \;=\; \sigma(\boldsymbol{W}\boldsymbol{x}). \tag{2}$$

Here, $g_{\boldsymbol{W}}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$, $\boldsymbol{W} \in \mathbb{R}^{D\times d}$ is the weight matrix, and $\sigma(\cdot)$ is an entry-wise nonlinear activation function. Although the weight matrix $\boldsymbol{W}$ of the neural network is often learned by training on some dataset using a loss function, such as cross-entropy, we consider a *random feature model* where $\boldsymbol{W}$ is fixed and each entry is drawn from some random distribution. As illustrated in Figure 4, based on the above setup, we are interested in the following problem:

Problem 1. Consider a union of two subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ that satisfy Assumption 1. Under what conditions does there exist a separating hyperplane $\boldsymbol{v} \in \mathbb{R}^D$ such that

$$\boldsymbol{v}^\top f_{\boldsymbol{W}}(\boldsymbol{U}_1\boldsymbol{\alpha}) > 0 \;\text{ and }\; \boldsymbol{v}^\top f_{\boldsymbol{W}}(\boldsymbol{U}_2\boldsymbol{\alpha}) < 0 \tag{3}$$

for all $\boldsymbol{\alpha} \in \mathbb{R}^r \setminus \{\mathbf{0}_r\}$?

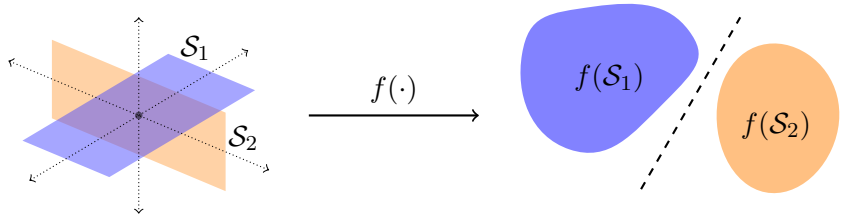


Figure 4: **An illustration of Problem 1.** We aim to find conditions on the network f so a union of subspaces (left) transforms into linearly separable sets (right).

Essentially, our focus is on the linear separability of random features derived from the UoS model. As elaborated below, Problem 1 remains challenging even under the simplified setup considered here. Generally, data from two distinct subspaces are not inherently linearly separable, and neither a linear mapping $g(\boldsymbol{x})$ nor nonlinear activations $\sigma(\cdot)$ alone are sufficient to transform such data into linearly separable sets.

- **Subspaces are not linearly separable in general.** This can be easily shown by a counter-example that we illustrate below. Suppose we have two one-dimensional subspaces $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^2$ with bases $\boldsymbol{u}_1 = \begin{bmatrix}1 & 1\end{bmatrix}^\top$ and $\boldsymbol{u}_2 = \begin{bmatrix}-1 & 1\end{bmatrix}^\top$, respectively. As shown in Figure 5, there does not exist any hyperplane (line) that can separate $\mathcal{S}_1$ and $\mathcal{S}_2$ because they both pass through the origin.

- **Linear mapping alone is insufficient for linear separability.** We first show $g_{\boldsymbol{W}}(\mathcal{S}_1)$ and $g_{\boldsymbol{W}}(\mathcal{S}_2)$ are not linearly separable sets. For any $k \in \{1, 2\}$ and $\boldsymbol{\alpha}^{(k)} \in \mathbb{R}^{r_k}$, we have

$$g_{\boldsymbol{W}}(\boldsymbol{U}_k \boldsymbol{\alpha}^{(k)}) = \boldsymbol{W}\boldsymbol{U}_k \boldsymbol{\alpha}^{(k)} = \tilde{\boldsymbol{U}}_k \boldsymbol{\alpha}^{(k)},$$

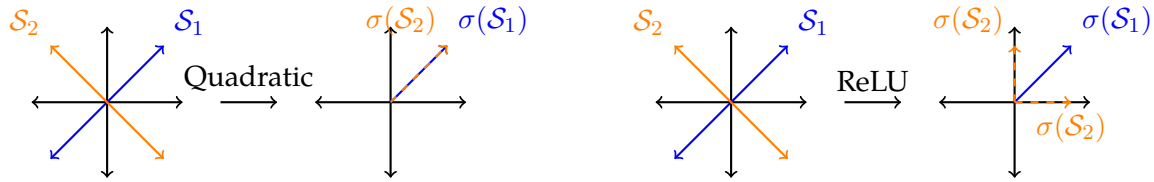
  where $\tilde{\boldsymbol{U}}_k \in \mathbb{R}^{D \times r_k}$. Therefore, the point $g_{\boldsymbol{W}}(\boldsymbol{U}_k \boldsymbol{\alpha}^{(k)})$ lies in an r_k-dimensional subspace in $\mathbb{R}^D$. Since this holds for all $\boldsymbol{z}^{(k)} \in \mathbb{R}^{r_k}$, the sets $g_{\boldsymbol{W}}(\mathcal{S}_1)$ and $g_{\boldsymbol{W}}(\mathcal{S}_2)$ remain to be linear subspaces of $\mathbb{R}^D$ that pass through the origin, which is not linearly separable in general.

- **Nonlinear activations alone are insufficient for linear separability.** Second, for various activation functions, we show $\sigma(\mathcal{S}_1)$ and $\sigma(\mathcal{S}_2)$ are not linearly separable sets through counterexamples. Again, suppose that the bases of $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^2$ are $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, respectively. Let us first consider the entry-wise quadratic activation. Note that we consider the quadratic activation in our theoretical analysis later on. For any $\alpha \in \mathbb{R}$, we have

$$\sigma(\boldsymbol{u}_1 \alpha) = \begin{bmatrix}1^2 \\ 1^2\end{bmatrix} \alpha^2 = \begin{bmatrix}\alpha^2 \\ \alpha^2\end{bmatrix}, \quad \sigma(\boldsymbol{u}_2 \alpha) = \begin{bmatrix}(-1)^2 \\ 1^2\end{bmatrix} \alpha^2 = \begin{bmatrix}\alpha^2 \\ \alpha^2\end{bmatrix},$$

  so $\sigma(\boldsymbol{u}_1 \alpha) = \sigma(\boldsymbol{u}_2 \alpha)$. After applying the quadratic function, two points in $\mathcal{S}_1$ and $\mathcal{S}_2$ with the same coefficient α *cannot be distinguished from each other*, implying the sets $\sigma(\mathcal{S}_1)$ and $\sigma(\mathcal{S}_2)$ are *identical* (see Figure 5, left).

  Next, consider $\sigma(\cdot) = \mathrm{ReLU}(\cdot)$, which is more commonly used in practice. For any nonzero $\alpha \in \mathbb{R}$, $\sigma(\boldsymbol{u}_1 \alpha) = \boldsymbol{u}_1 \alpha$ if $\alpha > 0$, and $\sigma(\boldsymbol{u}_1 \alpha) = \boldsymbol{0}_2$ if $\alpha < 0$. Additionally, $\sigma(\boldsymbol{u}_2 \alpha) = \begin{bmatrix}0 & \alpha\end{bmatrix}^\top$ if $\alpha > 0$, and $\sigma(\boldsymbol{u}_2 \alpha) = \begin{bmatrix}-\alpha & 0\end{bmatrix}^\top$ if $\alpha < 0$. Therefore, $\sigma(\mathcal{S}_1) = \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 > 0, x_2 > 0\} \cup \{\boldsymbol{0}_2\}$, while $\sigma(\mathcal{S}_2) = \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 > 0, x_2 = 0\} \cup \{\boldsymbol{x} \in \mathbb{R}^2 : x_1 = 0, x_2 > 0\}$, where x_1 and x_2 respectively denote the first and second elements of $\boldsymbol{x} \in \mathbb{R}^2$. These sets are *not* linearly separable (see Figure 5, right).



(a) Quadratic applied to $\mathrm{span}(\boldsymbol{u}_1) \cup \mathrm{span}(\boldsymbol{u}_2)$. (b) ReLU applied to $\mathrm{span}(\boldsymbol{u}_1) \cup \mathrm{span}(\boldsymbol{u}_2)$.

Figure 5: **Activation alone is insufficient for linearly separating two subspaces.** When $\mathcal{S}_1 = \mathrm{span}(\boldsymbol{u}_1)$ and $\mathcal{S}_2 = \mathrm{span}(\boldsymbol{u}_2)$, the sets $\sigma(\mathcal{S}_1)$ and $\sigma(\mathcal{S}_2)$ are not linearly separable for $\sigma(\cdot) =$ quadratic (left) and $\sigma(\cdot) = \mathrm{ReLU}(\cdot)$ (right).

Thus, to tackle Problem 1, we must *jointly* apply a linear mapping and a nonlinear transformation to achieve linear separability of the subspaces. Specifically, we now introduce the following assumptions on the network (2), based upon which we characterize the sufficient conditions for achieving linear separability in Section 3.

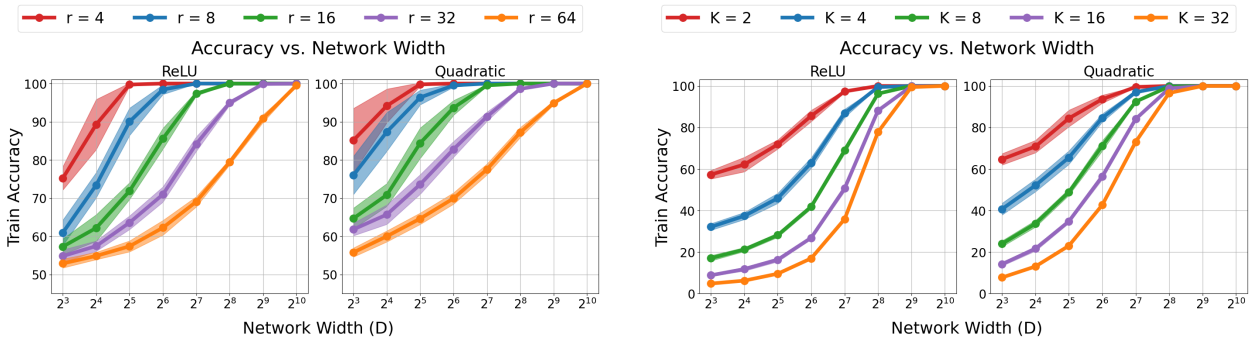
Assumption 2. *For the feature mapping $f_{\boldsymbol{W}}(\boldsymbol{x})$ in (2), we assume that the activation function $\sigma(\cdot)$ is the quadratic (entry-wise square) function, and the entries of $\boldsymbol{W}$ are independent and identically distributed (iid) standard Gaussian, i.e., $W_{ij} \overset{iid}{\sim} \mathcal{N}(0,1)$ for all $(i,j) \in [D] \times [d]$.*

Remarks. We briefly discuss Assumption 2 below.

- **Quadratic activation.** In this work, we consider the quadratic activation due to its smoothness and simplicity. Such activations have also been considered in many previous theoretical results of analyzing nonlinear networks [31, 51, 16, 17, 47] (see also Section 5.1). Moreover, we believe the results can be extended to many other nonlinear activations, such as ReLU. In Section 4, we empirically show if one replaces the quadratic activation with other activations, the output features from (2) are still linearly separable under a UoS data model. Additionally, we empirically observe the required width to achieve linear separability scales similarly with the intrinsic dimension and the number of classes under both ReLU and quadratic activations (see Figure 6).

- **Random weights.** Assumption 2 yields a random feature model, which has been widely studied in the literature, e.g., [41, 42, 45, 7, 32] (see [33] for a survey). Moreover, it can also shed light on trained DNNs. For example, in the infinite-width limit [23, 6, 12, 2] random networks behave similarly to fully-trained networks. This is called the Neural Tangent Kernel (NTK) [23] regime, where the random initialization determines the NTK and the NTK remains constant during training [23]. Furthermore, the Neural Network Gaussian Process kernel (NNGP) [29] is the kernel associated with a network at random initialization. Recently, [25] studied Neural Collapse (NC) [37] of nonlinear networks from a kernel perspective. They showed that NNGP and NTK exhibited similar amounts of NC.

  Even for networks with finite width, we empirically observe if the initial-layer features under a UoS data model are linearly separable at random initialization, pushing the layer weights away from their randomly initialized values via training *does not impact the linearly separability of these features* (see Figure 7). This could be partially explained by recent results [60, 36], showing that training only happens within an invariant subspace of the weights.



(a) Sweeping $r \in \{4, 8, 16, 32, 64\}$.

(b) Sweeping $K \in \{2, 4, 8, 16, 32\}$.

Figure 6: **The effects of different activations on linear separability.** ReLU and quadratic activations exhibit similar width requirements w.r.t. the intrinsic dimension (left) and number of subspaces (right) for achieving linear separability. See Section 4.2 for details.

3 Theoretical Results

We first state our main theoretical results and their implications in Section 3.1, and correspondingly provide a sketch of the proof in Section 3.2.

3.1 Main Results

First, we state our main theoretical result in the binary case $K = 2$, and then we generalize the result to multiple subspaces $K > 2$ in Section 3.1.1.

Theorem 1 (Linear Separability of $f(\mathcal{S}_1)$ and $f(\mathcal{S}_2)$)**.** *Suppose Assumption 1 and Assumption 2 hold, and let $\delta \in (0, 1)$. If the network width D satisfies*

$$D \geq \frac{2\pi\left(4r^2 + \sqrt{\sum_{\ell=1}^{r} \sin^2(\theta_\ell)}\right)(r+1)}{\sin^2(\theta_{min})} \cdot \log\left(\frac{2r}{\delta}\right), \tag{4}$$

then the sets $f(\mathcal{S}_1)$ and $f(\mathcal{S}_2)$ are linearly separable with probability at least $1 - \delta$ w.r.t. the randomness of $\boldsymbol{W}$.

In short, Theorem 1 states that the nonlinear feature model can transform a union of two subspaces into linearly separable sets, given that the network width scales in *polynomial* with the intrinsic dimension of the subspaces. We discuss the implications of our result below.

Requirement of the network width. Our findings demonstrate that significantly fewer neurons are needed to achieve linear separability of initial-layer features compared to previous studies. Specifically, [15] and [18] showed that one- and two-layer random-ReLU networks can linearly separate nonlinearly-separated classes, but their required network widths scale exponentially with the ambient dimension for unstructured classes, or intrinsic dimensions for classes on a union of subspaces. These exponential scaling requirements are much larger than those used in practical DNNs, limiting their applicability. In contrast, Theorem 1 only requires network widths to scale

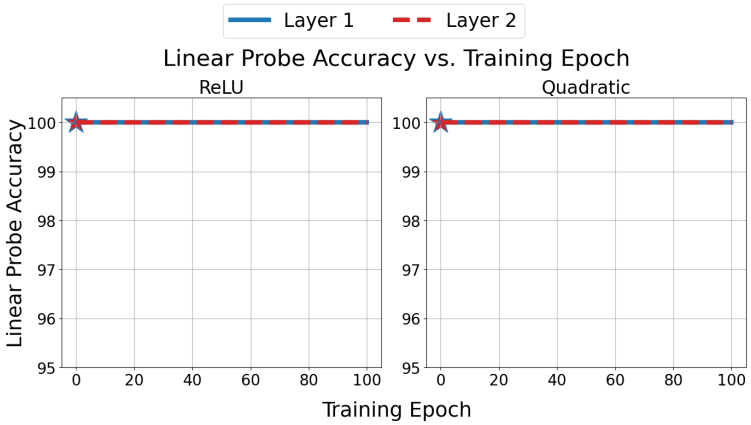


Figure 7: **Linear separability of nonlinear features through the training dynamics.** If the initial-layer features are linearly separable at random initialization, they remain linearly separable throughout training. See Section 4.2 for details.

polynomially with the intrinsic dimension, aligning more closely with real-world network sizes.[1] Therefore, our results provide a more accurate characterization of how initial layers in practical DNNs create linearly separable features from raw data.

Additionally, DNNs are often *overparameterized*, i.e., the number of parameters is larger than the number of training samples N, but Theorem 1 states a nonlinear layer of $\Omega\big(\mathrm{poly}(r)\big)$ width is needed to transform the subspaces into linearly separable sets. Since typically $r \ll N$, our result implies an *underparameterized* network can transform the subspaces into linearly separable sets. Although overparameterizaion yields many optimization and generalization benefits, it may not be necessary specifically for making raw input features linearly separable at the shallow layers. Thus, in deep representation learning, overparameterization may be more critical in compressing the features at the deeper layers.

Complete understanding of deep representation learning across layers. Previous work [55] explored *feature compression* in deeper layers of DNNs, showing empirically that linear layers mimic deeper nonlinear layers and theoretically that DLNs compress features at a geometric rate, assuming the input features are already linearly separable. However, these findings do not address how initial nonlinear layers transform raw data into linearly separable features. In contrast, our result characterizes the initial layer features, showing that a *single* nonlinear layer can achieve linear separability under a UoS data model. Together, these results provide a theoretical understanding of feature transformations across the entire depth of DNNs.

In-distribution generalization when learning with random features. Our result also provides insight into in-distribution generalization when learning with random features. Theorem 1 states a single random nonlinear layer makes *all points* in the two subspaces linearly separable. Suppose we have a dataset with N train samples lying on a union of two subspaces, apply a nonlinear random feature map with $\Omega(r^3 \cdot \log(Nr))$ features and quadratic activation, and train a linear classifier on the random features to classify the train samples. If the test samples are in-distribution, i.e., they

[1] For example, with the CIFAR-10 dataset having an intrinsic dimension of approximately 25 [39], previous methods would need a network width around $\exp(25)$, whereas our theorem requires a width of about $25^3 \cdot \log(25)$—a much more feasible size.

lie in the same subspaces as the train samples, then the trained classifier will also achieve perfect test accuracy with probability at least $1 - \frac{1}{N}$.

3.1.1 Extension to Multiple Subspaces

While Theorem 1 assumed $K = 2$ subspaces, we can generalize Theorem 1 to $K > 2$ subspaces as follows.

Corollary 1. *Suppose there are $K > 2$ subspaces each of dimension r, where $(K-1)r < d/2$. For all $k \in [K]$, let $\tilde{\mathcal{S}}_k \supset \mathcal{S}_k$ and $\overline{\mathcal{S}}_k \supset \bigcup_{j\in[K], j\neq k} \mathcal{S}_j$ be $\tilde{r}$-dimensional subspaces with principal angles $\theta_{k,\ell}, \theta_{k,2}, \ldots, \theta_{k,\tilde{r}}$ that satisfy Assumption 1, where $\tilde{r} := (K-1)r$. Also we assume that Assumption 2 holds and $\delta \in (0,1)$. If the network width D satisfies*

$$D \geq \max_{k\in[K]} \left\{ \frac{2\pi}{\sin^2(\theta_{k,min})} \left(4\tilde{r}^2 + \sqrt{\sum_{\ell=1}^{\tilde{r}} \sin^2(\theta_{k,\ell})} \right) \left(\tilde{r} + 1 \right) \right\} \cdot \log\left(\frac{2K\tilde{r}}{\delta} \right), \tag{5}$$

then the sets $f(\mathcal{S}_k)$ and $f\left(\bigcup_{j\in[K], j\neq k} \mathcal{S}_j\right)$ are linearly separable for all $k \in [K]$ with probability at least $1 - K\delta$ w.r.t. the randomness of $\boldsymbol{W}$.

Corollary 1 states if the layer width scales in polynomial order w.r.t. both the intrinsic dimension *and* the number of subspaces, the nonlinear features are one-vs.-all separable: each individual subspace is separated from *all* of the remaining subspaces. In contrast, Theorem 1 only depends on the intrinsic dimension, as it only considers the binary subspaces setting.

3.2 Proof Sketches

In the following, we first provide a proof sketch of Theorem 1 for binary subspaces $K = 2$, and later we generalize the analysis to multiple subspaces $K > 2$.

Proof sketch for Theorem 1. We first provide a proof sketch of Theorem 1. We defer the full proof to Appendix B. Let $\boldsymbol{X} := \boldsymbol{W}\boldsymbol{U}_1 \in \mathbb{R}^{D\times r}$ and $\boldsymbol{Y} := \boldsymbol{W}\boldsymbol{U}_2 \in \mathbb{R}^{D\times r}$, and let $\boldsymbol{x}_n \in \mathbb{R}^r$ and $\boldsymbol{y}_n \in \mathbb{R}^r$ denote the nth row vectors of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Note $\boldsymbol{x}_n = \boldsymbol{U}_1^\top \boldsymbol{w}_n$ and $\boldsymbol{y}_n = \boldsymbol{U}_2^\top \boldsymbol{w}_n$, where $\boldsymbol{w}_n \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$ denotes the nth row in $\boldsymbol{W}$. First, under Assumption 2, (3) holds if and only if there exists a vector $\boldsymbol{v} \in \mathbb{R}^D$ such that

$$\sum_{n=1}^{D} v_n \boldsymbol{x}_n \boldsymbol{x}_n^\top \succ 0 \;\text{ and }\; \sum_{n=1}^{D} v_n \boldsymbol{y}_n \boldsymbol{y}_n^\top \prec 0. \tag{6}$$

Next, we are interested in the *existence* of a hyperplane $\boldsymbol{v}$ that separates the random features, which is not necessarily a max-margin hyperplane. Thus, we choose a linear classifier $\boldsymbol{v}$ with the following entries:

$$\textit{For all } n \in [D],\ v_n = \operatorname{sign}\big(\|\boldsymbol{x}_n\|^2 - \|\boldsymbol{y}_n\|^2\big).$$

This choice of $\boldsymbol{v}$ is a *projection-based classifier*: the subspace onto which $\boldsymbol{w}_n$ has the largest projection determines the sign of v_n. If $\|\boldsymbol{U}_1^\top \boldsymbol{w}_n\|^2 > \|\boldsymbol{U}_2^\top \boldsymbol{w}_n\|^2$, then we set $v_n = +1$ to push the inner product $\boldsymbol{v}^\top f_{\boldsymbol{W}}(\boldsymbol{U}_1\boldsymbol{\alpha})$ to be "more positive" for any $\boldsymbol{\alpha} \in \mathbb{R}^r$. Likewise, setting $v_n = -1$ when $\|\boldsymbol{U}_1^\top \boldsymbol{w}_n\|^2 < \|\boldsymbol{U}_2^\top \boldsymbol{w}_n\|^2$ pushes $\boldsymbol{v}^\top f_{\boldsymbol{W}}(\boldsymbol{U}_2\boldsymbol{\alpha})$ to be "more negative" for any $\boldsymbol{\alpha} \in \mathbb{R}^r$. Since $\|\boldsymbol{U}_k^\top \boldsymbol{w}_n\|^2 \sim \chi_r^2$ for

all $k \in \{1,2\}$, $\|\boldsymbol{U}_1^\top \boldsymbol{w}_n\|^2 = \|\boldsymbol{U}_2^\top \boldsymbol{w}_n\|^2$ occurs with probability zero under Assumption 1. With this choice of $\boldsymbol{v}$, (6) is equivalent to

$$\boldsymbol{Q}_1 := \sum_{i\in\mathcal{I}} \boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{x}_j\boldsymbol{x}_j^\top \succ 0 \text{ and } \boldsymbol{Q}_2 := \sum_{i\in\mathcal{I}} \boldsymbol{y}_i\boldsymbol{y}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{y}_j\boldsymbol{y}_j^\top \prec 0, \tag{7}$$

where $\mathcal{I} := \{n : v_n = +1\}$ and $\mathcal{I}^c := \{n : v_n = -1\}$.

We now wish to upper bound the *failure probability* $P\Big(\boldsymbol{Q}_1 \not\succ 0 \cup \boldsymbol{Q}_2 \not\prec 0\Big)$. Note $\boldsymbol{Q}_1 \not\succ 0$ if and only if $\lambda_r(\boldsymbol{Q}_1) \leq 0$, and $\boldsymbol{Q}_2 \not\prec 0$ if and only if $\lambda_1(\boldsymbol{Q}_2) \geq 0$. Therefore, upper bounding the failure probability is equivalent to upper bounding $P\Big(\lambda_r(\boldsymbol{Q}_1) \leq 0 \cup \lambda_1(\boldsymbol{Q}_2) \geq 0\Big)$. Next, we show $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are sums of sub-exponential random matrices[2], which allows us to use Bernstein's matrix inequality (Theorem 6.2 in [53]) to obtain upper bounds on $P\Big(\lambda_r(\boldsymbol{Q}_1) \leq 0\Big)$ and $P\Big(\lambda_1(\boldsymbol{Q}_2) \geq 0\Big)$. Applying the union bound by some constant $\delta \in (0,1)$, and then re-arranging the appropriate terms to lower bound D, leads to the result in Theorem 1.

Extension to $K > 2$ subspaces in Corollary 1. We now present a proof sketch for Corollary 1, omitting the full details as it directly follows from an application of Theorem 1. Note we assume $(K-1)r < d/2$. This assumption is not very limiting when the number of classes is small, since K and r are typically much smaller than d in practice.

Let $k \in [K]$ be arbitrary, and let $\overline{\mathcal{S}}_k := \mathcal{R}\Big(\begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2 & \dots & \boldsymbol{U}_{k-1} & \boldsymbol{U}_{k+1} & \dots & \boldsymbol{U}_K\end{bmatrix}\Big)$ be an $\tilde{r}$-dimensional subspace, where $\tilde{r} = (K-1)r$ and $\mathcal{R}(\cdot)$ denotes the column space of a matrix. Note $\overline{\mathcal{S}}_k \supset \bigcup_{j=1,j\neq k}^{K} \mathcal{S}_j$. Also let $\tilde{\mathcal{S}}_k$ denote an $\tilde{r}$-dimensional subspace $\tilde{\mathcal{S}}_k \supset \mathcal{S}_k$ such that $\tilde{\mathcal{S}}_k$ and $\overline{\mathcal{S}}_k$ satisfy Assumption 1. Such a $\tilde{\mathcal{S}}_k$ exists if $(K-1)r < d/2$. Since $\mathcal{S}_k \subset \tilde{\mathcal{S}}_k$ and $\bigcup_{j\in[K],j\neq k} \mathcal{S}_j \subset \overline{\mathcal{S}}_k$, it suffices to transform $\overline{\mathcal{S}}_k$ and $\tilde{\mathcal{S}}_k$ into linearly separable sets.

We directly apply Theorem 1 to transform $\overline{\mathcal{S}}_k$ and $\tilde{\mathcal{S}}_k$, and thus $\mathcal{S}_k$ and $\bigcup_{j\in[K],j\neq k} \mathcal{S}_j$, into linearly separable sets with high probability. Since this is now a problem of separating two $\tilde{r}$-dimensional subspaces, the r in Theorem 1 becomes $\tilde{r}$. Applying the Union Bound over all $k \in [K]$, a random nonlinear layer of width $\Omega\big(\text{poly}(Kr)\big)$ transforms a union of K subspaces into K one-vs-all linearly separable sets with high probability.

4 Experimental Results

In this section, we empirically verify a single random nonlinear layer makes a UoS linearly separable for both synthetic and real-world data. Specifically, in Sections 4.1 and 4.2, we describe the experimental setups for Figures 2, 6 and 7, verify our main results Theorem 1 and corollary 1, and explore settings beyond our assumptions on synthetic data in Figure 8. In Section 4.3, we provide experimental results on CIFAR-10 again supporting our theoretical results.

4.1 Phase Transition in Terms of Intrinsic Dimension

In this subsection, we describe the setup and results in Figure 2, which verifies the required network width to achieve linear separability of the initial-layer features grows polynomially w.r.t. the intrinsic dimension.

[2]Here, sub-exponential random matrices refer to random matrices whose higher-order moments are analogous to the higher-order moments of sub-exponential random variables. See Theorem 6.2 in [53].

Experimental setup. Over 25 trials, we randomly sampled two matrices $\boldsymbol{U}_1, \boldsymbol{U}_2$ from the $d \times r$ Stiefel manifold, and a weight matrix $\boldsymbol{W} \in \mathbb{R}^{D \times d}$ with iid standard Gaussian entries. We varied the ambient dimension d while keeping the intrinsic dimension r fixed, and also varied r while keeping d fixed. In both scenarios, we tested different layer widths D. For combination of (D, d) and (D, r), we checked for linear separability using the necessary and sufficient conditions (6), and recorded the proportion of successful trials.

Experimental results. As seen in Figure 2, when d increases for a fixed r, the values of D at which the proportion of successful trials transitions from 0 to 1, or the *phase transition*, remains constant. In contrast, as r increases for a fixed d, this phase transition region clearly increases. Thus, Figure 2 verifies the required width to achieve linear separability of the random features only depends on the intrinsic dimension of the subspaces.

4.2 Classification on Synthetic Data following a UoS Model

In this subsection, we verify the initial-layer features are linearly separable for synthetic data generated from the UoS model under various settings, including different nonlinear activations $\sigma(\cdot)$, network widths D, and the number of subspaces K. In the following experiments, we generated synthetic data from a UoS using the following process.

Synthetic data generation. We first generated K matrices $\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_K$ uniformly at random from the $d \times r$ Stiefel manifold. We then generated $N = K \cdot N_k$ training samples as follows, where $N_k = 5 \cdot 10^3$. For all $k \in [K]$, we created N_k samples via $\boldsymbol{x}_{k,i} = \boldsymbol{U}_k \boldsymbol{z}_i$, where $\boldsymbol{z}_i$ were sampled iid from $\mathcal{N}(\boldsymbol{0}_r, \boldsymbol{I}_r)$ for all $i \in [N_k]$. When applicable, we then generated N test samples using the same procedure.

4.2.1 Linear Separability of Features: Random vs. Trained Weights

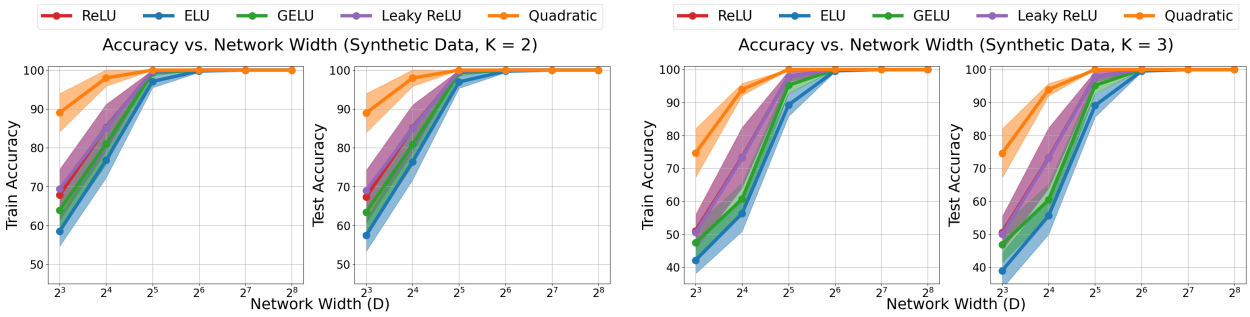
We first describe the setup and results in Figure 7, which investigates how training the network weights away from their random initialization impacts the linear separability of the initial-layer features.

Experimental setup. We first created a training set using the above data generation process with $K = 2$, $d = 16$, and $r = 4$. We then trained two 3-layer MLPs of width $D = 128$ for 100 epochs. One MLP had ReLU activations, and the other had quadratic activations. After each training epoch, we performed a linear probing on the features extracted by the two hidden layers. At initialization (marked by a star), all weights were sampled i.i.d. from a zero-mean Gaussian distribution. We averaged the results over 5 trials.

Experimental results. Across all 5 trials in both MLPs, the features from the hidden layers were linearly separable at random initialization, as evidenced by the perfect linear probing accuracy. After each epoch, the linear probe accuracy remained perfect, implying the features from the hidden layers remained linearly separable during training. Thus, *training the weights away from the random initialization did not impact the linear separability of the features*.

4.2.2 Effects of Nonlinear Activations

We now investigate how different activations affect the linear separability of the random features.



(a) $K = 2$ subspaces. (b) $K = 3$ subspaces.

Figure 8: **Linear separability of random features on synthetic UoS data.** When the input data perfectly lie on a union of $K = 2$ (left) or $K = 3$ (right) subspaces, a linear classifier achieves perfect train and test accuracy when trained on features extracted by a sufficiently wide nonlinear layer with randomly initialized weights.

Experimental setup. We created train and test sets using the above data generation process. Afterwards, we trained a linear classifier upon the random feature model. Specifically, we sampled the entries of $\boldsymbol{W} \in \mathbb{R}^{D \times d}$ iid from $\mathcal{N}(0, 10^{-2})$, then applied the random feature mapping (2) on the train and test samples. Afterwards, we trained a linear classifier $\boldsymbol{V} \in \mathbb{R}^{K \times D}$ on the train set random features under cross-entropy loss. After training, we used the trained classifier $\boldsymbol{V}$ to classify the test samples. We averaged all results over 10 trials.

In Figure 6, we considered ReLU and quadratic activations, and set $d = 128$. In Figure 6a, we set $K = 2$ and swept through r from 2^2 to 2^6 by powers of 2. In Figure 6b, we fixed $r = 16$ and swept through the number of subspaces K from 2 to 2^5, again by powers of 2. In both sweeps, we varied the network width D from 2^5 to 2^{10} by powers of 2.

In Figure 8, we set $d = 16$, $r = 4$, and considered $K = 2$ and $K = 3$. We used the following nonlinear activations: ReLU, ELU with parameter $\alpha = 1$, GELU, Leaky-ReLU with negative slope 0.01, and quadratic. We varied the network width D from 2^3 to 2^8 by powers of 2.

Experimental results. Based upon the above setup, we discuss the results below.

- **Dependence on K and r.** Figure 6 shows the width of ReLU and quadratic layers have similar dependence w.r.t. the intrinsic dimension and the number of subspaces. At all values of r and K, the linear classifier achieved perfect accuracy at similar widths for both activations. Thus, although our analysis assumes a quadratic activation, our empirical findings in Figure 6 imply similar results hold under the ReLU activation.

- **Effects of nonlinear activations.** Figure 8 shows the mean and standard deviation of the train and test accuracies at each network width for every activation function. Regardless of the activation, the linear classifier's mean accuracy across the trials increased as the network width grew, eventually achieving perfect classification performance. Furthermore, the standard deviation of the accuracies approached zero at sufficiently large widths.

  Although linear classifiers eventually achieve perfect accuracy for all activations, different activations required different widths to do so. Specifically, the quadratic requires noticeably smaller widths to achieve linear separability compared to the other activations. The quadratic is the only activation to make negative entries positive – all of the other activations either zero-out negative

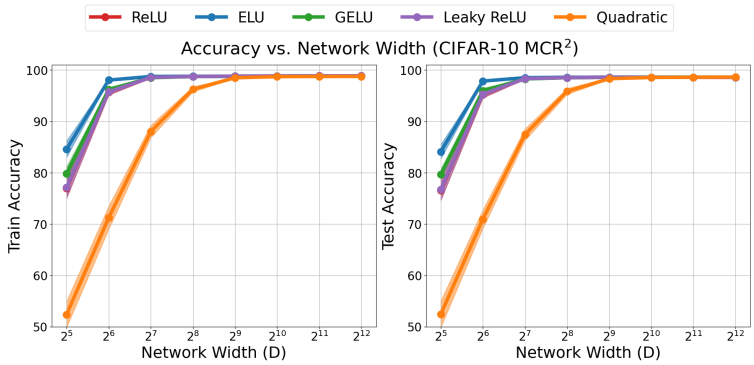


Figure 9: **Linear separability of random features on MCR^2 representations.** A linear classifier can achieve near-perfect training (left) and test (right) accuracy on MCR^2 features of CIFAR-10 data when trained on a sufficiently wide nonlinear layer with randomly initialized weights.

entries, or keep them negative. We hypothesize this property of the quadratic aids in requiring fewer random features to achieve linear separability.

4.3 CIFAR-10 Classification via MCR^2 Representations

Second, we validate our results via experiments on the CIFAR-10 image dataset [26]. While natural images do not inherently adhere to a UoS model, they can be transformed into a UoS structure through nonlinear transformations. Specifically, Maximal Coding Rate Reduction (MCR^2) [61] is employed as a framework to learn data representations, ensuring that the embeddings lie within a UoS [56].

Experimental setup. We trained a ResNet-18 model [21] to learn MCR^2 representations of the CIFAR-10 dataset [26]. We adhered to the same architectural modifications, hyperparameter settings, and training procedures as described in [61]. The resulting representations reside in the union of $K = 10$ subspaces embedded in $\mathbb{R}^d$ with $d = 128$. According to [61], for each class $k \in [K]$, the representations of images in the kth class approximately lie on a 10-dimensional subspace, implying that $r \approx 10$. Additionally, the learned representations across different classes are nearly orthogonal, meaning that $\theta_\ell \approx \pi/2$ for all $\ell \in [r]$.

After generating the MCR^2 representations, we created training and testing sets, each containing $N = 10^4$ samples, with $N_k = 10^3$ samples per class. We then sampled a random weight matrix $\boldsymbol{W} \in \mathbb{R}^{D\times d}$ with iid $\mathcal{N}(0, 10^{-2})$ entries, and applied the random feature map $f_{\boldsymbol{W}}(\boldsymbol{x})$ to the MCR^2 representations. We then trained a linear classifier $\boldsymbol{V} \in \mathbb{R}^{K\times D}$ on the random features to classify the MCR^2 representations into the K classes using cross-entropy loss. We employed the same activation functions as specified in Section 4.2. We varied the network width from 2^5 to 2^{12} in powers of 2, and averaged all results over 10 trials.

Experimental results. Figure 9 illustrates the mean and standard deviation of train and test accuracies achieved on the CIFAR-10 MCR^2 features across different activation functions and network widths. Consistently, for each activation function, the mean accuracy of the linear classifier increased with the network width, ultimately approaching *near-perfect* accuracy (approximately 99%). The standard deviation of accuracy across trials also diminished to nearly zero as the net-

work width increased. We hypothesize that the failure to achieve 100% accuracy is due to the representations not perfectly conforming to subspaces.

Interestingly, among all of the activations, the quadratic function required the largest network width to achieve near-perfect linear classification accuracy. This result contrasts with our findings in Section 4.2, where the quadratic activation achieved linear separability with the *smallest* network width. We conjecture that this discrepancy arises because the learned MCR^2 representations do not precisely lie on linear subspaces. The quadratic activation appears to be more sensitive to the noise in the MCR^2 features compared to other activation functions.

5 Discussion & Conclusion

In this work, we studied the linear separability of initial-layer features in nonlinear networks for low-dimensional data, using a UoS model motivated by the low intrinsic dimensionality of image data. We proved that a single nonlinear layer with random weights and quadratic activation can transform $K = 2$ subspaces into linearly separable sets with high probability, and extended this to the $K > 2$ case. Our result improves upon previous work by relaxing the required network width for linear separability and contributes to a complete theoretical understanding of representation learning in deep nonlinear networks. Additionally, it provides insight into the role of overparameterization and explains why random features promote good in-distribution generalization. Empirically, we found that a single nonlinear layer with random weights and various activations transforms a UoS into linearly separable sets, with the required network width for ReLU activation scaling similarly to that of the quadratic activation. In the following, we discuss the related work in Section 5.1, and conclude with future directions in Section 5.2.

5.1 Comparison with Existing Literature

We provide a more detailed discussion of the relationship between our results and prior work, complementing Section 3.1.

Separation capacity of nonlinear networks. As discussed in Section 1.1, [15, 18] analyzed two arbitrarily-structured, nonlinearly-separated classes and showed that features from random ReLU networks (two- and one-layer) are linearly separable with high probability. In the worst case, their network widths scale *exponentially* with the ambient dimension, and for classes on a union of two subspaces, this exponential scaling is in the intrinsic dimension of the subspaces. We improve upon these results by requiring only *polynomial* scaling in the intrinsic dimension. Another related work, [3], also considers two arbitrary, nonlinearly-separated sets and proves the existence of a two-layer ReLU network that achieves linear separability, while the considered network is purely *deterministic*.

Neural collapse in shallow nonlinear networks. Recently, [22] studied the Neural Collapse (NC) phenomenon in shallow ReLU networks. Specifically, they identified sufficient data-dependent conditions on when shallow ReLU networks exhibit NC. Although our work and [22] study the properties of the features in nonlinear networks, the settings have fundamental differences. Notably, NC characterizes the structure of the features from the *penultimate* layer. Additionally, [22] consider shallow ReLU networks to analyze NC in more realistic settings compared to previous works. In contrast, we study the linear separability of the features in the *initial* layers in DNNs, and study a shallow nonlinear network to facilitate such analysis.

Learning with random features. Our theoretical result uses random weights in the nonlinear layer, yielding a random feature map. Learning with random features was introduced in [41] as an alternative to kernel methods, and its generalization properties have been widely studied [42, 45, 7, 32, 14]. Although our result does not directly imply broader conclusions about learning with random features, we show that a random feature map can transform subspaces into linearly separable sets with high probability. This implies if train and test samples lie on the same subspaces, a linear classifier can perfectly classify the test samples.

Analysis of nonlinear networks with quadratic activation. Our theoretical result assumes that the nonlinear layer activation is the entry-wise quadratic function. While previous works on quadratic activation have focused on the optimization landscape and generalization abilities of overparameterized networks [31, 51, 16, 17, 47], our contribution lies in demonstrating that data on a union of subspaces can be made linearly separable with high probability under this quadratic activation. This perspective provides new insights into the feature separation properties of quadratic activation networks, complementing the optimization-centric findings of prior studies.

Rare Eclipse problem. Moreover, our problem shares conceptual similarities with the Rare Eclipse problem studied in [8, 11], which focuses on mapping two linearly separable sets into a lower-dimensional space where they become disjoint with high probability. Using Gordon's Escape through a Mesh [19], [8] demonstrated that a random Gaussian matrix achieves this and provides a lower bound on the required dimension. Similarly, we show that a nonlinear random mapping can *transform* two sets (linear subspaces) into linearly separable sets with high probability. However, beyond this shared goal of increasing separability, the two problems differ fundamentally in approach and context.

5.2 Limitations & Future Directions

This work has opened many interesting avenues for future work. First, as discussed in Section 2.1, a union of low-dimensional linear subspaces is a simplified model to capture the local linear structure in nonlinear manifolds. Relaxing the UoS data model to consider the global nonlinear structure in manifolds would be a natural extension to this work. For example, one could model a data sample $\boldsymbol{x}$ in the kth class as $\boldsymbol{x} := \phi(\boldsymbol{U}_k\boldsymbol{\alpha})$, where $\phi(\cdot)$ is from a class of nonlinear functions, and $\boldsymbol{U}_k \in \mathbb{R}^{d\times r_k}$ captures the data's low intrinsic dimensionality. Additionally, our experiments demonstrated replacing the quadratic with other activations, such as ReLU, yield similar requirements on the network width. Extending our analysis to consider the ReLU activation is another possible direction for future work. Finally, in this work, we proved there exists a hyperplane $\boldsymbol{v}$ that separates the random features with high probability. However, this is not necessarily the max-margin hyperplane. Thus, considering an optimization over $\boldsymbol{v}$ would yield tighter bounds on the required network width, i.e., reduce the polynomial degree on K and/or r.

Acknowledgements

AX, CY, PW, and QQ acknowledge support from NSF CAREER CCF-2143904, NSF IIS 2312842, NSF IIS 2402950, ONR N00014-22-1-2529, and a gift grant from KLA. The authors would also like to thank Dr. Samet Oymak (University of Michigan) and Dr. Boris Hanin (Princeton University) for their valuable insights.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *International Conference on Learning Representations*, 5, 2017.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[3] Senjian An, Farid Boussaid, and Mohammed Bennamoun. How can deep rectifier networks achieve linear separability and preserve distances? In *International Conference on Machine Learning*, pages 514–523. PMLR, 2015.

[4] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *International Conference on Learning Representations*, 6, 2018.

[6] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of machine learning research*, 18(21):1–38, 2017.

[8] Afonso S Bandeira, Dustin G Mixon, and Benjamin Recht. Compressive classification and the rare eclipse problem. In *Compressed Sensing and its Applications: Second International MATHEON Conference 2015*, pages 197–220. Springer, 2017.

[9] Harry Bateman and Arthur Erdélyi. Higher transcendental functions, volume ii. *Bateman Manuscript Project) Mc Graw-Hill Book Company*, 1953.

[10] Bradley CA Brown, Anthony L Caterini, Brendan Leigh Ross, Jesse C Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. *International Conference on Learning Representations*, 11, 2023.

[11] Valerio Cambareri, Chunlei Xu, and Laurent Jacques. The rare eclipse problem on tiles: Quantised embeddings of disjoint convex sets. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 241–245. IEEE, 2017.

[12] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspace in diffusion models for controllable image editing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[14] Zhijun Chen and Hayden Schaeffer. Conditioning of random fourier feature matrices: double descent and generalization error. *Information and Inference: A Journal of the IMA*, 13, 2024.

[15] Sjoerd Dirksen, Martin Genzel, Laurent Jacques, and Alexander Stollenwerk. The separation capacity of random neural networks. *Journal of Machine Learning Research*, 23(309):1–47, 2022.

[16] Simon Du and Jason Lee. On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, pages 1329–1338. PMLR, 2018.

[17] David Gamarnik, Eren C Kizildaug, and Ilias Zadik. Stationary points of a shallow neural network with quadratic activations and the global optimality of the gradient descent algorithm. *Mathematics of Operations Research*, 2024.

[18] Promit Ghosal, Srinath Mahankali, and Yihang Sun. Randomly initialized one-layer neural networks make data linearly separable. *arXiv preprint arXiv:2205.11716*, 2022.

[19] Yehoram Gordon. On milman's inequality and random subspaces which escape through a mesh in r n. In *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 1986–87*, pages 84–106. Springer, 1988.

[20] Hangfeng He and Weijie J Su. A law of data separation in deep learning. *Proceedings of the National Academy of Sciences*, 120(36):e2221704120, 2023.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Wanli Hong and Shuyang Ling. Beyond unconstrained features: Neural collapse for shallow neural networks with general data. *arXiv preprint arXiv:2409.01832*, 2024.

[23] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[24] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *International Conference on Learning Representations*, 7, 2019.

[25] Vignesh Kothapalli and Tom Tirer. Kernel vs. kernel: Exploring how the data structure affects neural collapse. *arXiv preprint arXiv:2406.02105*, 2024.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *International Conference on Learning Representations*, 7, 2019.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *International Conference on Learning Representations*, 6, 2018.

[30] Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Understanding and improving transfer learning of deep models via neural collapse. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[31] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.

[32] Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random fourier features. *Journal of Machine Learning Research*, 22(108):1–51, 2021.

[33] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7128–7148, 2021.

[34] Wojciech Masarczyk, Mateusz Ostaszewski, Ehsan Imani, Razvan Pascanu, Piotr Miłoś, and Tomasz Trzcinski. The tunnel effect: Building data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[36] Soo Min Kwon, Zekai Zhang, Dogyoon Song, Laura Balzano, and Qing Qu. Efficient low-dimensional compression of overparameterized models. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pages 1009–1017, May 2024.

[37] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117 (40):24652–24663, 2020.

[38] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[39] Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *International Conference on Learning Representations*, 2020.

[40] Qing Qu, Ju Sun, and John Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *Advances in Neural Information Processing Systems*, 27, 2014.

[41] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

[42] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 21, 2008.

[43] Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *International Conference on Machine Learning*, pages 28729–28745. PMLR, 2023.

[44] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

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

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

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

[48] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23): 11537–11546, 2019.

[49] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 3, 2015.

[50] Lucy Joan Slater. Generalized hypergeometric functions. (*No Title*), 1966.

[51] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

[52] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[53] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012.

[54] Ashish Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[55] Peng Wang, Xiao Li, Can Yaras, Zhihui Zhu, Laura Balzano, Wei Hu, and Qing Qu. Understanding deep representation learning via layerwise feature compression and discrimination. *arXiv preprint arXiv:2311.02960*, 2023.

[56] Peng Wang, Huikang Liu, Druv Pai, Yaodong Yu, Zhihui Zhu, Qing Qu, and Yi Ma. A global geometric analysis of maximal coding rate reduction. In *Forty-first International Conference on Machine Learning*, 2024.

[57] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

[58] Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[59] Can Yaras, Peng Wang, Wei Hu, Zhihui Zhu, Laura Balzano, and Qing Qu. The law of parsimony in gradient descent for learning deep linear networks. *arXiv preprint arXiv:2306.01154*, 2023.

[60] Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning & adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

[61] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434, 2020.

[62] Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *Journal of Machine Learning Research*, 23(67):1–28, 2022.

[63] Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. In *International Conference on Machine Learning*, pages 27179–27202. PMLR, 2022.

[64] Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34:29820–29834, 2021.

A Supporting Results

We provide supporting Lemmas that are useful in proving Theorem 1. Beforehand, we re-state previous notation here for convenience, and introduce some new notation. We use $\mathcal{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and χ^2_m to denote a chi-squared distribution with m degrees of freedom. We use $Z \,\big|\, \mathcal{A}$ to denote random variable Z conditioned on an event $\mathcal{A}$. We denote the pdf of a random variable Z with $f_Z(\cdot)$, and the covariance of a random vector with $\mathrm{Cov}(\cdot)$.

We use $\|\cdot\|$ to denote the Euclidean norm of a vector, $\sigma_i(\cdot)$ to denote the ith largest singular value of a matrix, and $\lambda_i(\cdot)$ to denote the ith largest eigenvalue of a symmetric matrix. We also use $\mathbf{0}_m$ to denote the m-dimensional vector of all zeroes.

For any positive integer N, we use $[N]$ to denote the set $\{1, 2, \ldots, N\}$. With a slight abuse of notation, for some function ϕ and set $\mathcal{X}$, $\phi(\mathcal{X})$ denotes the set $\{\boldsymbol{x} \in \mathcal{X} : \phi(\boldsymbol{x})\}$.

Let $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, $\boldsymbol{U}_1, \boldsymbol{U}_2 \in \mathbb{R}^{d\times r}$ be such that their columns are orthonormal bases for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, where the subspaces satisfy Assumption 1. Also let $\boldsymbol{x} := \boldsymbol{U}_1^\top \boldsymbol{w}$, and $\boldsymbol{y} := \boldsymbol{U}_2^\top \boldsymbol{w}$. Note $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_r, \boldsymbol{I}_r)$ and $\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}_r, \boldsymbol{I}_r)$. Also let $X := \|\boldsymbol{x}\|^2$ and $Y := \|\boldsymbol{y}\|^2$, meaning $X, Y \sim \chi^2_r$. Note $\boldsymbol{x}$ and $\boldsymbol{y}$, as well as X and Y, are *correlated*. Finally, let $\boldsymbol{a}$ and $\boldsymbol{b}$ be random vectors with the following distributions:

$$\boldsymbol{a} \sim \boldsymbol{x} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2 \;\text{ and }\; \boldsymbol{b} \sim \boldsymbol{y} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2.$$

A.1 Expectation of Order Statistics: χ^2_m Random Variables

Lemma 1 provides exact expressions for the expectation of the maximum and minimum of two iid χ^2_m random variables.

Lemma 1. *Let $X, Y \overset{iid}{\sim} \chi^2_m$, $A = \max\{X, Y\}$, and $B = \min\{X, Y\}$. Then,*

$$\mathbb{E}[A] = m + \frac{2}{\sqrt{\pi}} \frac{\Gamma((m+1)/2)}{\Gamma(m/2)}, \;\; \textit{and} \;\; \mathbb{E}[B] = m - \frac{2}{\sqrt{\pi}} \frac{\Gamma((m+1)/2)}{\Gamma(m/2)},$$

where $\Gamma(\cdot)$ denotes the Gamma function.

Proof. Note $A + B = X + Y$, so $\mathbb{E}[A + B] = \mathbb{E}[X + Y] = 2m$. Therefore, it suffices to compute $\mathbb{E}[A]$:

$$\begin{aligned}
\mathbb{E}[A] &= \int_0^\infty \int_0^\infty \max\{x, y\} f_X(x) f_Y(y)\, dx\, dy \\
&= \int_0^\infty \int_y^\infty x f_X(x) f_Y(y)\, dx\, dy + \int_0^\infty \int_x^\infty y f_X(x) f_Y(y)\, dy\, dx = 2 \int_0^\infty \int_y^\infty x f_X(x) f_Y(y)\, dx\, dy \\
&\overset{(a)}{=} \frac{2}{2^m \Gamma(m/2)^2} \int_0^\infty \int_y^\infty x^{m/2} e^{-x/2} y^{m/2-1} e^{-y/2}\, dx\, dy,
\end{aligned}$$

where we substituted the pdf of a χ^2_m distribution in (a). Letting $t = \frac{x}{2}$ results in

$$\begin{aligned}
&\frac{2}{2^m\Gamma(m/2)^2}\int_0^\infty\int_y^\infty x^{m/2}e^{-x/2}y^{m/2-1}e^{-y/2}\,dx\,dy \\
&= \frac{4}{2^{m/2}\Gamma(m/2)^2}\int_0^\infty\int_{y/2}^\infty t^{m/2}e^{-t}y^{m/2-1}e^{-y/2}\,dt\,dy \\
&\overset{(b)}{=} \frac{4}{2^{m/2}\Gamma(m/2)^2}\int_0^\infty \Gamma(m/2+1,y/2)y^{m/2-1}e^{-y/2}\,dy,
\end{aligned}$$

where in (b), we substituted the definition of the upper incomplete Gamma function, denoted as $\Gamma(p,x)$. Using the recurrence relation $\Gamma(p+1,x) = p\Gamma(p,x) + x^pe^{-x}$ yields

$$\begin{aligned}
&\frac{4}{2^{m/2}\Gamma(m/2)^2}\int_0^\infty \Gamma(m/2+1,y/2)y^{m/2-1}e^{-y/2}\,dy \\
&= \underbrace{\frac{m}{\Gamma(m/2)^2}\int_0^\infty \Gamma(m/2,y/2)(y/2)^{m/2-1}e^{-y/2}\,dy}_{(c)} + \underbrace{\frac{1}{2^{m-2}\Gamma(m/2)^2}\int_0^\infty y^{m-1}e^{-y}\,dy}_{(d)}.
\end{aligned}$$

We first simplify (c). Letting $s = y/2$, (c) becomes

$$\frac{2m}{\Gamma(m/2)^2}\int_0^\infty \Gamma(m/2,s)s^{m/2-1}e^{-s}\,ds.$$

From pg. 137, eq. (8) in [9]:

$$\int_0^\infty \Gamma(m/2,s)s^{m/2-1}e^{-s}\,ds = \frac{\Gamma(m)}{(m/2)\cdot 2^m}{}_2F_1(1,m;m/2+1;1/2)$$

where ${}_2F_1(a,b;c,d)$ denotes the ordinary hypergeometric function. By Gauss's Second Summation Theorem [50]:

$$\frac{\Gamma(m)}{(m/2)\cdot 2^m}{}_2F_1(1,m;m/2+1;1/2) = \frac{\Gamma(m)\Gamma(1/2)\Gamma(m/2+1)}{(m/2)\cdot 2^m\cdot\Gamma((m+1)/2)}. \tag{8}$$

By Legendre's duplication formula, $\Gamma(m) = \frac{\Gamma(m/2)\Gamma((m+1)/2)}{2^{1-m}\sqrt{\pi}}$. Additionally, the Gamma function satisfies the recurrence relation $\Gamma(z+1) = z\Gamma(z)$ for all $z > 0$. Substituting these expressions into (8) leads to

$$\frac{\Gamma(m)\Gamma(1/2)\Gamma(m/2+1)}{(m/2)\cdot 2^m\cdot\Gamma((m+1)/2)} = \frac{\Gamma(m/2)\Gamma(m/2+1)}{m} = \frac{\Gamma(m/2)^2}{2}.$$

Therefore, (c) fully simplifies to the following:

$$\frac{m}{\Gamma(m/2)^2}\int_0^\infty \Gamma(m/2,y/2)(y/2)^{m/2-1}e^{-y/2}\,dy = \frac{2m}{\Gamma(m/2)^2}\frac{\Gamma(m/2)^2}{2} = m.$$

We now simplify (d):

$$\frac{1}{2^{m-2}\Gamma(m/2)^2}\int_0^\infty y^{m-1}e^{-y}\,dy \overset{(e)}{=} \frac{\Gamma(m)}{2^{m-2}\Gamma(m/2)^2} \overset{(f)}{=} \frac{2\Gamma((m+1)/2)}{\Gamma(m/2)\sqrt{\pi}},$$

where (e) is by the definition of the Gamma function, and (f) is by Legendre's duplication formula. Thus,

$$\mathbb{E}[A] = m + \frac{2}{\sqrt{\pi}}\frac{\Gamma((m+1)/2)}{\Gamma(m/2)}.$$

We then use the property $\mathbb{E}[A+B] = \mathbb{E}[A] + \mathbb{E}[B] = 2m$ to obtain $\mathbb{E}[B]$:

$$\mathbb{E}[B] = m - \frac{2}{\sqrt{\pi}}\frac{\Gamma((m+1)/2)}{\Gamma(m/2)}.$$

□

A.2 Eigenvalues of Difference between Projection Matrices

Next, we provide a result about the eigenvalues of $\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top$.

Lemma 2. *Let* $\boldsymbol{U}_1, \boldsymbol{U}_2 \in \mathbb{R}^{d\times r}$ *s.t.* $\boldsymbol{U}_1^\top\boldsymbol{U}_1 = \boldsymbol{U}_2^\top\boldsymbol{U}_2 = \boldsymbol{I}_r$*, and* $\sigma_\ell(\boldsymbol{U}_1^\top\boldsymbol{U}_2) = \cos(\theta_\ell)$ *for all* $\ell \in [r]$*, where* $\theta_1 := \theta_{min} > 0$*. Then,* $\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top$ *has* r *eigenvalues equal to* $\sin(\theta_1), \sin(\theta_2), \ldots, \sin(\theta_r)$*,* r *eigenvalues equal to* $-\sin(\theta_1), -\sin(\theta_2), \ldots, -\sin(\theta_r)$*, and* $d-2r$ *eigenvalues equal to* 0*.*

Proof. Let $\Phi := \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top \in \mathbb{R}^{d\times d}$. We derive an exact expression for the characteristic polynomial $\det\Big(\Phi - \lambda\boldsymbol{I}_d\Big)$. First, note

$$\Phi = \begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1^\top \\ -\boldsymbol{U}_2^\top\end{bmatrix},$$

and let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be a singular value decomposition of $\boldsymbol{U}_1^\top\boldsymbol{U}_2 \in \mathbb{R}^{r\times r}$. Then, assuming $\lambda \neq 0$,

$$\begin{aligned}
\det\Big(\Phi - \lambda\boldsymbol{I}_d\Big) &= (-1)^d\lambda^d\det\left(\boldsymbol{I}_d - \frac{1}{\lambda}\Phi\right) = (-1)^d\lambda^d\det\left(\boldsymbol{I}_d - \frac{1}{\lambda}\begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_1^\top \\ -\boldsymbol{U}_2^\top\end{bmatrix}\right) \\
&\overset{(a)}{=} (-1)^d\lambda^d\det\left(\boldsymbol{I}_{2r} - \frac{1}{\lambda}\begin{bmatrix}\boldsymbol{I}_r & \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\ -\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top & -\boldsymbol{I}_r\end{bmatrix}\right) \\
&= (-1)^d\lambda^d\det\left(\begin{bmatrix}(1-1/\lambda)\boldsymbol{I}_r & -(1/\lambda)\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\ (1/\lambda)\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top & (1+1/\lambda)\boldsymbol{I}_r\end{bmatrix}\right) \\
&\overset{(b)}{=} (-1)^d\lambda^d(1-1/\lambda)^r\det\left((1+1/\lambda)\boldsymbol{I}_r + \frac{(1/\lambda^2)}{1-1/\lambda}\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top\right) \\
&= (-1)^d\lambda^d(1-1/\lambda)^r\det(\boldsymbol{V})\det\left((1+1/\lambda)\boldsymbol{I}_r + \frac{(1/\lambda^2)}{1-1/\lambda}\boldsymbol{\Sigma}^2\right)\det(\boldsymbol{V}^\top) \\
&= (-1)^d\lambda^d\det\left((1-1/\lambda^2)\boldsymbol{I}_r + (1/\lambda^2)\boldsymbol{\Sigma}^2\right) = (-1)^d\lambda^{d-2r}\prod_{\ell}^{r}\Big[\lambda^2 - 1 + \cos^2(\theta_\ell)\Big] \\
&= (-1)^d\lambda^{d-2r}\prod_{\ell=1}^{r}\Big[\big(\lambda + \sin(\theta_\ell)\big)\big(\lambda - \sin(\theta_\ell)\big)\Big]
\end{aligned}$$

where (a) is from Sylvester's Determinant Identity, and (b) is from the fact that

$$\det\left(\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{bmatrix}\right) = \det(\boldsymbol{A})\det(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})$$

for invertible $\boldsymbol{A}$. Solving for the roots of $\det\Big(\Phi - \lambda \boldsymbol{I}_d\Big) = 0$ yields $\lambda = \pm\sin(\theta_\ell)$ for all $\ell \in [r]$. Therefore, Φ has $2r$ eigenvalues equal to $\pm\sin(\theta_1), \pm\sin(\theta_2), \ldots, \pm\sin(\theta_r)$. Although we also have $\lambda = 0$ with multiplicity $d - 2r$, we initially assumed $\lambda \neq 0$, so these roots are invalid.

We now show the remaining $d - 2r$ eigenvalues must be 0. We showed there are at least $2r$ eigenvalues that are non-zero, so $2r \leq \operatorname{rank}(\Phi)$. Additionally, we have

$$\operatorname{rank}(\Phi) = \operatorname{rank}(\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top) \leq \operatorname{rank}(\boldsymbol{U}_1\boldsymbol{U}_1^\top) + \operatorname{rank}(-\boldsymbol{U}_2\boldsymbol{U}_2^\top) = 2r.$$

Thus, $2r \leq \operatorname{rank}(\Phi) \leq 2r$, which implies $\operatorname{rank}(\Phi) = 2r$. Therefore, Φ must have *exactly* $2r$ non-zero eigenvalues, implying the remaining $d - 2r$ eigenvalues must all be equal to 0. □

A.3 Expectation of Random Symmetric Rank-1 Matrices

We provide upper and lower bounds for $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\top]$. We first show $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\top]$ are isotropic matrices.

Lemma 3. *Let $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$, $\boldsymbol{x} := \boldsymbol{U}_1^\top \boldsymbol{w}$, $\boldsymbol{y} := \boldsymbol{U}_2^\top \boldsymbol{w}$, $\boldsymbol{a} \sim \boldsymbol{x} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$, and $\boldsymbol{b} \sim \boldsymbol{y} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$. Then, $\mathbb{E}\big[\boldsymbol{a}\boldsymbol{a}^\top\big]$ and $\mathbb{E}\big[\boldsymbol{b}\boldsymbol{b}^\top\big]$ are both isotropic matrices.*

Proof. Since $\operatorname{Cov}(\boldsymbol{x}) = \boldsymbol{I}_r$ and $\operatorname{Cov}(\boldsymbol{y}) = \boldsymbol{I}_r$, which are isotropic matrices, $\operatorname{Cov}(\boldsymbol{a})$ and $\operatorname{Cov}(\boldsymbol{b})$ are also isotropic matrices. Thus, it suffices to show $\mathbb{E}[\boldsymbol{a}] = \boldsymbol{0}_r$ and $\mathbb{E}[\boldsymbol{b}] = \boldsymbol{0}_r$.

$$\begin{aligned}\mathbb{E}[\boldsymbol{a}] &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{N}(\boldsymbol{0}_r,\boldsymbol{I}_r)}\big[\boldsymbol{x} \mid \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2\big]\\ &= \int_0^\infty\int_y^\infty \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0}_r,\boldsymbol{I}_r)}\big[\boldsymbol{x} \mid \|\boldsymbol{x}\|^2 = x\big] f_{X,Y}(x,y)\,dx\,dy \overset{(a)}{=} \boldsymbol{0}_r,\end{aligned}$$

where $X, Y \sim \chi_r^2$, and (a) is because $\boldsymbol{x} \,\big|\, \|\boldsymbol{x}\|^2 = x$ is distributed uniformly on the sphere of radius $\sqrt{x}$, so $\mathbb{E}\big[\boldsymbol{x} \mid \|\boldsymbol{x}\|^2 = x\big] = \boldsymbol{0}_r$. We can use the same argument to show $\mathbb{E}[\boldsymbol{b}] = \boldsymbol{0}_r$. Therefore, $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top] = \operatorname{Cov}(\boldsymbol{a})$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\top] = \operatorname{Cov}(\boldsymbol{b})$, which are both isotropic matrices. □

The next result provides upper and lower bounds for $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\top]$.

Lemma 4. *Let $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$, $\boldsymbol{x} := \boldsymbol{U}_1^\top \boldsymbol{w}$, $\boldsymbol{y} := \boldsymbol{U}_2^\top \boldsymbol{w}$, $\boldsymbol{a} \sim \boldsymbol{x} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$, and $\boldsymbol{b} \sim \boldsymbol{y} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$. Then, we have*

$$\left(1 + \sqrt{\frac{2}{\pi}} \cdot \frac{\sin(\theta_1)}{\sqrt{r+1}}\right)\boldsymbol{I}_r \preceq \mathbb{E}\big[\boldsymbol{a}\boldsymbol{a}^\top\big] \preceq \left(1 + \frac{1}{r} \cdot \sqrt{\sum_{\ell=1}^r \sin^2(\theta_\ell)}\right)\boldsymbol{I}_r, \quad \textit{and}$$

$$\left(1 - \frac{1}{r} \cdot \sqrt{\sum_{\ell=1}^r \sin^2(\theta_\ell)}\right)\boldsymbol{I}_r \preceq \mathbb{E}\big[\boldsymbol{b}\boldsymbol{b}^\top\big] \preceq \left(1 - \sqrt{\frac{2}{\pi}} \cdot \frac{\sin(\theta_1)}{\sqrt{r+1}}\right)\boldsymbol{I}_r.$$

Proof. By Lemma 3, $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]$ is an isotropic matrix, so it suffices to upper and lower bound $\mathrm{Tr}\big(\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]\big) = \mathbb{E}\big[\mathrm{Tr}(\boldsymbol{a}\boldsymbol{a}^\top)\big] = \mathbb{E}[\|\boldsymbol{a}\|^2]$. By definition of $\boldsymbol{a}$, $\|\boldsymbol{a}\|^2 \sim \max\{X, Y\}$, where $X, Y \sim \chi_r^2$ are not necessarily independent. We first note

$$\|\boldsymbol{a}\|^2 = \frac{1}{2}\Big(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 + \big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big).$$

Therefore,

$$\mathbb{E}[\|\boldsymbol{a}\|^2] = \frac{1}{2}\Big(\mathbb{E}[\|\boldsymbol{x}\|^2] + \mathbb{E}[\|\boldsymbol{y}\|^2] + \mathbb{E}\big[|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2|\big]\Big) = r + \frac{1}{2}\Big(\mathbb{E}\big[|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2|\big]\Big), \tag{9}$$

so it suffices to upper and lower bound $\mathbb{E}\big[|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2|\big]$. First, note

$$\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] = \mathbb{E}\Big[\big|\|\boldsymbol{U}_1^\top\boldsymbol{w}\|^2 - \|\boldsymbol{U}_2^\top\boldsymbol{w}\|^2\big|\Big] = \mathbb{E}\Big[\big|\boldsymbol{w}^\top(\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)\boldsymbol{w}\big|\Big]. \tag{10}$$

Let $\Phi := \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top$. We first establish an upper bound as such:

$$\begin{aligned}\mathbb{E}\Big[\big|\boldsymbol{w}^\top(\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)\boldsymbol{w}\big|\Big] &= \mathbb{E}\Big[\sqrt{(\boldsymbol{w}^\top\Phi\boldsymbol{w})^2}\Big] \overset{(a)}{\leq} \sqrt{\mathbb{E}\Big[(\boldsymbol{w}^\top\Phi\boldsymbol{w})^2\Big]} \\ &= \sqrt{\mathrm{Var}(\boldsymbol{w}^\top\Phi\boldsymbol{w})} \overset{(b)}{=} \sqrt{2\mathrm{Tr}\Big(\Phi^2\Big)} = 2\sqrt{\sum_{\ell=1}^r \sin^2(\theta_\ell)},\end{aligned}$$

where (a) is from Jensen's inequality, (b) is from Eq. (381) in [38], and the last equality is due to Lemma 2. Therefore,

$$\mathbb{E}[\|\boldsymbol{a}\|^2] \leq r + \sqrt{\sum_{\ell=1}^r \sin^2(\theta_\ell)} \implies \mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top] \preceq \left(1 + \frac{1}{r}\cdot\sqrt{\sum_{\ell=1}^r \sin^2(\theta_\ell)}\right)\boldsymbol{I}_r.$$

We now establish a lower bound for $\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big]$. Note Φ is symmetric, so there exists an eigendecomposition $\Phi = \boldsymbol{Q}\Lambda\boldsymbol{Q}^\top$ where $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix, and Λ is a diagonal matrix consisting of the eigenvalues of Φ. We assume the eigenvalues are listed in descending order in Λ. By Lemma 2, Φ has $2r$ non-zero eigenvalues equal to $\pm\sin(\theta_1), \pm\sin(\theta_2), \ldots, \pm\sin(\theta_r)$. Therefore:

$$\boldsymbol{w}^\top\Phi\boldsymbol{w} = \boldsymbol{w}^\top\boldsymbol{Q}\Lambda\boldsymbol{Q}^\top\boldsymbol{w} := \boldsymbol{z}^\top\Lambda\boldsymbol{z} = \sum_{\ell=1}^r \sin(\theta_\ell)\big[z_\ell^2 - z_{d-\ell+1}^2\big], \tag{11}$$

where $\boldsymbol{z} := \boldsymbol{Q}^\top\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$. Therefore,

$$\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] = \mathbb{E}\Big[|\boldsymbol{z}^\top\Lambda\boldsymbol{z}|\Big] = \mathbb{E}\Big[\Big|\sum_{\ell=1}^r \sin(\theta_\ell)\big[z_\ell^2 - z_{d-\ell+1}^2\big]\Big|\Big].$$

Before we proceed, we first note $0 < \sin(\theta_1) \leq \sin(\theta_\ell) \leq 1$ for all $\ell \in [r]$, so

$$\Big|\sin(\theta_\ell)\big[z_\ell^2 - z_{d-\ell+1}^2\big]\Big| \geq \Big|\sin(\theta_1)\big[z_\ell^2 - z_{d-\ell+1}^2\big]\Big| = \sin(\theta_1)\big|z_\ell^2 - z_{d-\ell+1}^2\big|$$

for all $\ell \in [r]$. Thus, we have

$$\sum_{\ell=1}^r\Big|\sin(\theta_\ell)\big[z_\ell^2 - z_{d-\ell+1}^2\big]\Big| \geq \sin(\theta_1)\sum_{\ell=1}^r\big|z_\ell^2 - z_{d-\ell+1}^2\big| \overset{(c)}{\geq} \sin(\theta_1)\Big|\sum_{\ell=1}^r z_\ell^2 - z_{d-\ell+1}^2\Big|,$$

where (c) is from the Triangle Inequality. Let $Z_1 := \sum_{i=1}^{r} z_i^2$ and $Z_2 := \sum_{\ell=1}^{r} z_{d-\ell+1}^2$. Note $Z_1, Z_2 \overset{\text{iid}}{\sim} \chi_r^2$ random variables, and $|Z_1 - Z_2| = \max\{Z_1, Z_2\} - \min\{Z_1, Z_2\}$. Substituting this lower bound into (10) yields

$$\begin{aligned}\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] &\geq \sin(\theta_1)\mathbb{E}\Big[|Z_1 - Z_2|\Big] = \sin(\theta_1)\Big(\mathbb{E}\big[\max\{Z_1, Z_2\}\big] - \mathbb{E}\big[\min\{Z_1, Z_2\}\big]\Big) \\ &\overset{(d)}{=} \frac{4\sin(\theta_1)}{\sqrt{\pi}} \frac{\Gamma((r+1)/2)}{\Gamma(r/2)},\end{aligned}$$

where (d) is from Lemma 1. We can then lower bound $\frac{\Gamma((r+1)/2)}{\Gamma(r/2)}$ as such. First, let $x := r/2$. Then, by Wendel's Inequality,

$$\frac{\Gamma(x+1/2)}{x^{1/2}\Gamma(x)} \geq \left(\frac{x}{x+1/2}\right)^{1/2} \iff \frac{\Gamma((r+1)/2)}{\Gamma(r/2)} \geq \frac{r}{\sqrt{2(r+1)}},$$

so

$$\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] \geq \frac{4r\sin(\theta_1)}{\sqrt{2\pi(r+1)}}.$$

Substituting this lower bound into (9) yields

$$\mathbb{E}\big[\|\boldsymbol{a}\|^2\big] \geq r + \sqrt{\frac{2}{\pi}} \cdot \frac{r\sin(\theta_1)}{\sqrt{r+1}} \implies \mathbb{E}\big[\boldsymbol{a}\boldsymbol{a}^\top\big] \succeq \left(1 + \sqrt{\frac{2}{\pi}} \cdot \frac{\sin(\theta_1)}{\sqrt{r+1}}\right)\boldsymbol{I}_r.$$

We can then use the fact $\mathbb{E}\big[\|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2\big] = 2r$ to show

$$\left(1 - \frac{1}{r} \cdot \sqrt{\sum_{\ell=1}^{r} \sin^2(\theta_\ell)}\right)\boldsymbol{I}_r \preceq \mathbb{E}\big[\boldsymbol{b}\boldsymbol{b}^\top\big] \preceq \left(1 - \sqrt{\frac{2}{\pi}} \cdot \frac{\sin(\theta_1)}{\sqrt{r+1}}\right)\boldsymbol{I}_r,$$

which completes the proof. □

A.4 Matrix Bernstein's Inequality

We use Bernstein's matrix inequality to bound the largest and smallest eigenvalues of sums of independent, random symmetric matrices.

Lemma 5 (Bernstein's inequality, adapted from Theorem 6.2 in [53])**.** *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be independent random symmetric matrices of dimension m. Assume that there exist a positive number R and matrices $\boldsymbol{A}_i$ such that*

$$\mathbb{E}[\boldsymbol{X}_i^p] \preceq \frac{p!}{2} \cdot R^{p-2} \cdot \boldsymbol{A}_i^2$$

for all $i \in [n]$ and integers $p \geq 2$. Then, for all $t \geq 0$:

$$P\left(\lambda_1\left(\sum_{i=1}^{n} \boldsymbol{X}_i - \mathbb{E}[\boldsymbol{X}_i]\right) \geq t\right) \leq m \cdot \exp\left(-\frac{t^2}{2(\sigma^2 + Rt)}\right),$$

where $\sigma^2 = \sigma_1\left(\sum_{i=1}^{n} \boldsymbol{A}_i^2\right)$. We refer to the condition $\mathbb{E}[\boldsymbol{X}_i^p] \preceq \frac{p!}{2} \cdot R^{p-2} \cdot \boldsymbol{A}_i^2$ as Bernstein's condition. *We show $\boldsymbol{a}\boldsymbol{a}^\top$ and $\boldsymbol{b}\boldsymbol{b}^\top$ satisfy Bernstein's condition.*

Lemma 6. *Let $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$ $\boldsymbol{x} := \boldsymbol{U}_1^\top \boldsymbol{w}$, $\boldsymbol{y} := \boldsymbol{U}_2^\top \boldsymbol{w}$, $\boldsymbol{a} \sim \boldsymbol{x} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$, and $\boldsymbol{b} \sim \boldsymbol{y} \,\big|\, \|\boldsymbol{x}\|^2 > \|\boldsymbol{y}\|^2$. Then, we have*

$$\mathbb{E}\big[(\boldsymbol{a}\boldsymbol{a}^\top)^p\big] \preceq \frac{p!}{2} \cdot (2r)^{p-2} \cdot 8r^2 \boldsymbol{I}_r, \;\; \textit{and} \;\; \mathbb{E}\big[(\boldsymbol{b}\boldsymbol{b}^\top)^p\big] \preceq \frac{p!}{2} \cdot (2r)^{p-2} \cdot 8r^2 \boldsymbol{I}_r$$

for all integers $p \geq 1$.

Proof. We first focus on $\mathbb{E}\big[(\boldsymbol{a}\boldsymbol{a}^\top)^p\big]$. It suffices to upper bound $\lambda_1\Big(\mathbb{E}\big[(\boldsymbol{a}\boldsymbol{a}^\top)^p\big]\Big)$:

$$\lambda_1\Big(\mathbb{E}\big[(\boldsymbol{a}\boldsymbol{a}^\top)\big]\Big) \overset{(a)}{\leq} \mathbb{E}\Big[\lambda_1\big((\boldsymbol{a}\boldsymbol{a}^\top)^p\big)\Big] \overset{(b)}{=} \mathbb{E}\big[(\|\boldsymbol{a}\|^2)^p\big],$$

where (a) is due to Jensen's inequality, and (b) is because $(\boldsymbol{a}\boldsymbol{a}^\top)^p$ is a rank-1 matrix for all integers $p \geq 1$. Recall $\|\boldsymbol{a}\|^2 \sim \max\{X, Y\}$, where $X := \|\boldsymbol{x}\|^2$ and $Y := \|\boldsymbol{y}\|^2$. Therefore,

$$\begin{aligned}
\mathbb{E}\big[(\|\boldsymbol{a}\|^2)^p\big] &= \int_0^\infty \int_0^\infty \max\{x, y\}^p f_{X,Y}(x, y)\, dx\, dy = \int_0^\infty \int_0^\infty \max\{x^p, y^p\} f_{X,Y}(x, y)\, dx\, dy \\
&= 2 \int_0^\infty \int_y^\infty x^p f_{X,Y}(x, y)\, dx\, dy \overset{(a)}{\leq} 2 \int_0^\infty \int_0^\infty x^p f_{X|Y}(x|y) f_Y(y)\, dx\, dy = 2 \int_0^\infty \mathbb{E}_{X \sim \chi_r^2}\big[X^p \mid Y\big] f_Y(y)\, dy \\
&= 2\mathbb{E}_{Y \sim \chi_r^2}\Big[\mathbb{E}_{X \sim \chi_r^2}\big[X^p \mid Y\big]\Big] = 2\mathbb{E}\big[X^p\big] \overset{(b)}{\leq} p!(2r)^p = \frac{p!}{2} \cdot (2r)^{p-2} \cdot 8r^2,
\end{aligned}$$

where (a) is because X and Y have non-negative support, and (b) is from Lemma A.6 in [40]. Therefore:

$$\mathbb{E}\big[(\boldsymbol{a}\boldsymbol{a}^\top)^p\big] \preceq \frac{p!}{2} \cdot (2r)^{p-2} \cdot 8r^2 \boldsymbol{I}_r = \frac{p!}{2} \cdot R_a^{p-2} \cdot \boldsymbol{A}^2,$$

where $R_a = 2r$ and $\boldsymbol{A}^2 = 8r^2 \boldsymbol{I}_r$. We can bound $\mathbb{E}\big[(\boldsymbol{b}\boldsymbol{b}^\top)^p\big]$ in a similar manner to obtain:

$$\mathbb{E}\big[(\boldsymbol{b}\boldsymbol{b}^\top)^p\big] \preceq \frac{p!}{2} \cdot (2r)^{p-2} \cdot 8r^2 \boldsymbol{I}_r = \frac{p!}{2} \cdot R_b^{p-2} \cdot \boldsymbol{B}^2,$$

where $R_b = 2r$ and $\boldsymbol{B}^2 = 8r^2 \boldsymbol{I}_r$. □

B Proof of Theorem 1

We now provide the full proof of Theorem 1. For ease of exposition, let $\boldsymbol{X} := \boldsymbol{W}\boldsymbol{U}_1$ and $\boldsymbol{Y} := \boldsymbol{W}\boldsymbol{U}_2$, and $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ denote the nth row in $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, written as column vectors. Note $\boldsymbol{x}_n = \boldsymbol{U}_1^\top \boldsymbol{w}_n$ and $\boldsymbol{y}_n = \boldsymbol{U}_2^\top \boldsymbol{w}_n$, where $\boldsymbol{w}_n \sim \mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$.

B.1 Conditions for Linear Separability

We first identify necessary and sufficient conditions to achieve linear separability between $f(\mathcal{S}_1)$ and $f(\mathcal{S}_2)$. By definition of linear separability, we aim to show there exists a $\boldsymbol{v} \in \mathbb{R}^D$ such that (3) holds for all $\boldsymbol{\alpha} \in$

$\mathbb{R}^r \setminus \{\mathbf{0}_r\}$. *Focusing only on* $\boldsymbol{U}_1$*, we can re-write* (3) *under Assumption* 2 *as such:*

$$\begin{aligned}\boldsymbol{v}^\top f\big(\boldsymbol{U}_1\boldsymbol{\alpha}\big) &= \sum_{n=1}^{D} v_n(\boldsymbol{w}_n^\top \boldsymbol{U}_1\boldsymbol{\alpha})^2 = \sum_{n=1}^{D} v_n(\boldsymbol{w}_n^\top \boldsymbol{U}_1\boldsymbol{\alpha})(\boldsymbol{w}_n^\top \boldsymbol{U}_1\boldsymbol{\alpha}) \\ &= \sum_{n=1}^{D} v_n(\boldsymbol{\alpha}^\top \boldsymbol{U}_1^\top \boldsymbol{w}_n)(\boldsymbol{w}_n^\top \boldsymbol{U}_1\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top\bigg(\sum_{n=1}^{D} v_n \boldsymbol{U}_1^\top \boldsymbol{w}_n\boldsymbol{w}_n^\top \boldsymbol{U}_1\bigg)\boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^\top\bigg(\sum_{n=1}^{D} v_n \boldsymbol{x}_n\boldsymbol{x}_n^\top\bigg)\boldsymbol{\alpha} > 0 \iff \sum_{n=1}^{D} v_n \boldsymbol{x}_n\boldsymbol{x}_n^\top \succ 0.\end{aligned}$$

We can re-write the $\boldsymbol{U}_2$ *part of* (3) *similarly to obtain the following necessary and sufficient conditions for linear separability:*

$$\sum_{n=1}^{D} v_n \boldsymbol{x}_n\boldsymbol{x}_n^\top \succ 0 \;\; \textit{and} \;\; \sum_{n=1}^{D} v_n \boldsymbol{y}_n\boldsymbol{y}_n^\top \prec 0. \tag{12}$$

We then construct the linear classifier $\boldsymbol{v}$ *with the following entries:*

$$\textit{For all } n \in [D],\ v_n = \operatorname{sign}\big(\|\boldsymbol{x}_n\|^2 - \|\boldsymbol{y}_n\|^2\big).$$

With this choice of $\boldsymbol{v}$*,* (12) *becomes*

$$\boldsymbol{Q}_1 := \sum_{i\in\mathcal{I}} \boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{x}_j\boldsymbol{x}_j^\top \succ 0 \;\; \textit{and} \;\; \boldsymbol{Q}_2 := \sum_{i\in\mathcal{I}} \boldsymbol{y}_i\boldsymbol{y}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{y}_j\boldsymbol{y}_j^\top \prec 0,$$

where $\mathcal{I} := \{n \in [D] : v_n = +1\}$ *and* $\mathcal{I}^c := \{n \in [D] : v_n = -1\}$*. We now upper bound the failure probability* $P\Big(\boldsymbol{Q}_1 \not\succ 0 \cup \boldsymbol{Q}_2 \not\prec 0\Big)$*.*

Dependence between v_n***,*** $\boldsymbol{x}_n$***, and*** $\boldsymbol{y}_n$***.*** *For all* $n \in [D]$*,* v_n *is statistically dependent on* $\boldsymbol{x}_n$ *and* $\boldsymbol{y}_n$*, so we cannot directly apply matrix concentration inequalities to upper bound the failure probability. However, we can construct random matrices identically distributed to* $\boldsymbol{Q}_1$ *and* $\boldsymbol{Q}_2$ *without this dependence, and apply concentration inequalities to the newly-constructed random matrices.*

First, note $v_1, v_2, \ldots, v_D$ *are iid Rademacher random variables. Also note* $\boldsymbol{x}_i, \boldsymbol{y}_j \sim \boldsymbol{a}$*, and* $\boldsymbol{x}_j, \boldsymbol{y}_i \sim \boldsymbol{b}$*, for all* $i \in \mathcal{I}$ *and* $j \in \mathcal{I}^c$*. Now, let* $Z_1, \ldots, Z_D$ *be* D *iid Rademacher random variables, and* $\boldsymbol{a}_n$ *and* $\boldsymbol{b}_n$ *be iid copies of* $\boldsymbol{a}$ *and* $\boldsymbol{b}$*, respectively, that are independent from* Z_n *for all* $n \in [D]$*. We define the following independent random matrices* $\boldsymbol{S}_{n,1}$ *and* $\boldsymbol{S}_{n,2}$ *:*

$$\boldsymbol{S}_{n,1} \sim \frac{Z_n+1}{2}\boldsymbol{a}_n\boldsymbol{a}_n^\top + \frac{Z_n-1}{2}\boldsymbol{b}_n\boldsymbol{b}_n^\top \;\; \textit{and} \;\; \boldsymbol{S}_{n,2} \sim \frac{Z_n+1}{2}\boldsymbol{b}_n\boldsymbol{b}_n^\top + \frac{Z_n-1}{2}\boldsymbol{a}_n\boldsymbol{a}_n^\top$$

for all $n \in [D]$*. We now define* $\boldsymbol{S}_1$ *and* $\boldsymbol{S}_2$ *as follows:*

$$\boldsymbol{S}_1 := \sum_{n=1}^{D} \boldsymbol{S}_{n,1} \;\; \textit{and} \;\; \boldsymbol{S}_2 := \sum_{n=1}^{D} \boldsymbol{S}_{n,2},$$

or equivalently,

$$\boldsymbol{S}_1 = \sum_{i\in\mathcal{J}} \boldsymbol{a}_i\boldsymbol{a}_i^\top - \sum_{j\in\mathcal{J}^c} \boldsymbol{b}_j\boldsymbol{b}_j^\top \;\; \textit{and} \;\; \boldsymbol{S}_2 = \sum_{i\in\mathcal{J}} \boldsymbol{b}_i\boldsymbol{b}_i^\top - \sum_{j\in\mathcal{J}^c} \boldsymbol{a}_j\boldsymbol{a}_j^\top$$

where $\mathcal{J} := \{n \in [D] : Z_n = +1\}$ *and* $\mathcal{J}^c := \{n \in [D] : Z_n = -1\}$*. By definition,* $\boldsymbol{S}_1$ *and* $\boldsymbol{S}_2$ *are identically distributed to* $\boldsymbol{Q}_1$ *and* $\boldsymbol{Q}_2$*, respectively. Upper bounding* $P\Big(\boldsymbol{Q}_1 \not\succ 0 \cup \boldsymbol{Q}_2 \not\prec 0\Big)$ *is therefore*

equivalent to upper bounding $P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big)$. *However, for all* $n \in [D]$, Z_n *has* no dependence *on* $\boldsymbol{a}_n$ *and* $\boldsymbol{b}_n$, *so we can directly apply standard concentration inequalities to upper bound* $P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big)$. *We define the random variable* $Q := \frac{1}{D}\sum_{n=1}^{D} \mathbb{1}[Z_n = +1]$, *where* $\mathbb{1}$ *denotes the indicator function.*

B.2 Bounding the Failure Probability

We aim to upper bound $P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big)$ *by some (arbitrarily) small* $\delta \in (0,1)$. *We first upper bound* $P\Big(\boldsymbol{S}_1 \not\succ 0\Big)$ *and* $P\Big(\boldsymbol{S}_2 \not\prec 0\Big)$ *individually. Let* $\gamma_1 := \sqrt{\frac{2}{\pi}} \cdot \frac{\sin(\theta_1)}{\sqrt{r+1}}$ *and* $\gamma_2 := \frac{1}{r} \cdot \sqrt{\sum_{\ell=1}^{r} \sin^2(\theta_\ell)}$. *Also let* $\alpha_1 := 1 + \gamma_1$, $\alpha_2 := 1 + \gamma_2$, $\beta_1 := 1 - \gamma_1$, *and* $\beta_2 := 1 - \gamma_2$.

We first upper bound $P\Big(\boldsymbol{S}_1 \not\succ 0\Big)$. *Note* $\boldsymbol{S}_1 \not\succ 0$ *if and only if* $\lambda_r(\boldsymbol{S}_1) \le 0$. *By Lemma 6,* $\boldsymbol{S}_1$ *and* $\boldsymbol{S}_2$ *are sums of random matrices that satisfy Bernstein's condition. Therefore, we can upper bound* $P\Big(\boldsymbol{S}_1 \not\succ 0\Big) = P\Big(\lambda_r(\boldsymbol{S}_1) \le 0\Big)$ *using Bernstein's inequality:*

$$\begin{aligned}
P\Big(\boldsymbol{S}_1 \not\succ 0\Big) &= P\Big(\lambda_r(\boldsymbol{S}_1) \le 0\Big) = P\Big(\lambda_r(\boldsymbol{S}_1) - \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big) \le -\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) \\
&\overset{(a)}{\le} P\Big(\lambda_r\big(\boldsymbol{S}_1 - \mathbb{E}[\boldsymbol{S}_1]\big) \le -\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) = P\Big(\lambda_1\big(-\boldsymbol{S}_1 - \mathbb{E}[-\boldsymbol{S}_1]\big) \ge \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) \\
&\overset{(b)}{\le} r \cdot \exp\bigg(-\frac{\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)^2}{16r^2 D + 4r\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)}\bigg),
\end{aligned} \tag{13}$$

where (a) *is due to Weyl's inequality, and* (b) *is from Lemma 5. We now upper and lower bound* $\mathbb{E}[\boldsymbol{S}_1]$ *as follows. First, using Lemma 4,*

$$(2Q - \beta_1)D\boldsymbol{I}_r \preceq \mathbb{E}\Big[\boldsymbol{S}_1 \mid Q\Big] \preceq (2Q - \beta_2)D\boldsymbol{I}_r.$$

Then, taking the expectation over Q *yields*

$$\gamma_1 D\boldsymbol{I}_r \preceq \mathbb{E}[\boldsymbol{S}_1] \preceq \gamma_2 D\boldsymbol{I}_r. \tag{14}$$

Therefore, $\gamma_1 D \le \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big) \le \gamma_2 D$. *Substituting (14) into (13) leads to*

$$P\Big(\boldsymbol{S}_1 \not\succ 0\Big) \le r \cdot \exp\bigg(-\frac{\gamma_1^2 D}{16r^2 + 4\gamma_2 r}\bigg).$$

By similar argument, we can show by $-\gamma_2 D\boldsymbol{I}_r \preceq \mathbb{E}[\boldsymbol{S}_2] \preceq -\gamma_1 D\boldsymbol{I}_r$ *that*

$$P\Big(\boldsymbol{S}_2 \not\prec 0\Big) \le r \cdot \exp\bigg(-\frac{\gamma_1^2 D}{16r^2 + 4\gamma_2 r}\bigg).$$

Finally, we apply the Union Bound to obtain the following upper bound on the failure probability:

$$P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big) \le 2r \cdot \exp\bigg(-\frac{\gamma_1^2 D}{16r^2 + 4\gamma_2 r}\bigg). \tag{15}$$

B.3 Final Result

Upper bounding (15) *by some* (*arbitrarily small*) $\delta \in (0,1)$*, and then re-arranging the terms to lower bound* D*, results in*

$$D \geq \frac{16r^2 + 4\gamma_2 r}{\gamma_1^2} \cdot \log\left(\frac{2r}{\delta}\right). \tag{16}$$

Substituting the definitions of γ_1 *and* γ_2*, as well as* $\theta_{min} := \theta_1$*, into* (16) *leads to our final result. Let* $\delta \in (0,1)$*. Then,* $\boldsymbol{S}_1 \succ 0$ *and* $\boldsymbol{S}_2 \prec 0$*, and thus* $f(\mathcal{S}_1)$ *and* $f(\mathcal{S}_2)$ *are linearly separable, if the network width* D *satisfies*

$$D \geq \frac{2\pi\left(4r^2 + \sqrt{\sum_{\ell=1}^{r} \sin^2(\theta_\ell)}\right)(r+1)}{\sin^2(\theta_{min})} \cdot \log\left(\frac{2r}{\delta}\right). \tag{17}$$