Linearly Separable Features in Shallow Nonlinear Networks: Width Scales Polynomially with Intrinsic Data Dimension

Alec S. Xu University of Michigan
Can Yaras University of Michigan
Peng Wang University of Macao
Qing Qu University of Michigan

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.

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 (He et al., 2016; Simonyan, 2015) and natural language processing (Sutskever et al., 2014; Vaswani, 2017). However, despite recent advances (Arora et al., 2018; Jacot et al., 2018; Ji and Telgarsky, 2019; Lampinen and Ganguli, 2019; Mei et al., 2018; Papyan et al., 2020; Yaras et al., 2022; Zhou et al., 2022; Zhu et al., 2021), 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 interclass separation (Alain and Bengio, 2017; Masarczyk et al., 2024; Papyan et al., 2020; Rangamani et al., 2023; Wang et al., 2025; Yaras et al., 2023). 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 — see Appendix A for a more detailed discussion.

Empirical studies on linear separability of early-layer features. Recent empirical studies investigated the role of the intermediate layers in deep nonlinear networks, e.g., Alain and Bengio (2017); Ansuini et al. (2019); He and Su (2023); Li et al. (2024); Masarczyk et al. (2024); Recanatesi et al. (2019); Wang et al. (2025); Yaras et al. (2023); Zhang et al. (2022). These studies indicate that the shallow layers expand the features such that they become linearly separable between classes. For instance, in image classification, Alain and Bengio (2017); Masarczyk et al. (2024); Wang et al. (2025) observed linear probing accuracy improves significantly across the early layers of neural networks. This implies the early layers play a critical role in achieving linear separability of the input data.

Theoretical works on linear separability of early-layer features. To our knowledge, there are

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

limited theoretical studies on the linear separability of features across nonlinear layers in DNNs. Recent works (Dirksen et al., 2022; Ghosal et al., 2022) studied the separability of features in shallow ReLU networks. These studies rigorously showed the features extracted from a two-layer (Dirksen et al., 2022) and one-layer (Ghosal et al., 2022) 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 gap between theory and practice.

Theoretical studies on 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 (Saxe et al., 2019; Wang et al., 2025). Building on the empirical observation that linear layers can emulate the behavior of deeper layers in nonlinear networks, Wang et al. (2025) 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 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 data with low intrinsic dimensions. Specifically, we show

> *a single nonlinear layer 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 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 with other activations, such as ReLU, and still achieve linear separability with similar requirements on the subspace dimensions and number of subspaces (see Figure 4).

Our findings 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 i^{th} largest eigenvalue of a symmetric matrix, and $\sigma_i(\cdot)$ to denote the i^{th} 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 results and provide a proof sketch in Section 3, with the full proof in Appendix E. In Section 4, we provide empirical evidence supporting our theoretical results, and investigate settings not considered in our analysis. Finally, we summarize our 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. (2020) used a nearest-neighbor approach to estimate the intrinsic dimension of many popular image datasets, including MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky et al., 2009), and ImageNet (Russakovsky et al., 2015). They showed the intrinsic dimension of these datasets is at most around 40,

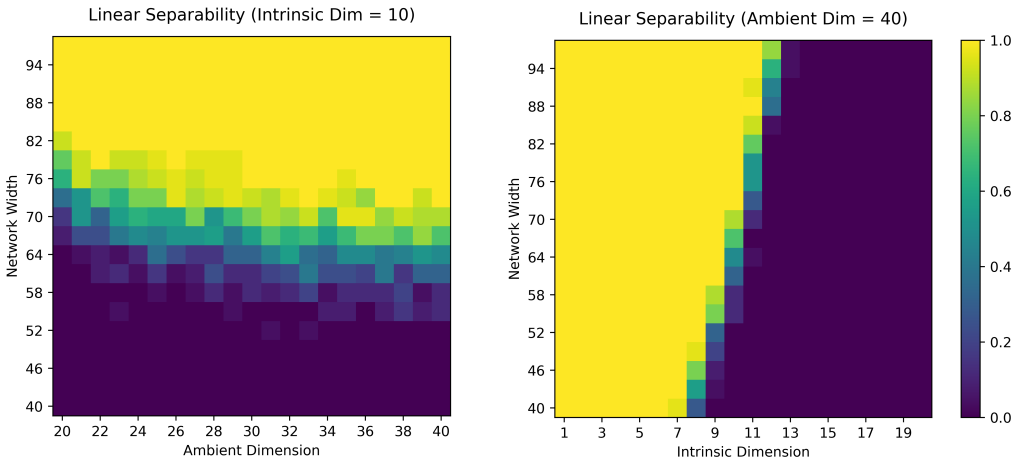


Figure 1: **Phase transition of linear separability w.r.t. dimensions (d, r) and network width D.** We demonstrate that the network width required to achieve linear separability of a union of two subspaces scales polynomially with the intrinsic dimension. See Appendix F.1 for experimental details.

even though the images themselves contain thousands of pixels. Furthermore, Brown et al. (2023) 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.

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 K low-dimensional subspaces (UoS)** that capture the local structures of manifolds. Similar models have recently been explored for understanding generative models (Chen et al., 2024; Wang et al., 2024b). Furthermore, Figure 9 in Appendix B shows common image datasets *approximately* lie on a UoS, where each subspace represents a different class.

For ease of analysis and exposition, we focus on the case where $K = 2$. Nonetheless, our results extend to the case where $K > 2$, as discussed in Section 3.2. To set the stage for our analysis, we 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 $\boldsymbol{U}_k \in \mathbb{R}^{d \times r_k}$ be an orthonormal basis of $\mathcal{S}_k$ for all $k \in [K]$. The union of subspaces $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K$ is defined as such:*

$$\bigcup_{k=1}^{K} \mathcal{S}_k := \Big\{ \boldsymbol{z} \in \mathbb{R}^d : \exists k \in [K], \boldsymbol{\alpha} \in \mathbb{R}^{r_k} \text{ s.t. } \boldsymbol{z} = \boldsymbol{U}_k \boldsymbol{\alpha} \Big\}.$$

The *principal angles* between two subspaces is a generalization of the angles between two vectors (i.e., two one-dimensional subspaces). For 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. These angles are formally defined as follows.

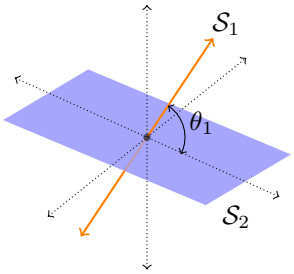


Figure 2: **The principal angle between a one-dimensional subspace $\mathcal{S}_1$ and two-dimensional subspace $\mathcal{S}_2$.**

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 $r := \min\{r_1, r_2\}$. For all $\ell \in [r]$, the ℓ^{th} principal angle $\theta_\ell \in [0, \pi/2]$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ is defined as*

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

The principal angle is illustrated in Figure 2. 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. Building on these definitions, we will make the following assumption on the UoS model for our analysis in Section 3.

Assumption 1. *There are $K = 2$ subspaces $\mathcal{S}_1, \mathcal{S}_2$ with equal dimensions, i.e., $r_1 = r_2 := r$. Fur-*

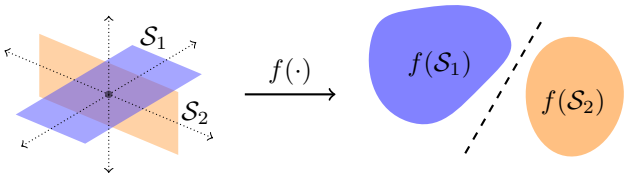


Figure 3: **An illustration of Problem 1.** We aim to find conditions on f so a union of subspaces (left) transforms into linearly separable sets (right).

thermore, the principal angles between $\mathcal{S}_1$ and $\mathcal{S}_2$ are strictly positive, i.e., $0 < \theta_1 \leq \theta_2 \leq \cdots \leq \theta_r \leq \pi/2$.

We discuss Assumption 1 below.

Number of subspaces. We assume $K = 2$ subspaces to simplify both the analysis and exposition. The results can be generalized to consider $K > 2$ subspaces, which we discuss in detail in Section 3.2.

Subspace dimensions. We assume equal dimensionality for each subspace for simplicity. In practice, each subspace in a UoS can have different dimensions. We believe our result can be generalized to this setting, and 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 non-zero points in $\mathcal{S}_1 \cap \mathcal{S}_2$. We note $\theta_1 > 0$ if and only if $r < d/2$. This assumption is typically satisfied in practice, as usually $r \ll d$.

2.2 LINEAR SEPARABILITY OF A 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 is a feature mapping from the input space $\mathbb{R}^d$ to a feature space $\mathbb{R}^D$:

$$f_{\boldsymbol{W}}(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x}). \tag{1}$$

Here, $\boldsymbol{W} \in \mathbb{R}^{D\times d}$ is the weight matrix, and $\sigma(\cdot)$ is an entry-wise nonlinear activation function. As illustrated in Figure 3, 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*

$$\boldsymbol{v}^\top f_{\boldsymbol{W}}\big(\boldsymbol{U}_1\boldsymbol{\alpha}\big) > 0 \;\; \textit{and} \;\; \boldsymbol{v}^\top f_{\boldsymbol{W}}\big(\boldsymbol{U}_2\boldsymbol{\alpha}\big) < 0 \tag{2}$$

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

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 nor a nonlinear activation alone are sufficient to transform such data into linearly separable sets — see Appendix C for a more detailed discussion. 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 (1), based upon which we characterize the sufficient conditions for achieving linear separability in Section 3.

Assumption 2. *For the mapping $f_{\boldsymbol{W}}(\boldsymbol{x})$ in (1), 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 $(i,j) \in [D] \times [d]$.*

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, e.g., Du and Lee (2018); Gamarnik et al. (2024); Li et al. (2018); Sarao Mannelli et al. (2020); Soltanolkotabi et al. (2018) — see Appendix A for a more detailed discussion. Moreover, we believe the results approximately hold for several 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 (1) are still linearly separable under a UoS data model. We also 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 4.

Random weights. Assumption 2 yields a random feature model, which has been widely studied in the literature, e.g., (Bach, 2017; Li et al., 2021; Rahimi and Recht, 2007, 2008; Rudi and Rosasco, 2017) (see (Liu et al., 2021) for a survey). Moreover, it can also shed light on trained DNNs. For example, in the infinite-width limit (Allen-Zhu et al., 2019; Arora et al., 2019; Cao and Gu, 2019; Jacot et al., 2018) random networks behave similarly to fully-trained networks. This is called the Neural Tangent Kernel (NTK) (Jacot et al., 2018) regime, where the random initialization determines the NTK, which remains constant during training (Jacot et al., 2018). Furthermore, the Neural Network Gaussian Process kernel (NNGP) (Lee et al., 2018) is the kernel associated with a network at random initialization. Recently, (Kothapalli and Tirer, 2024) studied Neural Collapse (NC) (Papyan et al.,

2020) of nonlinear networks from a kernel perspective. They showed that NNGP and NTK exhibited similar amounts of NC.

Additionally, for finite-width networks, 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 5. Thus, studying the linear separability of the features from random layers provides insight into the linear separability of the features from trained layers in finite-width networks.

3 THEORETICAL RESULTS

We first state our main theoretical results and their implications in Sections 3.1 and 3.2, and correspondingly provide a sketch of the proof in Section 3.3.

3.1 MAIN RESULTS: $K = 2$ SUBSPACES

First, we state our main theoretical result in the binary case $K = 2$.

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

$$D \geq \mathcal{O}\left(\frac{r^3}{\sin^2(\theta_{min})} \cdot \log\left(\frac{r}{\delta}\right)\right), \tag{3}$$

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

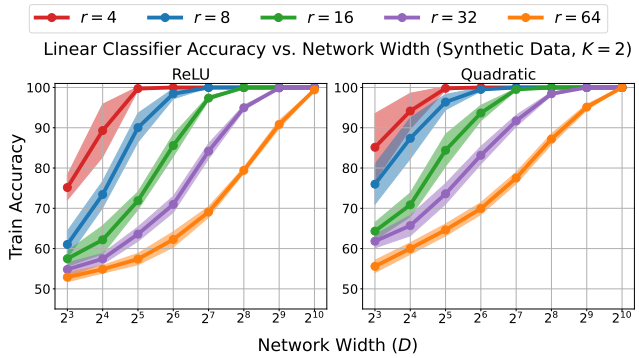
Theorem 1 states that the random feature model in (1) transforms a two subspaces into linearly separable sets, given that the network width scales *polynomially* with the subspaces' intrinsic dimension. We discuss the implications of our result below.

Network width. Our result indicates fewer neurons are needed to achieve linear separability of early-layer features compared to previous studies. Specifically, (Dirksen et al., 2022; Ghosal et al., 2022) showed one-layer and two-layer random-ReLU networks make arbitrarily structured, nonlinearly separated classes linearly separable. However, under our data model, their network widths scale *exponentially* with the intrinsic dimension. These scaling requirements are much larger than those used in practical DNNs, limiting their applicability. In contrast, Theorem 1 requires network widths to scale *polynomially* with the intrinsic dimension, aligning more closely with real-world network sizes. For example, Figure 9 in Appendix B shows the CIFAR-10 dataset (Krizhevsky et al., 2009) approximately satisfies a UoS model, where each class subspace is of rank on the order of 10^1. Previous results (Dirksen et al., 2022; Ghosal et al., 2022) require a network width on the order of $\exp(10^1)$, whereas our theorem requires a width on the order of 10^3, which more closely aligns with network sizes used in practice. Therefore, our results provide a more accurate characterization of how the early layers in practical DNNs make low-dimensional data linearly separable.

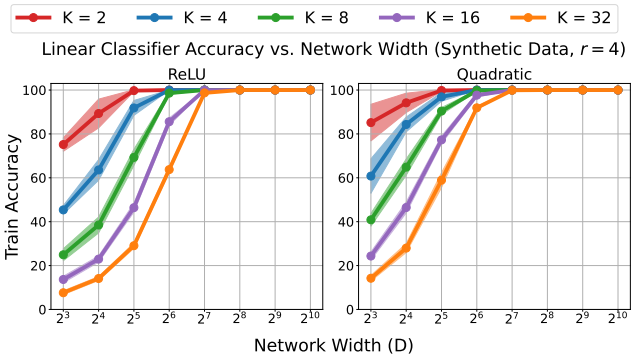
Connection to NTK-based results. Previous work (Du et al., 2019; Huang and Yau, 2020) leveraged NTK approaches to show global convergence of gradient descent in overparameterized two-layer networks. In their analyses, they showed sufficiently wide networks remain close to their random initializations during training, which is closely related to our random feature model in Assumption 2. With N training samples, Du et al. (2019); Huang and Yau (2020) showed for two-layer networks of widths $\mathcal{O}(\text{poly}(N))$, gradient descent converges to zero training loss, implying perfect linear classifier accuracy under a classification setting. In comparison, our work and Dirksen et al. (2022); Ghosal et al. (2022) consider separating *infinitely many points* in the underlying class sets, which are linear subspaces in our case. Under this setting, results from Du et al. (2019); Huang et al. (2006) require *infinitely* wide layers achieve linear separability. However, our result is *independent* of the number of data points, and only depends polynomially on the intrinsic dimension.

Overparameterization in representation learning. DNNs are often *overparameterized*, i.e., the number of parameters is larger than the number of training samples N. Theorem 1 states that with probability at least $1 - \delta$, a layer with $\mathcal{O}\left(dr^3 \cdot \log(1/\delta)\right)$ parameters transforms two subspaces (so an infinite number of data points) into linearly separable sets. In practice, one has a finite number of data points N to classify. In a finite-data setting where the data points lie on a union of two subspaces, by setting the failure probability $\delta = \frac{1}{N}$, a layer with $\mathcal{O}\left(dr^3 \cdot \log(N)\right)$ parameters correctly classifies the data points with probability at least $1 - \frac{1}{N}$. For large N, the number of parameters needed to correctly classify all N data points is much smaller than N itself. This implies an *underparameterized* network suffices in separating the data by class, and that overparameterization may serve other purposes in representation learning, e.g., feature compression (Wang et al., 2025).

In-distribution generalization. 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



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



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

Figure 4: **ReLU vs. quadratic layers for 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 Appendix F.2 for experimental details.

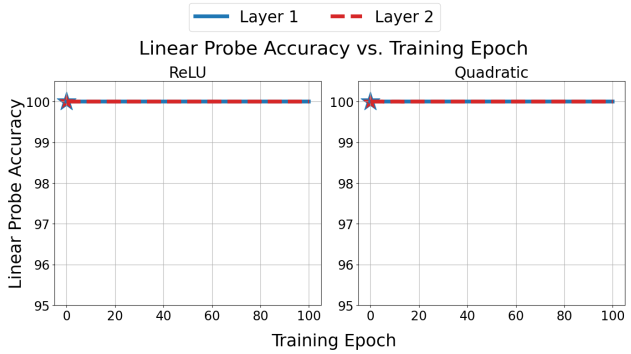


Figure 5: **Linear separability of hidden-layer features throughout training.** If the initial-layer features are linearly separable at random initialization, they remain linearly separable throughout training. See Appendix F.3 for experimental details.

two subspaces linearly separable. Suppose we have a dataset with N train samples lying on a union of two subspaces, apply the random feature map Equation (1) with $\mathcal{O}(r^3 \cdot \log(Nr))$ features, and train a linear classifier on the random features to classify the train samples. If the test samples lie in the same subspaces as the train samples, then the classifier will also achieve perfect test accuracy with probability at least $1 - \frac{1}{N}$.

3.2 EXTENSION TO $K > 2$ SUBSPACES

We now generalize the result from Theorem 1 to consider $K > 2$ subspaces.

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 $\theta_{k,1}, \theta_{k,2}, \ldots, \theta_{k,\tilde{r}}$ that satisfy Assumption 1, where $\tilde{r} := (K-1)r$. Also let Assumption 2 hold and $\delta \in (0,1)$. If the network width D satisfies*

$$D \geq \max_{k\in[K]} \mathcal{O}\left(\frac{\tilde{r}^3}{\sin^2(\theta_{k,min})} \cdot \log\left(\frac{\tilde{r}}{\delta}\right)\right) \tag{4}$$

then for all $k \in [K]$, the sets $f(\mathcal{S}_k)$ and $f\left(\bigcup_{j\in[K],j\neq k} \mathcal{S}_j\right)$ are linearly separable 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.

3.3 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, defering the full proof to Appendix E. 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 $\boldsymbol{x}_n, \boldsymbol{y}_n \in \mathbb{R}^r$ denote the n^{th} row of $\boldsymbol{X}$ 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 $\boldsymbol{w}_n \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$ denotes the n^{th} row in $\boldsymbol{W}$. First, under Assumption 2, (2) holds if and only if there exists a vector $\boldsymbol{v} \in \mathbb{R}^D$ such that

$$\sum_{n=1}^{D} v_n \boldsymbol{x}_n \boldsymbol{x}_n^\top \succ 0 \quad \text{and} \quad \sum_{n=1}^{D} v_n \boldsymbol{y}_n \boldsymbol{y}_n^\top \prec 0. \tag{5}$$

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. 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 $k \in \{1,2\}$, $\|\boldsymbol{U}_1^\top \boldsymbol{w}_n\|^2 = \|\boldsymbol{U}_2^\top \boldsymbol{w}_n\|^2$ occurs with probability zero. With this choice of $\boldsymbol{v}$, (5) is equivalent to

$$\begin{aligned} \boldsymbol{S}_1 &:= \sum_{i\in\mathcal{I}} \boldsymbol{x}_i\boldsymbol{x}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{x}_j\boldsymbol{x}_j^\top \succ 0, \text{ and} \\ \boldsymbol{S}_2 &:= \sum_{i\in\mathcal{I}} \boldsymbol{y}_i\boldsymbol{y}_i^\top - \sum_{j\in\mathcal{I}^c} \boldsymbol{y}_j\boldsymbol{y}_j^\top \prec 0, \end{aligned} \tag{6}$$

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

We now wish to upper bound the failure probability $P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big) = P\Big(\lambda_r\big(\boldsymbol{S}_1\big) \le 0 \cup \lambda_1\big(\boldsymbol{S}_2\big) \ge 0\Big)$. Next, we show $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are sums of sub-exponential random matrices, which allows us to use Bernstein's matrix inequality (Tropp, 2012, Theorem 6.2) to obtain individual bounds on $P\Big(\lambda_r(\boldsymbol{S}_1) \le 0\Big)$ and $P\Big(\lambda_1(\boldsymbol{S}_2) \ge 0\Big)$. 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 $\overline{\mathcal{S}}_k := \mathcal{R}\Big(\begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2 & \dots & \boldsymbol{U}_{k-1} & \boldsymbol{U}_{k+1} & \dots & \boldsymbol{U}_K\end{bmatrix}\Big)$ 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 iff $(K-1)r < d/2$. 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 nonlinear layer of $\mathcal{O}\left(\text{poly}(Kr)\right)$ width 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 Section 4.1, we verify our main results Theorem 1 and Corollary 1 on synthetic data, and explore settings beyond our assumptions. In Sections 4.2 and 4.3, we provide experimental results on real images, which again support our theoretical results. All experiments in this section were conducted on a single NVIDIA A40 GPU. Our code is available at `https://github.com/alecxu00/uos-linear-separability`.

4.1 SYNTHETIC DATA

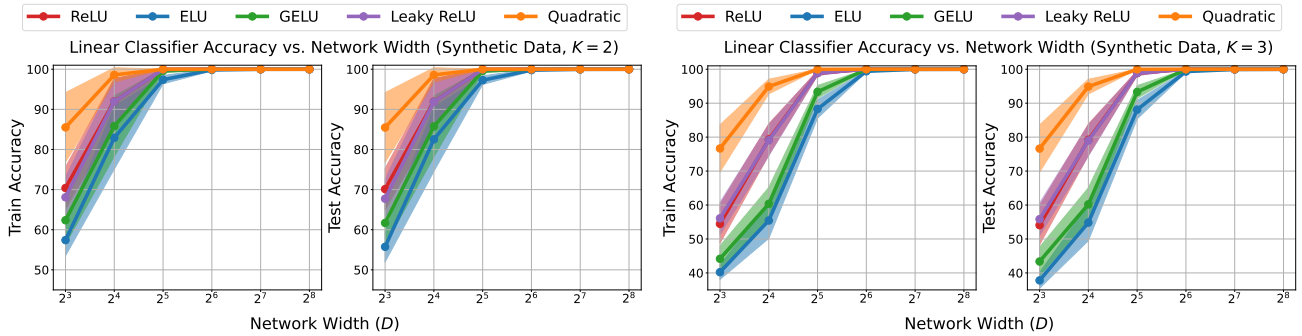
In this subsection, we verify the early-layer features are linearly separable for synthetic data generated from the UoS model under various settings, including different nonlinear activations $\sigma(\cdot)$, network widths D, and the number of subspaces K.

Synthetic data generation. We first generated K matrices $\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, \boldsymbol{U}_K$ uniformly at random from the $d \times r$ Stiefel manifold. We then generated $N = K \cdot N_k$ training samples as follows, where $N_k = 5 \cdot 10^3$ in all settings. 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 $i \in [N_k]$. Finally, when applicable, we generated N test samples using the same procedure.

Model training. We trained a linear classifier upon the random feature model in Equation (1). 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 (1) 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.

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

- **Dependence on K and r.** Figure 4 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



(a) $K = 2$ subspaces.

(b) $K = 3$ subspaces.

Figure 6: **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 random weights.

linear classifier achieved perfect accuracy at similar widths for both activations. Thus, although our analysis assumes a quadratic activation, our empirical findings in Figure 4 imply similar results hold under the ReLU activation. Further details on the experimental setup are in Appendix F.2.

- **Effects of nonlinear activations.** Figure 6 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.

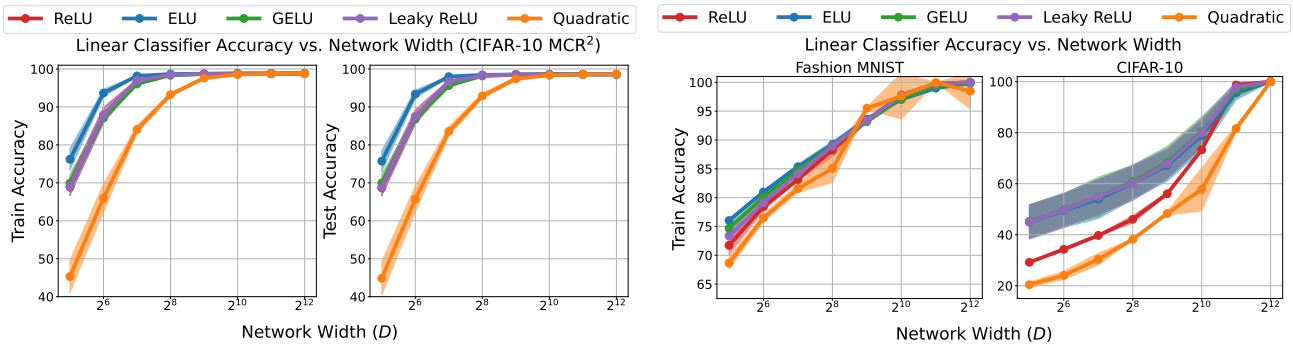
4.2 CIFAR-10 MCR^2 REPRESENTATIONS

Second, we validate our results via experiments on the Maximal Coding Rate Reduction (MCR^2) representations (Yu et al., 2020) of the CIFAR-10 image dataset (Krizhevsky et al., 2009). While natural images do not inherently adhere to a UoS model, they can be transformed into a UoS structure through nonlinear transformations. Specifically, MCR^2 (Yu et al., 2020) is a framework to learn data representations whose embeddings lie on a UoS (Wang et al., 2024a).

Setup. We trained a ResNet-18 model (He et al., 2016) to learn MCR^2 representations of the CIFAR-10 dataset (Krizhevsky et al., 2009). We adhered to the same architectural changes, hyperparameter settings, and training procedures as described in Yu et al. (2020). The resulting representations reside in a union of $K = 10$ subspaces embedded in $\mathbb{R}^d$ with $d = 128$. From Yu et al. (2020), for each class $k \in [K]$, the representations of images in the k^{th} class approximately lie on a 10-dimensional subspace, so $r \approx 10$. Additionally, the learned representations across different classes are nearly orthogonal, so $\theta_\ell \approx \pi/2$ for all $\ell \in [r]$.

We created training and testing sets with the MCR^2 representations, where each set contained $N = 10^4$ 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 $\mathcal{N}(0, 1)$ 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 using cross-entropy loss. We employed the same activation functions as specified in Section 4.1. We varied the network width from 2^5 to 2^{12} in powers of 2, and averaged all results over 10 trials.

Results. Figure 7a 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. 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 network width increased. We hypothesize that the failure to achieve 100% accuracy is due to the representations not perfectly conforming to subspaces.



(a) CIFAR-10 MCR^2 representations.

(b) Fashion MNIST (left) and CIFAR-10 (right) images

Figure 7: **Linear separability of random features on image data.** *Left:* after transforming image data to lie on a UoS using the MCR^2 (Yu et al., 2020) framework, sufficiently many random features are linearly separable. *Right:* sufficiently many random features using *raw* image data as input are also linearly separable.

4.3 FASHION MNIST AND CIFAR-10

Finally, we show that our theory approximately holds on the Fashion MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky et al., 2009) datasets. Both datasets contain $K = 10$ classes. Figure 9 in Appendix B shows both datasets approximately satisfy the UoS data model: in both datasets, a small number of singular values account for a large majority each class data matrix's Frobenius norm. In particular, both datasets' per-class intrinsic subspace dimensions are about on the order of 10^1, even though their ambient dimensions on the order of 10^2 (Fashion MNIST) or 10^3 (CIFAR-10). See Appendix B for more details.

Setup. For both datasets, we randomly sampled $N = 10^4$ training images, with $N_k = 10^3$ images per class. Then, we flattened the images into d-dimensional vectors, where $d = 784$ for Fashion MNIST, and $d = 3072$ for CIFAR-10. Finally, we followed the exact training procedure as described in Section 4.2, but averaged all results over 5 trials instead.

Results. Figure 7b shows the mean and standard deviation of the train accuracies achieved on the Fashion MNIST and CIFAR-10 random features across various activation functions. Recall from Appendix B, the per-class subspace dimensions are on the order of 10^1 in both datasets. Figure 7b shows the linear classifier achieves near-perfect accuracy at widths around 2^{11} or 2^{12} across all activations, which is on the order of 10^3. Thus, network widths that are polynomial in the *intrinsic* data dimension suffices for linear separability, which aligns with our theoretical results.

Furthermore, recall that CIFAR-10's ambient dimension is about $4\times$ that of Fashion MNIST ($d = 3072$ for CIFAR-10, while $d = 784$ for Fashion MNIST). Despite this difference, between 2^{11} and 2^{12} random features suffice for linear separability in *both* datasets, implying little to no dependence on the data ambient dimension.

5 CONCLUSION

In this work, we studied the linear separability of early-layer features in nonlinear networks for low-dimensional data, using a UoS model motivated by the low intrinsic dimensionality of image data. We rigorously proved that a single nonlinear layer with random weights and quadratic activation can transform $K \geq 2$ subspaces into (one-vs.-all) linearly separable sets with high probability. Notably, our result requires the network width to be polynomial in the intrinsic dimension, while previous results require exponential dependence. Although our analysis assumes a quadratic activation, our empirical findings on synthetic and real data indicate similar results hold for other activations, such as ReLU.

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

Alain, G. and Bengio, Y. (2017). Understanding intermediate layers using linear classifier probes. *International Conference on Learning Representations*, 5.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR.

An, S., Boussaid, F., and Bennamoun, M. (2015). How can deep rectifier networks achieve linear separability and preserve distances? In *International Conference on Machine Learning*, pages 514–523. PMLR.

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

Arora, S., Cohen, N., Golowich, N., and Hu, W. (2018). A convergence analysis of gradient descent for deep linear neural networks. *International Conference on Learning Representations*, 6.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. (2019). On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32.

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

Bandeira, A. S., Mixon, D. G., and Recht, B. (2017). Compressive classification and the rare eclipse problem. In *Compressed Sensing and its Applications: Second International MATHEON Conference 2015*, pages 197–220. Springer.

Bateman, H. and Erdélyi, A. (1953). Higher transcendental functions, volume ii. *Bateman Manuscript Project) Mc Graw-Hill Book Company*.

Brown, B. C., Caterini, A. L., Ross, B. L., Cresswell, J. C., and Loaiza-Ganem, G. (2023). Verifying the union of manifolds hypothesis for image data. *International Conference on Learning Representations*, 11.

Cambareri, V., Xu, C., and Jacques, L. (2017). 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.

Cao, Y. and Gu, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Chen, S., Zhang, H., Guo, M., Lu, Y., Wang, P., and Qu, Q. (2024). Exploring low-dimensional subspace in diffusion models for controllable image editing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chen, Z. and Schaeffer, H. (2024). Conditioning of random fourier feature matrices: double descent and generalization error. *Information and Inference: A Journal of the IMA*, 13.

Dirksen, S., Genzel, M., Jacques, L., and Stollenwerk, A. (2022). The separation capacity of random neural networks. *Journal of Machine Learning Research*, 23(309):1–47.

Du, S. and Lee, J. (2018). On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, pages 1329–1338. PMLR.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*.

Gamarnik, D., Kizildaug, E. C., and Zadik, I. (2024). Stationary points of a shallow neural network with quadratic activations and the global optimality of the gradient descent algorithm. *Mathematics of Operations Research*.

Ghosal, P., Mahankali, S., and Sun, Y. (2022). Randomly initialized one-layer neural networks make data linearly separable. *arXiv preprint arXiv:2205.11716*.

Glasgow, M. (2024). Sgd finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the xor problem. In *The Twelfth International Conference on Learning Representations*.

Gordon, Y. (1988). 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.

He, H. and Su, W. J. (2023). A law of data separation in deep learning. *Proceedings of the National Academy of Sciences*, 120(36):e2221704120.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hong, W. and Ling, S. (2024). Beyond unconstrained features: Neural collapse for shallow neural networks with general data. *arXiv preprint arXiv:2409.01832*.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501.

Huang, J. and Yau, H.-T. (2020). Dynamics of deep neural networks and neural tangent hierarchy. In *International conference on machine learning*, pages 4542–4551. PMLR.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31.

Ji, Z. and Telgarsky, M. (2019). Gradient descent aligns the layers of deep linear networks. *International Conference on Learning Representations*, 7.

Kothapalli, V. and Tirer, T. (2024). Kernel vs. kernel: Exploring how the data structure affects neural collapse. *arXiv preprint arXiv:2406.02105*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lampinen, A. K. and Ganguli, S. (2019). An analytic theory of generalization dynamics and transfer learning in deep linear networks. *International Conference on Learning Representations*, 7.

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

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep neural networks as gaussian processes. *International Conference on Learning Representations*, 6.

Li, X., Liu, S., Zhou, J., Lu, X., Fernandez-Granda, C., Zhu, Z., and Qu, Q. (2024). Understanding and improving transfer learning of deep models via neural collapse. *Transactions on Machine Learning Research*.

Li, Y., Ma, T., and Zhang, H. (2018). Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR.

Li, Z., Ton, J.-F., Oglic, D., and Sejdinovic, D. (2021). Towards a unified analysis of random fourier features. *Journal of Machine Learning Research*, 22(108):1–51.

Liu, F., Huang, X., Chen, Y., and Suykens, J. A. (2021). Random features for kernel approximation: A survey on algorithms, theory, and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7128–7148.

Masarczyk, W., Ostaszewski, M., Imani, E., Pascanu, R., Miłoś, P., and Trzcinski, T. (2024). The tunnel effect: Building data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 36.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671.

Meng, X., Zou, D., and Cao, Y. (2024). Benign overfitting in two-layer relu convolutional neural networks for xor data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 35404–35469.

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

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., and Goldstein, T. (2020). The intrinsic dimension of images and its impact on learning. *International Conference on Learning Representations*.

Qu, Q., Sun, J., and Wright, J. (2014). Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *Advances in Neural Information Processing Systems*, 27.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20.

Rahimi, A. and Recht, B. (2008). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 21.

Rangamani, A., Lindegaard, M., Galanti, T., and Poggio, T. A. (2023). Feature learning in deep classifiers through intermediate neural collapse. In *International Conference on Machine Learning*, pages 28729–28745. PMLR.

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

Rudi, A. and Rosasco, L. (2017). Generalization properties of learning with random features. *Advances in Neural Information Processing Systems*, 30.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252.

Sarao Mannelli, S., Vanden-Eijnden, E., and Zdeborová, L. (2020). Optimization and generalization of shallow neural networks with quadratic activation

functions. *Advances in Neural Information Processing Systems*, 33:13445–13455.

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

Simonyan, K. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 3.

Slater, L. J. (1966). Generalized hypergeometric functions. *(No Title)*.

Soltanolkotabi, M., Javanmard, A., and Lee, J. D. (2018). Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27.

Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, P., Li, X., Yaras, C., Zhu, Z., Balzano, L., Hu, W., and Qu, Q. (2025). Understanding deep representation learning via layerwise feature compression and discrimination. *Journal of Machine Learning Research*, 26(220):1–61.

Wang, P., Liu, H., Pai, D., Yu, Y., Zhu, Z., Qu, Q., and Ma, Y. (2024a). A global geometric analysis of maximal coding rate reduction. In *Forty-first International Conference on Machine Learning*.

Wang, P., Zhang, H., Zhang, Z., Chen, S., Ma, Y., and Qu, Q. (2024b). Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*.

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

Yaras, C., Wang, P., Hu, W., Zhu, Z., Balzano, L., and Qu, Q. (2023). The law of parsimony in gradient descent for learning deep linear networks. *arXiv preprint arXiv:2306.01154*.

Yaras, C., Wang, P., Zhu, Z., Balzano, L., and Qu, Q. (2022). Neural collapse with normalized features: A geometric analysis over the riemannian manifold. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Yu, Y., Chan, K. H. R., You, C., Song, C., and Ma, Y. (2020). Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434.

Zhang, C., Bengio, S., and Singer, Y. (2022). Are all layers created equal? *Journal of Machine Learning Research*, 23(67):1–28.

Zhou, J., Li, X., Ding, T., You, C., Qu, Q., and Zhu, Z. (2022). 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.

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]

SUPPLEMENTARY MATERIALS

Notation. 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 i^{th} largest singular value of a matrix, and $\lambda_i(\cdot)$ to denote the i^{th} 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 $\{\boldsymbol{x} \in \mathcal{X} : \phi(\boldsymbol{x})\}$.

A RELATED WORKS

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

Separation capacity of nonlinear networks. As discussed in Section 1.1, Dirksen et al. (2022); Ghosal et al. (2022) are most closely related to ours. They analyzed two arbitrarily-structured, nonlinearly-separated classes, and showed the resulting features from two-layer (Dirksen et al., 2022) and one-layer (Ghosal et al., 2022) random ReLU networks are linearly separable with high probability. In our work, we specifically model the inputs as lying on a union of low-dimensional subspaces, and consider the quadratic activation instead of ReLU. Under our data model, the results in Dirksen et al. (2022); Ghosal et al. (2022) require the network widths to scale *exponentially* with the intrinsic dimension of the input data. In contrast, our result requires *polynomial* dependence. Another related work, An et al. (2015), also assumes the data is from two arbitrary nonlinearly-separated sets. They prove there exists a *deterministic* two-layer ReLU network that makes these sets linearly separable.

XOR data. Previous works on neural network analyses have considered XOR input data (Glasgow, 2024; Meng et al., 2024), which is a special case of our UoS data model. To visualize this, consider an arbitrary data point $\boldsymbol{x} \in \{-1, +1\}^2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$, where $x_i \in \{-1, +1\}$. Then, the corresponding label is $y = \mathrm{XOR}(x_1, x_2) = \begin{cases} +1 & x_1 = x_2 \\ -1 & x_1 \neq x_2 \end{cases}$. Figure 8 shows these data points lie on a union of two one-dimensional linear subspaces, where each linear subspace represents a different class.

Neural collapse in shallow nonlinear networks. Recently, Hong and Ling (2024) 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 Hong and Ling (2024) 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, Hong and Ling (2024) 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 *early* layers in DNNs, and study a shallow nonlinear network to facilitate such analysis.

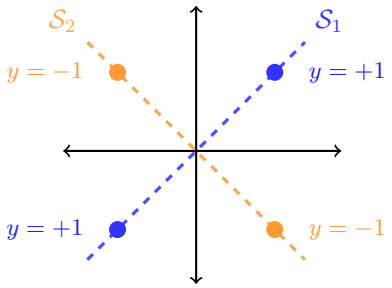


Figure 8: Two-dimensional XOR data lies on a union of one-dimensional subspaces.

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 (Rahimi and Recht, 2007) as an alternative to kernel methods, and its generalization properties have been widely studied (Bach, 2017; Chen and Schaeffer, 2024; Li et al., 2021; Rahimi and Recht, 2008; Rudi and Rosasco, 2017). 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.

Neural tangent kernel. As discussed in Section 3.1, previous works have shown the global convergence of gradient descent on overparameterized two-layer networks using NTK techniques (Du et al., 2019; Huang and Yau, 2020). These works showed that during training, a highly overparameterized two-layer network remains close to its random initialization, which is closely related to our random feature model. These results showed gradient descent converges to zero training loss for wide two-layer networks. In contrast, our work does not consider any training. Rather, we directly assume a random feature model, showing a nonlinear layer with random weights makes two linear subspaces linearly separable with high probability.

Analysis of quadratic-activation networks. Our theoretical result assumes the nonlinear 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 (Du and Lee, 2018; Gamarnik et al., 2024; Li et al., 2018; Sarao Mannelli et al., 2020; Soltanolkotabi et al., 2018), 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. Finally, our problem shares conceptual similarities with the Rare Eclipse problem studied in Bandeira et al. (2017); Cambareri et al. (2017), 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 (Gordon, 1988), Bandeira et al. (2017) 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.

B UNION OF SUBSPACES IN REAL IMAGE DATASETS

In this section, we provide empirical evidence showing common image datasets approximately lie on a union of low-dimensional subspaces, where each subspace represents a different class in the dataset.

Setup. We considered the MNIST, Fashion MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009) datasets. In each dataset, we first flattened the images so that they are d-dimensional vectors, where d is the number channels multiplied by the pixels. For MNIST and Fashion MNIST, $d = 1 \times 28 \times 28 = 784$, while for CIFAR-10, $d = 3 \times 32 \times 32 = 3072$. Then, for each class, we create a $d \times N$ data matrix, where $N = 5000$ is the number of data points in each class. Finally, for each class data matrix, we compute the number of singular values that account for 95% and 99% of the Frobenius norm in each class's data matrix.

Results. Figure 9 shows the proportion of singular values needed to capture 95% (darker bars) and 99% (lighter bars) of each class data matrix's Frobenius norms. For MNIST, Fashion MNIST, and CIFAR-10 respectively, about the first $15-20\%$ (about $115-155$ out of 784), $5-10\%$ (about $40-80$ out of 784), and $2-4\%$ (about $60-120$ out of 3072) of the singular values account for 99% of most class data matrix's Frobenius norms — the lone outlier is class 5 in Fashion MNIST. This implies each class *approximately* lies on its own low-dimensional subspace. In other words, **these datasets approximately satisfy the UoS data model.**

C LINEAR SEPARABILITY OF A UNION OF SUBSPACES

In this section, we discuss why a nonlinear layer $f_{\boldsymbol{W}}(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x})$ is necessary to transform a UoS into linearly separable sets. We first show two subspaces themselves are not linearly separable. Next, we show applying either a linear transformation or a nonlinear activation *individually* are insufficient in making a UoS linearly separable.

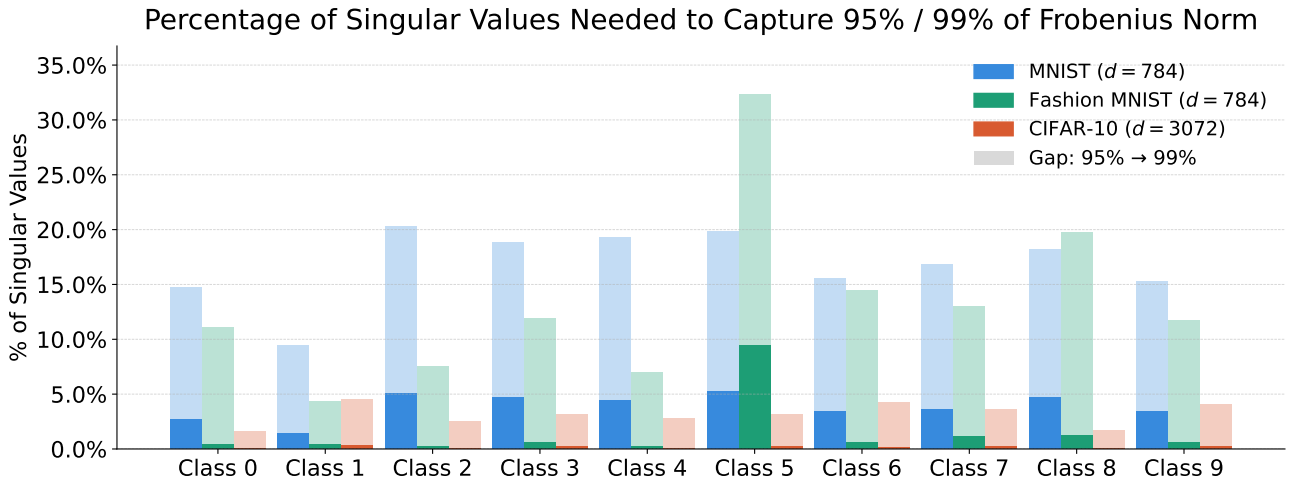


Figure 9: **In common image datasets, a small proportion of singular values in each class's data matrix capture** 95% / 99% **of the Frobenius norm.** The darker bars indicate the proportion of singular values needed to reach 95% of that class data matrix's Frobenius norm, while the lighter bars indicate the proportion needed to reach 99%.

Subspaces are not linearly separable in general. 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 10, there does not exist any hyperplane (line) that can linearly separate $\mathcal{S}_1$ and $\mathcal{S}_2$ because they both pass through the origin.

Linear mapping alone is insufficient for linear separability. Now suppose $\mathcal{S}_1$ and $\mathcal{S}_2$ are arbitrary r_1 and r_2-dimensional subspaces of $\mathbb{R}^d$, and define $g_{\boldsymbol{W}}(\boldsymbol{x}) := \boldsymbol{W}\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{W} \in \mathbb{R}^{D \times d}$. We 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 arbitrary $\boldsymbol{\alpha}^{(k)} \in \mathbb{R}^{r_k}$, we have

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

where $\tilde{\boldsymbol{U}}_k \in \mathbb{R}^{D \times r_k}$. Therefore, the point $\boldsymbol{W}\boldsymbol{U}_k\boldsymbol{\alpha}^{(k)}$ lies in an r_k-dimensional subspace in $\mathbb{R}^D$. Since this holds for all $\boldsymbol{\alpha}^{(k)} \in \mathbb{R}^{r_k}$, the sets $g_{\boldsymbol{W}}(\mathcal{S}_1)$ and $g_{\boldsymbol{W}}(\mathcal{S}_2)$ remain as linear subspaces of $\mathbb{R}^D$ that pass through the origin, which are 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 the bases of $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^2$ are $\boldsymbol{u}_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ and $\boldsymbol{u}_2 = \begin{bmatrix} -1 & 1 \end{bmatrix}^\top$, respectively. Let us first consider the entry-wise quadratic activation, which we considered in our theoretical analysis. For any $\alpha \in \mathbb{R}$, we have

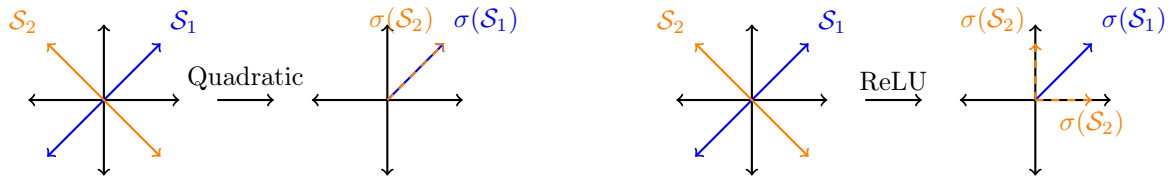
$$\sigma(\boldsymbol{u}_1\alpha) = \begin{bmatrix} 1^2 \\ 1^2 \end{bmatrix} \alpha^2 = \begin{bmatrix} \alpha^2 \\ \alpha^2 \end{bmatrix} \text{ and } \sigma(\boldsymbol{u}_2\alpha) = \begin{bmatrix} (-1)^2 \\ 1^2 \end{bmatrix} \alpha^2 = \begin{bmatrix} \alpha^2 \\ \alpha^2 \end{bmatrix},$$

so $\sigma(\boldsymbol{u}_1\alpha) = \sigma(\boldsymbol{u}_2\alpha)$. Therefore, the sets $\sigma(\mathcal{S}_1)$ and $\sigma(\mathcal{S}_2)$ are *identical* (see Figure 10, left), clearly implying they are not distinguishable.

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, $\sigma(\mathcal{S}_1) = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 > 0, x_2 > 0\right\} \cup \{\boldsymbol{0}_2\}$, while $\sigma(\mathcal{S}_2) = \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 > 0, x_2 = 0\right\} \cup \left\{\boldsymbol{x} \in \mathbb{R}^2 : x_1 = 0, x_2 > 0\right\}$, 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 10, right), since some points in $\sigma(\mathcal{S}_2)$ are above $\sigma(\mathcal{S}_1)$, and some points are below.

D AUXILIARY RESULTS

We provide auxiliary lemmas that are useful in proving Theorem 1. Let $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$, $\boldsymbol{U}_1, \boldsymbol{U}_2 \in \mathbb{R}^{d \times r}$ respectively be orthonormal bases for subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ that satisfy Assumption 1. Also let $\boldsymbol{x} := \boldsymbol{U}_1^\top\boldsymbol{w}$, and



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

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

Figure 10: **Activation alone is insufficient for linearly separating two subspaces.** When $\mathcal{S}_1 = \text{span}(\boldsymbol{u}_1)$ and $\mathcal{S}_2 = \text{span}(\boldsymbol{u}_2)$, the sets $\sigma(\mathcal{S}_1)$ and $\sigma(\mathcal{S}_2)$ are not linearly separable for $\sigma(\cdot) =$ quadratic (left) and $\sigma(\cdot) = \text{ReLU}(\cdot)$ (right).

$\boldsymbol{y} := \boldsymbol{U}_2^\top \boldsymbol{w}$. Note $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}_r, \boldsymbol{I}_r)$ and $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}_r, \boldsymbol{I}_r)$ are *correlated*. Finally, let $\boldsymbol{a}$ and $\boldsymbol{b}$ be random vectors with the following distributions:

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

D.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 derive $\mathbb{E}[A]$:

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

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 $\Gamma(p+1,x) = p\Gamma(p,x) + x^p e^{-x}$ yields

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

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 (Bateman and Erdélyi, 1953, Pg. 137, Eq. (8)):

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

where ${}_2F_1(a,b;c,d)$ denotes the ordinary hypergeometric function. By Gauss's Second Summation Theorem (Slater, 1966):

$$\frac{\Gamma(m)}{(m/2)\cdot 2^m}\,{}_2F_1(1,m;m/2+1;1/2) = \frac{\Gamma(m)\Gamma(1/2)\Gamma(m/2+1)}{(m/2)\cdot 2^m\cdot\Gamma((m+1)/2)}. \tag{7}$$

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 (7) 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 $\mathbb{E}[B]$:

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

□

D.2 EIGENVALUES OF DIFFERENCE OF 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 $\sin(\theta_1), \sin(\theta_2), \dots, \sin(\theta_r)$, r eigenvalues equal to $-\sin(\theta_1), -\sin(\theta_2), \dots, -\sin(\theta_r)$, 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 $\lambda \neq 0$,

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

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 $\pm\sin(\theta_1), \pm\sin(\theta_2), \dots, \pm\sin(\theta_r)$. 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. □

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

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

where $X, Y \sim \chi^2_r$, 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, $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top] = \mathrm{Cov}(\boldsymbol{a})$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\top] = \mathrm{Cov}(\boldsymbol{b})$, which are both isotropic matrices. □

The next result provides upper and lower bounds on $\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^\top]$ 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 $X, Y \sim \chi^2_r$ 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,

$$\mathbb{E}[\|\boldsymbol{a}\|^2] = \frac{1}{2}\Big(\mathbb{E}[\|\boldsymbol{x}\|^2] + \mathbb{E}[\|\boldsymbol{y}\|^2] + \mathbb{E}\big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\big]\Big) = r + \frac{1}{2}\Big(\mathbb{E}\big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\big]\Big), \tag{8}$$

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

$$\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] = \mathbb{E}\Big[\big|\|\boldsymbol{U}_1^\top\boldsymbol{w}\|^2 - \|\boldsymbol{U}_2^\top\boldsymbol{w}\|^2\big|\Big] = \mathbb{E}\Big[\big|\boldsymbol{w}^\top(\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top)\boldsymbol{w}\big|\Big]. \tag{9}$$

Let $\Phi := \boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top$. We 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 (Petersen et al., 2008, Eq. 381), 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:

$$\boldsymbol{w}^\top\Phi\boldsymbol{w} = \boldsymbol{w}^\top\boldsymbol{Q}\Lambda\boldsymbol{Q}^\top\boldsymbol{w} := \boldsymbol{z}^\top\Lambda\boldsymbol{z} = \sum_{\ell=1}^r \sin(\theta_\ell)\big(z_\ell^2 - z_{d-\ell+1}^2\big), \tag{10}$$

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

$$\mathbb{E}\Big[\big|\|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2\big|\Big] = \mathbb{E}\Big[|\boldsymbol{z}^\top \Lambda \boldsymbol{z}|\Big] = \mathbb{E}\bigg[\Big|\sum_{\ell=1}^{r} \sin(\theta_\ell)\big(z_\ell^2 - z_{d-\ell+1}^2\big)\Big|\bigg] \geq \sin(\theta_1)\mathbb{E}\left[\Big|\sum_{\ell=1}^{r}\big(z_\ell^2 - z_{d-\ell+1}^2\big)\Big|\right].$$

Let $Z_1 := \sum_{i=1}^{r} z_i^2$ and $Z_2 := \sum_{\ell=1}^{r} z_{d-\ell+1}^2$. Note $Z_1, Z_2 \overset{\text{iid}}{\sim} \chi_r^2$. Substituting this lower bound into (9) yields

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

where (c) is from the fact that $|Z_1 - Z_2| = \max\{Z_1, Z_2\} - \min\{Z_1, Z_2\}$ and 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 (8) 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

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

□

D.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 Tropp (2012))**.** *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 $t \geq 0$:

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

where $\sigma^2 = \sigma_1\Big(\sum_{i=1}^{n} \boldsymbol{A}_i^2\Big)$.

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. Our next result states $\boldsymbol{a}\boldsymbol{a}^\top$ and $\boldsymbol{b}\boldsymbol{b}^\top$ satisfy Bernstein's condition.

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

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

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,

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

where (a) is because X and Y have non-negative support, and (b) is from Lemma A.6 in (Qu et al., 2014). 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 $\boldsymbol{B}^2 = 8r^2\boldsymbol{I}_r$. □

E PROOF OF THEOREM 1

We now provide the full proof of Theorem 1. 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 n^{th} 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 $\boldsymbol{w}_n \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}_d, \boldsymbol{I}_d)$.

E.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 (2) holds for all $\boldsymbol{\alpha} \in \mathbb{R}^r \setminus \{\boldsymbol{0}_r\}$. Focusing only on $\boldsymbol{U}_1$, we can re-write (2) 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 (2) similarly to obtain the following necessary and sufficient conditions for linear separability:

$$\sum_{n=1}^D v_n\boldsymbol{x}_n\boldsymbol{x}_n^\top \succ 0 \;\text{ and }\; \sum_{n=1}^D v_n\boldsymbol{y}_n\boldsymbol{y}_n^\top \prec 0. \tag{11}$$

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 $\boldsymbol{v}$, (11) becomes

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

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 $P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big)$.

E.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 $\delta \in (0,1)$. It suffices to upper bound $P\Big(\boldsymbol{S}_1 \not\succ 0\Big)$ and $P\Big(\boldsymbol{S}_2 \not\prec 0\Big)$ individually due to the union bound. 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 bound $P\Big(\boldsymbol{S}_1 \not\succ 0\Big) = P\Big(\lambda_r(\boldsymbol{S}_1) \le 0\Big)$ using Bernstein's inequality:

$$\begin{aligned}
P\Big(\boldsymbol{S}_1 \not\succ 0\Big) &= P\Big(\lambda_r(\boldsymbol{S}_1) \le 0\Big) = P\Big(\lambda_r(\boldsymbol{S}_1) - \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big) \le -\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) \\
&\overset{(a)}{\le} P\Big(\lambda_r\big(\boldsymbol{S}_1 - \mathbb{E}[\boldsymbol{S}_1]\big) \le -\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) = P\Big(\lambda_1\big(-\boldsymbol{S}_1 - \mathbb{E}[-\boldsymbol{S}_1]\big) \ge \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)\Big) \\
&\overset{(b)}{\le} r\cdot\exp\bigg(-\frac{\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)^2}{16r^2D + 4r\lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big)}\bigg),
\end{aligned} \tag{12}$$

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. Let $Q := \frac{1}{D}\sum_{n=1}^{D}\mathbb{1}[v_n = +1]$. 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

$$\gamma_1 D\boldsymbol{I}_r \preceq \mathbb{E}[\boldsymbol{S}_1] \preceq \gamma_2 D\boldsymbol{I}_r. \tag{13}$$

Therefore, $\gamma_1 D \le \lambda_r\big(\mathbb{E}[\boldsymbol{S}_1]\big) \le \gamma_2 D$. Substituting (13) into (12) 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

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

We then apply the union bound on the failure probability:

$$P\Big(\boldsymbol{S}_1 \not\succ 0 \cup \boldsymbol{S}_2 \not\prec 0\Big) \le 2r\cdot\exp\bigg(-\frac{\gamma_1^2 D}{16r^2 + 4\gamma_2 r}\bigg). \tag{14}$$

E.3 FINAL RESULT

Upper bounding (14) by some (arbitrarily small) $\delta \in (0,1)$, and then re-arranging the terms to lower bound D, results in

$$D \geq \frac{16r^2 + 4\gamma_2 r}{\gamma_1^2} \cdot \log\left(\frac{2r}{\delta}\right). \tag{15}$$

Substituting the definitions of γ_1 and γ_2, as well as $\theta_{min} := \theta_1$, into (15) 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

$$D \geq \frac{2\pi\left(4r^2 + \sqrt{\sum_{\ell=1}^{r} \sin^2(\theta_\ell)}\right)(r+1)}{\sin^2(\theta_{min})} \cdot \log\left(\frac{2r}{\delta}\right) = \mathcal{O}\left(\frac{r^3}{\sin^2(\theta_{min})} \cdot \log\left(\frac{r}{\delta}\right)\right).$$

F ADDITIONAL EXPERIMENTAL DETAILS

In this section, we discuss the experimental setup and results for Figures 1, 4 and 5.

F.1 PHASE TRANSITION IN TERMS OF INTRINSIC DIMENSION

In this subsection, we describe the setup and results in Figure 1, which verifies the required network width to achieve linear separability of the initial-layer features grows polynomially w.r.t. the intrinsic dimension. This experiment was run on a MacBook Air with an Apple M3 chip.

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 settings, we tested different layer widths D. For each combination of (D, d) and (D, r), we checked for linear separability using the necessary and sufficient conditions (5), and recorded the proportion of successful trials.

Results. As seen in Figure 1, 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 1 verifies the required width to achieve linear separability of the random features only depends on the intrinsic dimension of the subspaces.

F.2 DEPENDENCE ON DIMENSION AND NUMBER OF CLASSES: QUADRATIC VS. RELU

Here, we describe the setup and results in Figure 4, which shows the required widths of quadratic and ReLU random feature models have similar dependence on the subspace dimension r, and number of classes K, to achieve linear separability of a UoS. All experiments were run on a single NVIDIA A40 GPU.

Setup. For all experiments, we set $d = 128$. In Figure 4a, we set $K = 2$ and swept through r from 2^2 to 2^6 by powers of 2. In Figure 4b, we fixed $r = 4$ 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.

F.3 LINEAR SEPARABILITY OF FEATURES: RANDOM VS. TRAINED WEIGHTS

We first describe the setup and results in Figure 5, which investigates how training the network weights away from their random initialization impacts the linear separability of the initial-layer features. Here, all experiments were run on a single NVIDIA V100 GPU.

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.

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 does not impact the linear separability of the features.