On the Emergence of Invariant Low-Dimensional Subspaces in Gradient Descent for Learning Deep Linear Networks

Can Yaras*[1], Peng Wang*[1], Wei Hu[1], Zhihui Zhu[2], Laura Balzano[1], and Qing Qu[1]

[1]Department of Electrical Engineering & Computer Science, University of Michigan
[2]Department of Computer Science & Engineering, Ohio State University

June 2023

Abstract

Over the past few years, an extensively studied phenomenon in training deep networks is the implicit bias of gradient descent towards parsimonious solutions. In this work, we investigate this phenomenon by narrowing our focus to deep linear networks. Through our analysis, we reveal a surprising "law of parsimony" in the learning dynamics when the data possesses low-dimensional structures. Specifically, we show that the evolution of gradient descent starting from orthogonal initialization only affects a minimal portion of singular vector spaces across all weight matrices. In other words, the learning process happens only within a small invariant subspace of each weight matrix, despite the fact that all weight parameters are updated throughout training. This simplicity in learning dynamics could have significant implications for both efficient training and a better understanding of deep networks. First, the analysis enables us to considerably improve training efficiency by taking advantage of the low-dimensional structure in learning dynamics. We can construct smaller, equivalent deep linear networks without sacrificing the benefits associated with the wider counterparts. Second, it allows us to better understand deep representation learning by elucidating the linear progressive separation and concentration of representations from shallow to deep layers. We also conduct numerical experiments to support our theoretical results. The code for our experiments can be found at https://github.com/cjyaras/lawofparsimony.

1 Introduction

In recent years, deep learning has demonstrated remarkable success across a wide range of applications in engineering and science [1]. Numerous studies have shown that the effectiveness of deep learning is partially due to the implicit bias of its learning dynamics, which favors some particular solutions that generalize exceptionally well without overfitting in the over-parameterized setting [2–5]. To gain insight into the implicit bias of deep networks, a line of recent work has shown that gradient descent (GD) tends to learn simple functions [6–10]. For instance, some studies have shown that gradient descent is biased towards max-margin solutions in linear networks trained for binary classification via separable data [10, 11]. In addition to the simplicity bias, another line of work showed that deep networks trained by GD exhibit a bias towards low-rank solutions [5, 12].

*The first two authors contributed to this work equally.

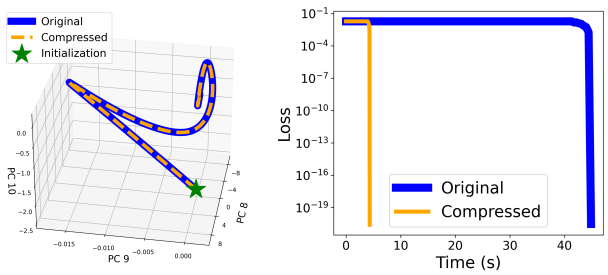


Figure 1: **Efficient training of deep linear networks.** *Left*: Principal components of end-to-end GD trajectories for overparameterized DLN and its equivalent compressed network. *Right*: Training loss vs. wall-time comparison.

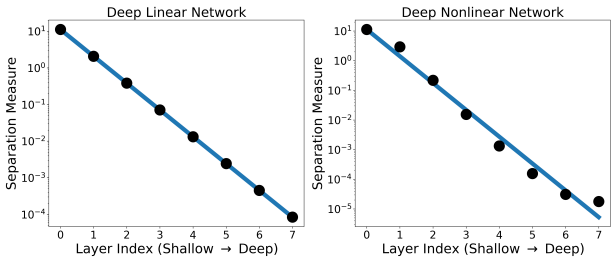


Figure 2: **Progressive collapse with linear decay on deep *linear* and *nonlinear* networks.** The x-axis denotes the layer index and the y-axis denotes the separation measure in (13).

The works [13, 14] demonstrated that adding depth to a matrix factorization enhances an implicit tendency towards low-rank solutions, leading to more accurate recovery.

Despite the abundant empirical evidence in practical nonlinear networks, most theoretical results are developed based on over-parameterized linear models [5, 8, 11, 14]. Notably, deep linear networks (DLNs), defined by multiple hidden layers with identity activations between layers, have been widely used as prototypes of practical deep networks for studying their nonlinear learning dynamics [13, 15–17]. Moreover, despite its simplicity, some properties of DLNs resemble those of their nonlinear counterparts. For example, for both linear and nonlinear networks, the work [5] empirically showed that the low-rank bias exists at both initialization and after training and is resilient to the choice of hyper-parameters and learning methods. The work [16] showed that a DLN exhibits a striking, hierarchical progressive differentiation of structures in its internal hidden representations, which is similar to its nonlinear counterparts. Additionally, [18, 19] demonstrated the practical applications of DLNs.

Contributions. In this work, we investigate a particular implicit bias of GD during the training of DLNs. When the cross-correlation matrix of the training data exhibits a low-dimensional structure, which manifests as either a low-rank or wide matrix in our context, we show that the dynamics of GD tend towards parsimonious solutions. By examining the evolution of GD started from orthogonal initialization, we unveil a surprising, yet common phenomenon:

*The learning process takes place only within a **minimal invariant subspace** of each weight matrix, while the remaining singular subspaces stay **unaffected** throughout training.*

Here, "unaffected" means that the remaining singular subspaces are not influenced or altered by the GD updates throughout the entire training process. Notably, this phenomenon, which we term the "law of parsimony", persists despite the fact that GD updates all parameters of the weight matrices. When we use initialization of small scale, our work sheds new light on the implicit low-rank bias of the GD trajectory, explicitly explaining why deeper networks favor low-rank solutions throughout the entire training process [5]. Moreover, we demonstrate that such a phenomenon could have far-reaching implications for both understanding and improving the training efficiency of DLNs, which we highlight below.

- **Dramatically more efficient training by constructing "equivalent" smaller networks.** As illustrated in Figure 1, since learning only happens within a small *invariant* subspace of the weights, we can construct and train significantly smaller DLNs that share the same learning dynamics as their wider counterparts. As such, we can significantly reduce the computational complexity of training deep networks without sacrificing the benefits of the associated wider networks,

achieving the best of both worlds. We experimentally demonstrate such benefits on applications like deep matrix completion [14], showing that we can obtain the same sampling complexity as the original network while significantly improving the optimization efficiency. We believe such a finding could also have broad applications in training and fine-tuning practical deep networks [20].

- **New theoretical insights into the progressive data separation in deep representation learning.** We show that the law of parsimony plays an important role in deciphering representation learning in the context of multi-class classification problems. Recent works [21–25] have experimentally shown that the features across layers of a deep network exhibit a certain law of data separation that manifests in modern deep architectures during the terminal phases of training. Specifically, each layer of a trained network roughly improves a certain measure of data separation by an *equal* multiplicative factor [22], which is shown in Figure 2. In this work, we theoretically investigate this phenomenon based on DLNs with orthogonal data. Thanks to the "law of parsimony" in the GD trajectory, we can precisely characterize the *linear* progressive separation and concentration of representations from shallow to deep layers, potentially shedding new light on understanding the phenomenon in deep nonlinear networks.

Notations and Organization. Let $\mathbb{R}^n$ be the n-dimensional Euclidean space and $\|\cdot\|$ be the Euclidean norm. Given any $n \in \mathbb{N}$, We use $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ to denote an identity matrix of size n, and $\mathbf{1}_n \in \mathbb{R}^n$ denote an all one vector of length n. Given any $L \in \mathbb{N}$, we use $[L]$ to denote the index set $\{1, \cdots, L\}$. Let $\mathcal{O}^{m\times n} = \{\boldsymbol{X} \in \mathbb{R}^{m\times n} : \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_n\}$ denote the set of all $m \times n$ orthogonal matrices and $\mathcal{O}^n$ the set of all $n \times n$ orthogonal matrices.

The rest of the paper is organized as follows. In Section 2, we introduce the basic problem setup. We present our main theoretical result in Section 3. The applications in deep matrix factorization and progressive collapse are demonstrated in Section 4.1 and Section 4.2, respectively.

2 Problem Formulation

Basic Setup of Deep Linear Networks. Suppose that we have N training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N \subset \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$. Let $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_N] \in \mathbb{R}^{d_x\times N}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1\ \boldsymbol{y}_2\ \ldots\ \boldsymbol{y}_N] \in \mathbb{R}^{d_y\times N}$ and define $\boldsymbol{Y}\boldsymbol{X}^\top$ to be the cross-correlation matrix. The goal of training a deep network is to learn a parameterized, hierarchical function $f : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$ that maps an input $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ to its corresponding label $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ for all $1 \le i \le N$. In this work, we consider an L-layer ($L \ge 2$) linear network $f_{\boldsymbol{\Theta}}(\cdot) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$, parameterized by $\boldsymbol{\Theta} = \{\boldsymbol{W}_l\}_{l=1}^L$ with input $\boldsymbol{x} \in \mathbb{R}^{d_x}$, i.e.,

$$f_{\boldsymbol{\Theta}}(\boldsymbol{x}) \ := \ \boldsymbol{W}_L \cdots \boldsymbol{W}_1 \boldsymbol{x} \ = \ \boldsymbol{W}_{L:1}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d_1\times d_x}$, $\boldsymbol{W}_l \in \mathbb{R}^{d_l\times d_{l-1}}$ $(l = 2, \ldots, L-1)$, and $\boldsymbol{W}_L \in \mathbb{R}^{d_y\times d_{L-1}}$ are weight matrices. For convenience, throughout the paper we adopt the abbreviations $\boldsymbol{W}_{j:i} = \boldsymbol{W}_j \cdots \boldsymbol{W}_i$ and $\boldsymbol{W}_{j:i}^\top = \boldsymbol{W}_i^\top \cdots \boldsymbol{W}_j^\top$ for $j \ge i$, where both are identity if $j < i$.

To learn the network parameters $\boldsymbol{\Theta}$, we consider minimizing the ℓ_2 loss on the training data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ as

$$\min_{\boldsymbol{\Theta}} \ \ell(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{i=1}^N \|f_{\boldsymbol{\Theta}}(\boldsymbol{x}_i) - \boldsymbol{y}_i\|_F^2 = \frac{1}{2}\|\boldsymbol{W}_{L:1}\boldsymbol{X} - \boldsymbol{Y}\|_F^2. \tag{2}$$

Training DLNs via GD. As the network is often over-parameterized, a common approach to enforce implicit regularization in solving the problem is to utilize GD starting from small initialization [13,14]. Here, we use $\boldsymbol{W}_l(t)$ to denote the weight of the l-th layer at the t-th iteration.

- **Orthogonal initialization.** We initialize the weight matrices $\boldsymbol{W}_l(0)$ for all $l \in [L]$ using ε-scaled orthogonal matrices for some $\varepsilon > 0$, i.e.,

$$\boldsymbol{W}_l^\top(0)\boldsymbol{W}_l(0) = \varepsilon^2 \boldsymbol{I} \quad \text{or} \quad \boldsymbol{W}_l(0)\boldsymbol{W}_l^\top(0) = \varepsilon^2 \boldsymbol{I}, \quad \forall l \in [L], \tag{3}$$

  which depends on the size of $\boldsymbol{W}_l$. It is worth noting that orthogonal weight initialization is a commonly employed technique in neural network training due to its ability to speed up the convergence of GD [15,26–29].

- **Learning dynamics of GD.** We update all weights via GD for $t = 1, 2, \dots$ as

$$\boldsymbol{W}_l(t) = (1 - \eta\lambda)\boldsymbol{W}_l(t-1) - \eta \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t-1)), \quad \forall\, l \in [L],$$

  where $\lambda \geq 0$ is an optional weight decay parameter and $\eta > 0$ is the learning rate. Substituting the explicit form of the gradient $\nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta})$ of (2) into the above equation, we obtain the analytical form of GD as

$$\boldsymbol{W}_l(t) = (1 - \eta\lambda)\boldsymbol{W}_l(t-1) - \eta \boldsymbol{W}_{L:l+1}^\top(t-1)\boldsymbol{\Gamma}(t-1)\boldsymbol{W}_{l-1:1}^\top(t-1), \tag{4}$$

  where we denote $\boldsymbol{\Gamma}(t) = (\boldsymbol{W}_{L:1}(t)\boldsymbol{X} - \boldsymbol{Y})\,\boldsymbol{X}^\top$ for simplicity.

3 A Law of Parsimony in Gradient Descent of DLNs

Before stating our main result, we make the following assumption for ease of exposition.

Assumption 1. *The weight matrices are square except the last layer, i.e., $d_1 = d_2 = \cdots = d_{L-1} = d$ for some $d \in \mathbb{N}_+$. In particular, we have $d_x = d$. Also, the input data is* whitened *in the sense that $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}_{d_x}$.*[1]

Note that the above assumptions can be relaxed to give very similar results to the ones we prove here. For example, Assumption 1 can be potentially relaxed to $d_{L-1} \geq d_{L-2} \geq \cdots \geq d_1$. Moreover, empirical evidence suggests that our main results should approximately hold for any well-conditioned $\boldsymbol{X}$, not necessarily restricted to whitened $\boldsymbol{X}$ – we leave this study for future work. Based on the assumption above, we show that all iterates $\boldsymbol{W}_l(t)$ along the GD trajectory exhibit parsimonious structures when the cross-correlation matrix possesses low-dimensional structures.

Theorem 1. *Suppose that an L-layer $f_{\boldsymbol{\Theta}}(\cdot)$ and the training data $(\boldsymbol{X}, \boldsymbol{Y})$ satisfy Assumption 1. We run GD (4) with weight decay parameter λ and learning rate η to train $f_{\boldsymbol{\Theta}}(\cdot)$ starting from ε-scaled orthogonal initialization (3). Then, the iterates $\{\boldsymbol{W}_l(t)\}_{l=1}^L$ for all $t \geq 0$ possess parsimonious structures in the following sense:*

Case 1. *Suppose the cross-correlation $\boldsymbol{Y}\boldsymbol{X}^\top \in \mathbb{R}^{d_y \times d_x}$ is of rank $r \in \mathbb{N}_+$ with $d_y = d_x$, and $m := d_x - 2r > 0$. Then there exist orthogonal matrices $\{\boldsymbol{U}_l\}_{l=1}^L \subseteq \mathcal{O}^d$ and $\{\boldsymbol{V}_l\}_{l=1}^L \subseteq \mathcal{O}^d$ satisfying $\boldsymbol{V}_{l+1} = \boldsymbol{U}_l$ for all $l \in [L-1]$, such that $\boldsymbol{W}_l(t)$ admits the following decomposition*

$$\boldsymbol{W}_l(t) = \boldsymbol{U}_l \begin{bmatrix} \widetilde{\boldsymbol{W}}_l(t) & \boldsymbol{0} \\ \boldsymbol{0} & \rho(t)\boldsymbol{I}_m \end{bmatrix} \boldsymbol{V}_l^\top \tag{5}$$

[1] For any full rank $\boldsymbol{X} \in \mathbb{R}^{d_x \times N}$ with $N \geq d_x$, whitened data can always be obtained with a data pre-processing step such as preconditioning.

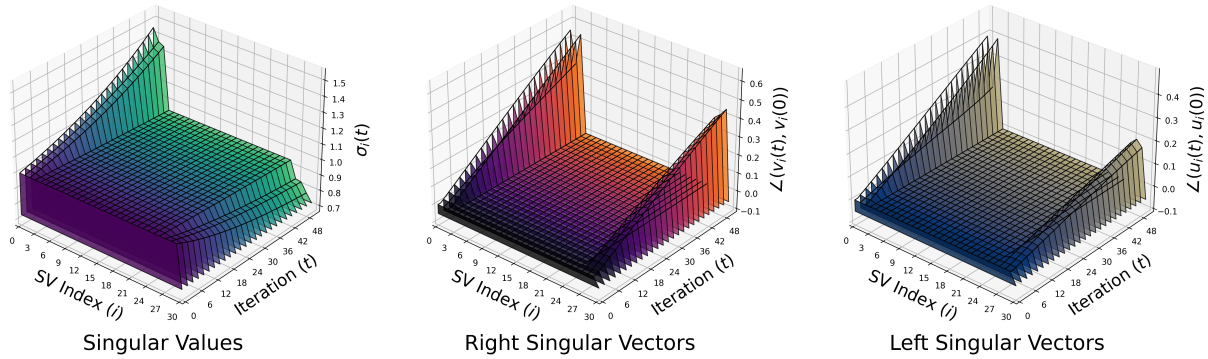


Figure 3: **Evolution of SVD of weight matrices.** We visualize the SVD of the first layer weight matrix of an $L = 3$ layer deep linear network with $d_x = d_y = 30$ and $r = 3$ (**Case 1**) throughout GD without weight decay. *Left*: Magnitude of the i-th singular value $\sigma_i(t)$ at iteration t. *Middle*: Angle $\angle(\boldsymbol{v}_i(t), \boldsymbol{v}_i(0))$ between the i-th right singular vector at iteration t and initialization. *Right*: Angle $\angle(\boldsymbol{u}_i(t), \boldsymbol{u}_i(0))$ between the i-th left singular vector at iteration t and initialization.

for all $l \in [L]$ and $t \geq 0$, where $\widetilde{\boldsymbol{W}}_l(t) \in \mathbb{R}^{2r \times 2r}$ for all $l \in [L]$ with $\widetilde{\boldsymbol{W}}_l(0) = \varepsilon \boldsymbol{I}_{2r}$, and

$$\rho(t) = \rho(t-1)\left(1 - \eta\lambda - \eta \cdot \rho(t-1)^{2(L-1)}\right) \tag{6}$$

for all $t \geq 1$ with $\rho(0) = \varepsilon$.

Case 2. *Suppose the cross-correlation $\boldsymbol{Y}\boldsymbol{X}^\top \in \mathbb{R}^{d_y \times d_x}$ with $d_y = r$ satisfies $m := d_x - 2d_y > 0$. Then, $\boldsymbol{W}_l(t)$ admits the same decomposition as in (5) for all $l \in [L-1]$ and $t \geq 0$ except*

$$\rho(t) = \varepsilon\,(1 - \eta\lambda)^t\,,\ \forall t \geq 0. \tag{7}$$

We defer comprehensive comparison with prior arts and detailed proofs to Section 5 and Appendix B, respectively. To help the reader gain more insights from our results, we make several remarks in the following.

Dynamics of singular values and vectors of weight matrices. It is worth noting that the decomposition (5) is closely related to the singular value decomposition (SVD) of $\boldsymbol{W}_l(t)$. Specifically, let $\boldsymbol{U}_l = [\boldsymbol{U}_{l,1}\ \boldsymbol{U}_{l,2}]$, $\boldsymbol{V}_l = [\boldsymbol{V}_{l,1}\ \boldsymbol{V}_{l,2}]$, where $\boldsymbol{U}_{l,1}, \boldsymbol{V}_{l,1} \in \mathcal{O}^{d \times 2r}$, $\boldsymbol{U}_{l,2}, \boldsymbol{V}_{l,2} \in \mathcal{O}^{d \times (d-2r)}$. Let $\widetilde{\boldsymbol{W}}_l(t) = \widetilde{\boldsymbol{U}}_l(t)\widetilde{\boldsymbol{\Sigma}}_l(t)\widetilde{\boldsymbol{V}}_l^\top(t)$ be an SVD of $\widetilde{\boldsymbol{W}}_l(t)$, where $\widetilde{\boldsymbol{U}}_l(t), \widetilde{\boldsymbol{V}}_l(t) \in \mathcal{O}^{2r}$ and $\widetilde{\boldsymbol{\Sigma}}_l(t) \in \mathbb{R}^{2r \times 2r}$ is a diagonal matrix. Then, we can rewrite (5) into

$$\boldsymbol{W}_l(t) = \begin{bmatrix} \boldsymbol{U}_{l,1}\widetilde{\boldsymbol{U}}_l(t) & \boldsymbol{U}_{l,2} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\Sigma}}_l(t) & \boldsymbol{0} \\ \boldsymbol{0} & \rho(t)\boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{l,1}\widetilde{\boldsymbol{V}}_l(t) & \boldsymbol{V}_{l,2} \end{bmatrix}^\top, \tag{8}$$

which is essentially an SVD of $\boldsymbol{W}_l(t)$ (besides the ordering of singular values). According to this, we can verify that the (repeated) singular value $\rho(t)$ undergoes minimal changes across iterations when ε is small according to (6) and (7) – this is illustrated in Figure 3 (left).

Invariance of subspaces. According to (8), it is evident that the subspace of dimension $d - 2r$ formed by left (resp. right) singular vectors in $\boldsymbol{U}_{l,2}$ (resp. $\boldsymbol{V}_{l,2}$) corresponding to the singular values in $\rho(t)$ remains unchanged during iterations; see Figure 3. This indicates that the learning process occurs only within an invariant subspace of dimension $2r$. This result allows us to gain insights into and improve the training efficiency of DLNs for deep matrix completion (Section 4.1), as well as elucidate the linear progressive separation of representations of DLNs (Section 4.2).

Comparison to prior arts. In our analysis, we specifically investigate the impact of weight decay on the implicit bias of GD. Unlike previous work on implicit bias [14, 30–32], which did not explicitly consider weight decay, we carefully examine the effect of this regularization technique. In particular, when weight decay regularizer $\lambda > 0$ is applied, we observe that the singular value $\rho(t)$ tends to zero asymptotically as t goes to infinity. This finding indicates that gradient descent with weight decay is biased towards finding low-rank solutions. Moreover, we believe our result can be generalized to other optimization methods beyond GD, such as Adam [33], AdaGrad [34], and RMSprop [35].

Low-rank implicit bias. We emphasize that our result sheds new light on the implicit low-rank bias of GD. Specifically, it follows from (6) and (7) that $\lim_{\varepsilon \to 0} \rho(t) = 0$ for all $t \geq 0$. This, together with (8), implies that the dynamics of GD are inherently biased towards finding low-rank solutions with a rank of at most $2r$. In contrast to existing work that demonstrates the tendency of GD to find low nuclear-norm solutions [12, 14], we directly show that GD tends to find low-rank solutions.

4 Applications and Experiments

In this section, we focus on showcasing two specific applications of Theorem 1. In Section 4.1, we present the utilization of **Case 1** in Theorem 1 to improve the speed of training in low-rank deep matrix completion. In Section 4.2, we apply the findings from **Case 2** in Theorem 1 to gain insights into the phenomenon of progressive feature collapse in representation learning.

4.1 Application I: Accelerating Deep Low-Rank Matrix Completion

First, we demonstrate how the parsimonious structures of GD in DLNs can be applied to dramatically improve the optimization efficiency of solving deep matrix completion [14].

Problem Setup. We consider the low-rank matrix completion problem [36–38] with ground-truth $\boldsymbol{\Phi} \in \mathbb{R}^{d \times d}$ with $r := \operatorname{rank}(\boldsymbol{\Phi}) \ll d$. Our goal is to recover $\boldsymbol{\Phi}$ from as few number of observations as possible, where the observed entries are encoded by an index matrix $\boldsymbol{\Omega} \in \{0, 1\}^{d \times d}$. To solve the problem, we consider the recent deep matrix factorization approach [14], by optimizing a variant of Problem (2) with $\boldsymbol{Y} = \boldsymbol{\Phi}$ and identity input $\boldsymbol{X} = \boldsymbol{I}_d$, where the objective[2] is defined as

$$\ell_{\mathrm{mc}}(\boldsymbol{\Theta}) := \frac{1}{2} \|\boldsymbol{\Omega} \odot (\boldsymbol{W}_{L:1} - \boldsymbol{\Phi})\|_F^2. \tag{9}$$

When the complete observation $\boldsymbol{\Omega} = \mathbf{1}_d \mathbf{1}_d^\top$ is available, the above problem simplifies to deep matrix factorization, as depicted in Problem (2). Moreover, if the network depth is $L = 2$, Problem (9) reduces to a Burer-Monteiro factorization [39]. Despite its nonconvexity, significant advances have been made in understanding its global optimality and GD convergence under various settings in the past few years [12, 40–49].

Benefits of Deep Networks. In practice, the true rank r is often not known exactly. Instead, we may have a rough estimate of its upper bound $\widehat{r}$, i.e., $r \leq \widehat{r} \ll d$. When we consider Problem (9) in the over-parameterized regime by overestimating the rank as $\widehat{r} \geq r$, as recent work has shown that GD starting from small initialization has an implicit bias towards the true rank solution [12], we do not need an explicit regularization and set the weight decay $\lambda = 0$ when we run GD for optimizing Problem (9). Furthermore, in the over-parameterized regime, more recent work [14] demonstrated

[2] The recovery error is defined by flipping the entries in $\boldsymbol{\Omega}$ above, i.e., the error in the unobserved entries.

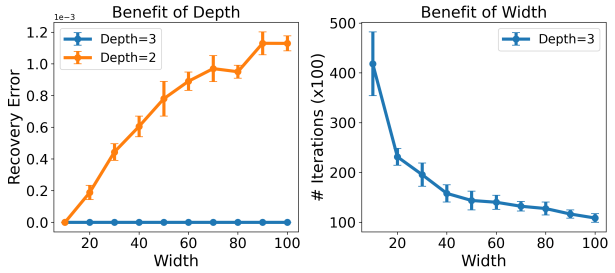


Figure 4: **Benefits of depth and width in over-parameterized matrix completion.** *Left*: Recovery error using $L = 2$ vs. $L = 3$. *Right*: Number of GD iterations to converge.

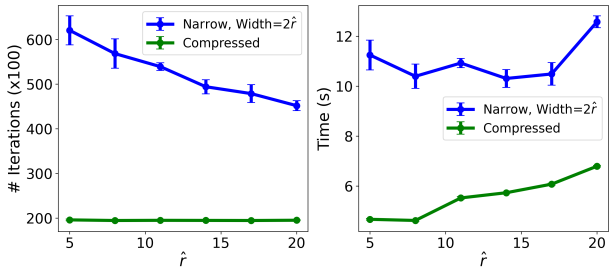


Figure 5: **Comparison of efficiency of compressed networks vs. narrow network** with different overestimated $\widehat{r}$. *Left*: Number of iterations to converge. *Right*: Time to converge.

that using a deeper network (i.e., $L \geq 3$) in solving Problem (9) enjoys several substantial benefits over the shallow counterpart $L = 2$.

- **Benefits of depth.** As shown in Figure 4 (left), when we increase over-parameterization, training deeper networks ($L = 3$) with GD is less prone to overfitting. Additionally, the work [14] has shown that deeper networks improve sample complexity over their shallow counterparts. It has also been shown that GD for deeper networks has a stronger implicit bias towards low-rank solutions [5].

- **Benefits of width.** On the other hand, increasing the width of the network results in accelerated convergence of GD in terms of iterations. As shown in Figure 4 (right), increasing the network width for a 3-layer network reduces the number of GD iterations needed to converge.

Nonetheless, the advantages of deeper and wider networks are accompanied by computational trade-offs: Increasing both the depth and width of a network significantly increases the number of parameters that need to be optimized, thereby causing the per-iteration cost of minimizing Problem (9) via GD to be much greater than that of shallower and narrower networks. In the following, we show that this computational challenge associated with training deeper and wider networks can be mitigated by invoking the law of parsimony in Theorem 1, where we can construct an approximately equivalent but considerably smaller network to speed up the optimization process.

Network Compression via Law of Parsimony. To build up intuition, we first describe our approach in a simplified setting of deep matrix factorization, where we have full observation of $\mathbf{\Phi}$. Then we extend the idea to over-parameterized low-rank matrix completion.

▶ ***The Deep Matrix Factorization Setting:*** With $\mathbf{\Omega} = \mathbf{1}_d\mathbf{1}_d^\top$, (9) now reduces to the vanilla problem (2) where we can apply Theorem 1 with $m = d - 2\widehat{r} > 0$.[3] Based on (5) as well as the fact that $\boldsymbol{V}_{l+1} = \boldsymbol{U}_l$ for all $l \in [L-1]$ from Theorem 1, we can always write the end-to-end matrix $\boldsymbol{W}_{L:1}(t)$ as

$$\boldsymbol{W}_{L:1}(t) = \begin{bmatrix}\boldsymbol{U}_{L,1} & \boldsymbol{U}_{L,2}\end{bmatrix}\begin{bmatrix}\widetilde{\boldsymbol{W}}_{L:1}(t) & \mathbf{0} \\ \mathbf{0} & \rho^L(t)\boldsymbol{I}_m\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{1,1}^\top \\ \boldsymbol{V}_{1,2}^\top\end{bmatrix} = \boldsymbol{U}_{L,1}\widetilde{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{1,1}^\top + \rho^L(t)\boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top$$

for all $t \geq 0$, where $\widetilde{\boldsymbol{W}}_{L:1}(t) \in \mathbb{R}^{2\widehat{r}\times 2\widehat{r}}$ is the end-to-end matrix for all the compressed weights, $\boldsymbol{U}_{L,1}, \boldsymbol{V}_{1,1} \in \mathcal{O}^{d\times 2\widehat{r}}$, and $\boldsymbol{U}_{L,2}, \boldsymbol{V}_{1,2} \in \mathcal{O}^{d\times m}$. Then, our claim for deep matrix factorization is that:

For optimizing (9) with small initialization, running GD on the original weights $\{\boldsymbol{W}_l\}_{l=1}^L \subseteq \mathbb{R}^{d\times d}$ is ***almost equivalent*** *to running GD on the compressed weights $\{\widetilde{\boldsymbol{W}}_l\}_{l=1}^L \subseteq \mathbb{R}^{2\widehat{r}\times 2\widehat{r}}$.*

[3] We note that we can replace r in Theorem 1 with $\widehat{r}$ and maintain the same dynamics provided that $d - 2\widehat{r} > 0$.

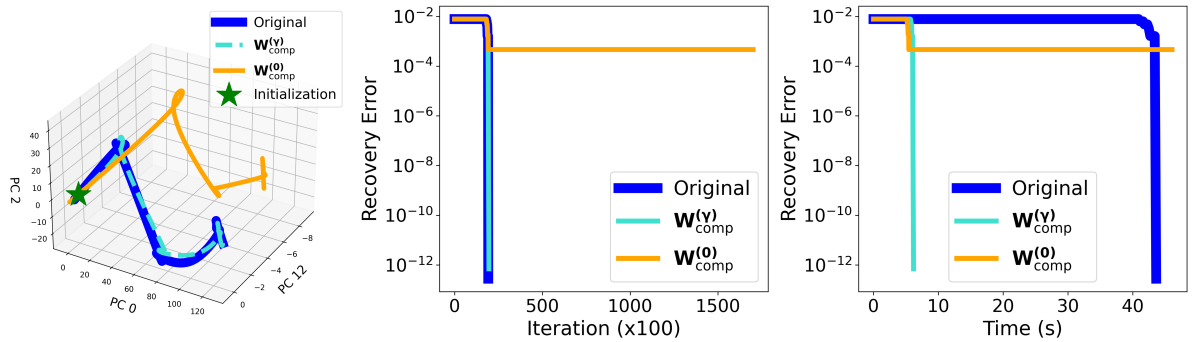


Figure 6: **Ablation of γ in compressed network for deep matrix completion.** Comparison of original overparameterized 3-layer network and its approximate compressed networks $\boldsymbol{W}_{\text{comp}}^{(\gamma)}$ ($\gamma > 0$) and $\boldsymbol{W}_{\text{comp}}^{(0)}$ ($\gamma = 0$). *Left*: Principal components of end-to-end trajectories of each network. *Middle*: Recovery error vs. iteration comparison. *Right*: Recovery error vs wall-time comparison.

As a result, since the compressed weights consist of only $4L\widehat{r}^2$ parameters as opposed to the Ld^2 parameters of the original weights, their optimization can be significantly more efficient when $\widehat{r} \ll d$. To numerically verify our claim, we train the compressed network for target $\boldsymbol{\Phi}$ with $d = 1000$, $\widehat{r} = r = 5$ in Figure 1. It can be observed that the end-to-end GD trajectory of optimizing the compressed network closely follows the trajectory of the original network, while converging to the optimal solution an entire order of magnitude earlier. While our claim may not possess complete rigor, let us briefly outline two fundamental components to elucidate why our claim above holds true:

- **The effects of small initialization ε and depth L.** From (6), we know that $\rho(t)$ depends on ε, with $\lim_{\varepsilon \to 0} \rho(t) = 0$. As such, the term $\rho^L(t) \cdot \boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top \approx \boldsymbol{0}$ when we use small initialization $\varepsilon \approx 0$, and its size decreases with larger depth L. Therefore, we have

$$\boldsymbol{W}_{L:1}(t) \;=\; \boldsymbol{U}_{L,1}\widetilde{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{1,1}^\top + \rho^L(t)\boldsymbol{U}_{L,2}\boldsymbol{V}_{1,2}^\top \;\approx\; \boldsymbol{U}_{L,1}\widetilde{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{1,1}^\top, \quad \forall t \geq 0. \tag{10}$$

- **Invariance of weight subspaces and GD dynamics throughout training.** Moreover, we know from Theorem 1 that all $\{\boldsymbol{U}_{l,1}\}_{l=1}^L$ and $\{\boldsymbol{V}_{l,1}\}_{l=1}^L$ remain unchanged throughout the GD dynamics, and gradients commute under orthogonal transformations across different layers. Therefore, running GD on the original weights $\boldsymbol{W}_l(t)$ is essentially equivalent to running GD on the compressed weights $\widetilde{\boldsymbol{W}}_l(t)$ for all $t \geq 0$ and $l \in [L]$.

► ***Extension to Deep Matrix Completion:*** We should mention that our result for deep matrix factorization cannot be directly applied to deep matrix completion – this is due to the fact that the observation matrix $\boldsymbol{\Omega} \odot \boldsymbol{\Phi}$ for a generic $\boldsymbol{\Omega}$ does not necessarily have the low-rank structure of $\boldsymbol{\Phi}$, and hence the $\boldsymbol{V}_{1,1}$ and $\boldsymbol{U}_{L,1}$ factors in (10) do *not* remain unchanged throughout the GD iterations for all $t \geq 0$. To deal with this issue, we propose to update both $\boldsymbol{V}_{1,1}(t)$ and $\boldsymbol{U}_{L,1}(t)$ factors via GD with learning rate $\gamma\eta$ in

$$\boldsymbol{W}_{\text{comp}}^{(\gamma)}(t) := \boldsymbol{U}_{L,1}(t)\widetilde{\boldsymbol{W}}_{L:1}(t)\boldsymbol{V}_{1,1}^\top(t). \tag{11}$$

This is done simultaneously with the GD updates on the subnetwork $\widetilde{\boldsymbol{W}}_{L:1}(t)$, which uses the original learning rate η. We call $\boldsymbol{W}_{\text{comp}}^{(\gamma)}(t)$ the *compressed network*, where $\gamma \in (0, 1)$ denotes the discrepancy in the learning rate. More specifically, we make the following modifications to our method:

- **Initialize $(\boldsymbol{V}_{1,1}(0), \boldsymbol{U}_{L,1}(0))$.** We initialize the factor $(\boldsymbol{V}_{1,1}(0), \boldsymbol{U}_{L,1}(0))$ using the ones described in (10), but they are calculated based upon $\boldsymbol{\Omega} \odot \boldsymbol{\Phi}$ instead of $\boldsymbol{\Phi}$.

- **Update $(\boldsymbol{V}_{1,1}(t), \boldsymbol{U}_{L,1}(t))$ with small learning rates.** Although the subspaces $(\boldsymbol{V}_{1,1}(t), \boldsymbol{U}_{L,1}(t))$ are changing, they are changing much more slowly compared to the compressed weights $\widetilde{\boldsymbol{W}}_l(t)$. Therefore, we update $\boldsymbol{V}_{1,1}(t)$ and $\boldsymbol{U}_{L,1}(t)$ using a discrepant and smaller learning rate $\gamma\eta$ with $(\gamma \ll 1)$ for all $t \geq 0$, compared to the learning rate η used for the compressed weights $\widetilde{\boldsymbol{W}}_l(t)$.

As illustrated in Figure 6 (left), monitoring the end-to-end GD trajectory on the compressed network $\boldsymbol{W}_{\text{comp}}^{(\gamma)}(t)$ in (11) reveals that maintaining $(\boldsymbol{U}_{L,1}(t), \boldsymbol{V}_{1,1}(t))$ unchanged (i.e., $\gamma = 0$) results in the GD trajectory of $\boldsymbol{W}_{\text{comp}}^{(0)}(t)$ deviating from that of the original network during the later stages of training, leaving the final test error fairly high. This is due to the accumulation of approximation error throughout the final training phase. On the other hand, updating $(\boldsymbol{V}_{1,1}(t), \boldsymbol{U}_{L,1}(t))$ using a small learning rate $\gamma\eta$ with $\gamma = 0.01$ ensures that the GD trajectory of $\boldsymbol{W}_{\text{comp}}^{(\gamma)}(t)$ closely mirrors that of the original network, resulting in a significantly reduced recovery error upon convergence of training.

Regarding computation, it is worth noting that the modified approach requires optimizing an additional $4\widehat{r}d$ parameters compared to the deep matrix factorization scenario – this is due to the additional updates on $(\boldsymbol{V}_{1,1}(t), \boldsymbol{U}_{L,1}(t))$. Nevertheless, compared to optimizing the original network, our modified approach remains considerably more efficient, optimizing only $O(d)$ parameters as opposed to the $O(d^2)$ parameters required for the original network. Our experimental result in Figure 6 (right) also supports this, demonstrating that the compressed network converges significantly faster in terms of time compared to the original network.

Compressed Networks vs. Narrow Networks. Despite the intriguing observation that the effective rank of the changing subspace is capped at $2\widehat{r}$, it still prompts the following question: *Does this imply that optimizing a narrow network of the same width $2\widehat{r}$ would perform just as efficiently as the compressed network with a true width of $d \gg \widehat{r}$?*

Our experimental results suggest that the answer is *no* in general – we compare the training efficiency of a $2\widehat{r}$-compressed network (within a wide network of width $d \gg \widehat{r}$) versus a narrow network with width $2\widehat{r}$ under different over-parameterized estimates $\widehat{r}$. As depicted in Figure 5 (left), the compressed network requires fewer iterations to reach convergence, and the number of iterations necessary is almost unaffected by $\widehat{r}$. Consequently, *training compressed networks is considerably more time-efficient than training narrow networks of the same size*, provided that $\widehat{r}$ is not significantly larger than r. The distinction between the compressed and narrow networks underscores the benefits of wide networks, as previously demonstrated and discussed in Figure 4 (right), where increasing the network width results in faster convergence. However, increasing the network width alone also increases the number of parameters. By employing our network compression methodology, we can achieve the best of both worlds.

Experimental Setups. Regarding our experimental setups, for Figure 4 we consider matrix completion problem with $d = 100$, $r = 10$, and 30% of entries observed, and for Figure 5 and Figure 6, we consider a matrix completion problem with $d = 1000$, $r = 5$, and 20% of entries observed. We optimize deep networks with depth $L = 3$ via GD starting from a small orthogonal initialization of scale $\varepsilon = 10^{-3}$, until the objective (9) achieves a value less than 10^{-10}.

4.2 Application II: Understanding Progressive Feature Collapse in DLN

For the multi-class classification problem, we employ our general result in Theorem 1 to demystify the intriguing phenomenon illustrated in Figure 2. We refer the reader to Appendix C for the proofs of the results presented in this section.

Problem Setup for Multi-Class Classification. We consider a K-class classification problem with training data samples $\{(\boldsymbol{x}_{k,i}, \boldsymbol{y}_k)\}_{i\in[n_k],k\in[K]}$, where $\boldsymbol{x}_{k,i} \in \mathbb{R}^d$ is the i-th sample in the k-th class, $\boldsymbol{y}_k \in \mathbb{R}^K$ is an one-hot label vector[4], and the number of samples n_k in each class is balanced with $n_1 = \cdots = n_K = n$. We denote the total number of samples by $N = nK$. Based upon the above, we train a L-layer DLN to learn weights $\boldsymbol{\Theta} = \{\boldsymbol{W}_l\}_{l=1}^L$ via minimizing the ℓ_2 loss in (2), where $\boldsymbol{W}_l \in \mathbb{R}^{d_l \times d_{l-1}}$, $d_0 = d$, and $d_L = K$. We write the feature $\boldsymbol{z}_{k,i}^l$ of an input sample $\boldsymbol{x}_{k,i}$ in the l-th layer as

$$\boldsymbol{z}_{k,i}^l := \boldsymbol{W}_l \ldots \boldsymbol{W}_1 \boldsymbol{x}_{k,i} = \boldsymbol{W}_{l:1} \boldsymbol{x}_{k,i}, \ \forall l \in [L], \tag{12}$$

and we denote $\boldsymbol{z}_{k,i}^0 = \boldsymbol{x}_{k,i}$. To characterize the network's capability to separate data across layers on the whole training dataset, we use a metric introduced in [50] as

$$D_l \ := \ \mathrm{Tr}(\boldsymbol{\Sigma}_W^l)/\mathrm{Tr}(\boldsymbol{\Sigma}_B^l), \tag{13}$$

where $\boldsymbol{\Sigma}_W^l$ and $\boldsymbol{\Sigma}_B^l$ characterize the between-class and with-class variabilities for the l-th layer respectively as

$$\boldsymbol{\Sigma}_W^l = \frac{1}{N}\sum_{k=1}^K \sum_{i=1}^n \left(\boldsymbol{z}_{k,i}^l - \bar{\boldsymbol{z}}_k^l\right)\left(\boldsymbol{z}_{k,i}^l - \bar{\boldsymbol{z}}_k^l\right)^\top, \ \boldsymbol{\Sigma}_B^l = \frac{1}{K}\sum_{k=1}^K \left(\bar{\boldsymbol{z}}_k^l - \bar{\boldsymbol{z}}^l\right)\left(\bar{\boldsymbol{z}}_k^l - \bar{\boldsymbol{z}}^l\right)^\top \tag{14}$$

for all $l \in [L-1]$, where $\bar{\boldsymbol{z}}_k^l = \frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_{i,k}^l$ denotes the sample of the k-th class for the l-th layer's feature, and $\bar{\boldsymbol{z}}^l = \frac{1}{K}\sum_{k=1}^K \bar{\boldsymbol{z}}_k^l$ denotes the corresponding global sample mean. Intuitively, $\mathrm{Tr}(\boldsymbol{\Sigma}_W^l)$ measures how well the features $\boldsymbol{z}_{k,i}^l$ in the l-th layer collapse to their means in each class, and $\mathrm{Tr}(\boldsymbol{\Sigma}_B)$ measures the discrimination between classes.

Therefore, by using $\mathrm{Tr}(\boldsymbol{\Sigma}_B)$ as a normalization factor, the metric D_l measures how the data are separated and concentrated to the class means in the l-th layer. The smaller the value of D_l is, the more collapsed the features are. Additionally, it should be noted that the metric D_l can be viewed as a simplification of the original metric $\mathrm{Tr}(\boldsymbol{\Sigma}_W^l(\boldsymbol{\Sigma}_B^l)^\dagger)$ extensively studied in [22, 51, 52], by replacing the pseudoinverse with a trace division.

Theoretical Result: Progressive Data Separation with Linear Decay. Based upon the above setup, we are ready to theoretically justify the progressive data separation phenomenon in Figure 2, where the metric D_l decays at least with a linear rate across layers. Our result is based on DLNs under some mild conditions. We defer all proofs to Appendix C.

Theorem 2. *For a K-class classification problem on a balanced dataset, suppose that the input dataset $\boldsymbol{X} \in \mathbb{R}^{d\times N}$ is square and orthogonal. For an L-layer DLN with parameters $\boldsymbol{\Theta} = \{\boldsymbol{W}_l\}_{l=1}^L$ described in (1) with $d_l = d > 2K$ for all $l \in [L-1]$, suppose that $\boldsymbol{\Theta}$ satisfies*
(i) Global Optimality: $\boldsymbol{W}_{L:1}\boldsymbol{X} = \boldsymbol{Y}$.
(ii) Balancedness: There exists a positive constant $\varepsilon \leq \left\{\frac{n^{1/2L}}{\sqrt{30L}\sqrt[4]{d-K}}, \frac{(n/2)^{1/4L}}{\sqrt[4]{d-K}}, \frac{1}{\sqrt{2(\sqrt{K}+1)}}\right\}$ such that $\boldsymbol{W}_{l+1}^\top \boldsymbol{W}_{l+1} = \boldsymbol{W}_l \boldsymbol{W}_l^\top, \forall l \in [L-2]$, $\|\boldsymbol{W}_L^\top \boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^\top\|_F \leq \varepsilon^2\sqrt{d-K}$.
(iii) Unchanged Spectrum: There exists a positive constant $\varepsilon > 0$ and an index set $\mathcal{A} \subseteq [d]$ with $|\mathcal{A}| = d - 2K$ such that for all $l \in [L-1]$ that $\sigma_i(\boldsymbol{W}_l) = \varepsilon$, $\forall i \in \mathcal{A}$.
Then, it holds for all $l = 0, 1, \ldots, L-2$ that

$$D_{l+1}/D_l \leq 2(\sqrt{K}+1)\varepsilon^2. \tag{15}$$

[4] A one-hot label, e.g., $\boldsymbol{y}_k$, has only the k-th entry equal to 1 with the remaining entries equal to 0.

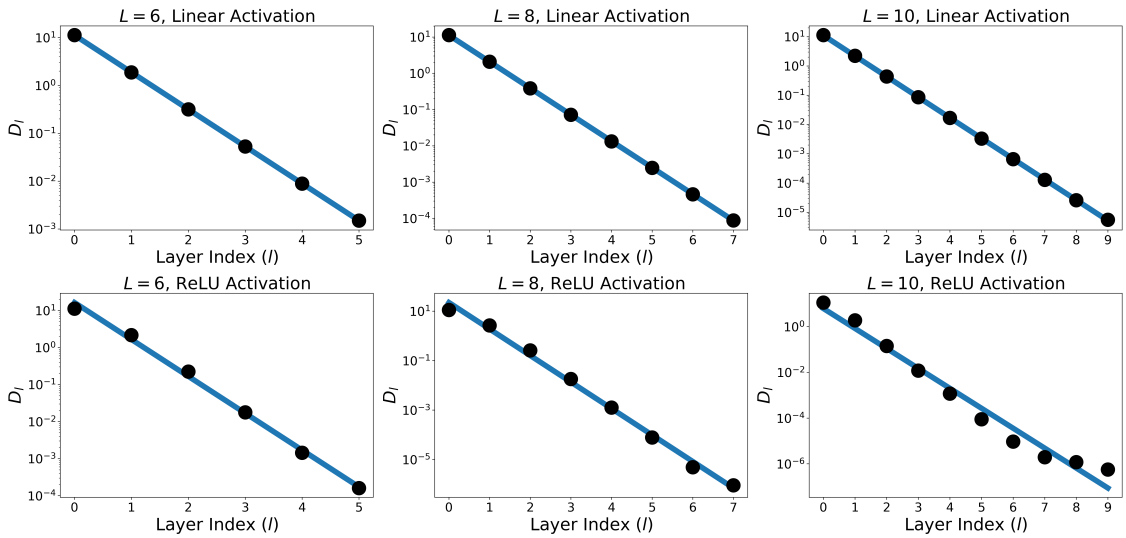


Figure 7: **Linear decay of feature separation in trained deep networks.** Varying the depth L and activation type (Linear vs ReLU), we plot the separation measure D_l in (13) at each layer l up to the penultimate layer $(L-1)$, along with the best log-linear fit line.

Note that (iii) depends on Theorem 1. Based on the above theorem, we can directly verify that DLNs trained by gradient flow (i.e., $\eta \to 0_+$ in (4)) with proper initialization exhibit linear progressive collapse.

Corollary 1. *Consider the setting in Theorem 2. Suppose that we employ gradient flow to train Problem (2), and initialize $\boldsymbol{W}_l(0)$ for all $l \in [L-1]$ to be arbitrary ε-scaled orthogonal matrices satisfying (3) and $\boldsymbol{W}_L(0) = \varepsilon[\boldsymbol{U}_L\ \boldsymbol{0}]$, where $\boldsymbol{U}_L \in \mathcal{O}^K$ and $\varepsilon \leq \left\{ \frac{n^{1/2L}}{\sqrt{30L}\sqrt[4]{d-K}}, \frac{(n/2)^{1/4L}}{\sqrt[4]{d-K}}, \frac{1}{\sqrt{2(\sqrt{K}+1)}} \right\}$. If an optimal solution $\boldsymbol{\Theta}^*$ is found, then (15) holds.*

Experimental Results. We conduct numerical experiments to verify our theoretical results. In the multi-class classification problem, we consider $K = 5$ classes and $n = 10$ samples per class, and randomly generate the input data $\boldsymbol{X} \in \mathbb{R}^{d \times N}$ with $d = N = 50$ such that $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$. We draw the network weights from random ε-scale orthogonal matrices with $\varepsilon = 0.5$, and vary the depth $L \in \{6, 8, 10\}$ as well as activation $\sigma \in \{\text{Linear}, \text{ReLU}\}$. The results are shown in Figure 7. First, we can see that the linear progressive decay phenomenon persists across networks of different depths and even in nonlinear networks. Second, as suggested by (13), the decay rate shown in Figure 7 is *independent* of the depth L, suggesting that deeper networks yield greater feature separation at the penultimate layer. Additional experimental results can be found in Appendix A.

5 Conclusion

This paper offers a comprehensive analysis of the law of parsimony in gradient descent for learning DLNs, contributing to the ongoing pursuit of more efficient and effective deep-learning models. By uncovering the mechanisms that drive parsimonious solutions, we hope to inspire future research and the development of advanced techniques that harness the power of simplicity in deep learning. Ultimately, our goal is to bridge the gap between theory and practice, enabling practitioners to design and train deep learning models with improved efficiency and effectiveness. Finally, we

conclude by providing a survey of related works and subsequently discuss the connections and distinctions between our results and the existing literature.

Deep linear networks. Due to their relative simplicity, deep linear networks are widely used as an alternative approach for investigating the optimization, generalization, and representation characteristics of non-linear networks. Regarding optimization properties, previous works such as [53, 54] have studied the optimization landscape of deep linear networks. Some recent works [17, 55, 56] have established convergence guarantees of gradient descent for training deep linear networks. Many seminal works are devoted to explaining the generalization ability of deep networks via different approaches, such as sharpness [57], neural tangent kernel [58,59], and implicit bias [8,60,61]. In regards to understanding deep representations, recent seminal works [51,52,62] studied an intriguing phenomenon termed neural collapse, which is prevalent across different network architectures, datasets, and training losses during the terminal phase of training. Notably, many researchers have provided theoretical explanations for this phenomenon by assuming the unconstrained feature model [52, 63–68]. This assumption simplifies over-parameterized nonlinear networks into two-layer linear networks, enabling a deeper understanding of the underlying mechanisms.

Implicit bias. In recent years, *implicit bias* (a.k.a. *implicit regularization*) has played an important role in understanding the phenomenon that deep neural networks in the over-parameterized setting often generalize well even when trained without any explicit regularization. Numerous studies have been dedicated to unraveling the mysteries of implicit bias, investigating it from both theoretical and empirical perspectives. In particular, *simplicity bias*, *low-rank bias*, and *spectral bias* have been extensively explored in the literature. Simplicity bias refers to the tendency of gradient descent (GD) for training deep networks to learn only the simplest features over other useful but more complex features [6, 31, 69], e.g., it has been shown [8, 11, 70] that GD favors max-margin solutions in linear models for classification problems. Low-rank bias refers to the notion that deep networks trained by GD are biased toward low-rank solutions [12, 14, 60]. For instance, [12] showed that GD for solving matrix factorization is biased towards minimum nuclear norm solutions. The line of works studied the robustness with overparameterization with implicit bias [71–73]. The works [13, 14] demonstrated that adding depth to matrix factorization enhances an implicit tendency towards minimum nuclear norm solutions, leading to more accurate recovery. Recently, [60] provided substantial empirical observations and concluded that deep networks exhibit an inductive bias towards solutions with lower effective ranks. As for spectral bias, it describes the phenomenon that the learning dynamics of deep networks tends to find low-frequency functions. [74, 75] studied the spectral bias of deep networks using tools from Fourier analysis.

Comparison to existing works. Here, we would like to highlight the differences and connections between existing work and our own. First, most of the existing works analyze the implicit bias by investigating the dynamics of gradient flow [14, 31, 76], whereas our work directly focuses on the dynamics of GD by utilizing the low-dimensional structure of the cross-correlation matrix $\boldsymbol{Y}\boldsymbol{X}^\top$. Our main proof idea is to show that the gradient updates across all iterates are low rank and share a common nullspace throughout the entire training process. Since the weights are initialized orthogonally, the right singular vectors can be chosen arbitrarily to align with the nullspace of gradient updates so that they remain fixed. Second, it is worth noting that we can incorporate weight decay regularization into our analysis. Unlike previous work on implicit bias [14, 30–32] which did not explicitly consider weight decay, we carefully examine the effect of this regularization technique.

In particular, when weight decay regularizer $\lambda > 0$ is applied, we observe that the singular value $\rho(t)$ tends to zero asymptotically as t goes to infinity. This finding indicates that gradient descent with weight decay is also biased towards finding low-rank solutions. Third, compared to the dynamics analysis of SVD of the product matrix (i.e., $\boldsymbol{W}_{L:1}$) in [14], which characterizes its singular values and vectors through partial differential equations, Theorem 1 directly characterizes the dynamics of SVD of each weight matrix (i.e., $\boldsymbol{W}_l$) via revealing the structures of singular values and vectors. More precisely, our results not only accurately predict the singular values of each weight matrix during the training process, as shown in (6) and (7), but also demonstrate that the subspace corresponding to the singular values in $\rho(t)$ remains unchanged throughout the iterations.

Acknowledgment

CY and QQ acknowledge support from NSF CAREER CCF-2143904, NSF CCF-2212066, NSF CCF-2212326, and Google Research Scholar Program. PW and LB acknowledge support from DoE award DE-SC0022186, ARO YIP W911NF1910027, and NSF CAREER CCF-1845076. ZZ acknowledges support from NSF grant CCF-2240708. WH acknowledges support from the Google Research Scholar Program.

References

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

[2] Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[3] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[4] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[5] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023.

[6] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[7] Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019.

[8] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[9] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[10] Daniel Kunin, Atsushi Yamamura, Chao Ma, and Surya Ganguli. The asymmetric maximum margin bias of quasi-homogeneous neural networks. *arXiv preprint arXiv:2210.03820*, 2022.

[11] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[12] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

[13] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[14] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

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

[17] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018.

[18] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] Pierre De Handschutter, Nicolas Gillis, and Xavier Siebert. Deep matrix factorizations. *arXiv preprint arXiv:2010.00380*, 2020.

[20] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[21] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *The Journal of Machine Learning Research*, 21(1):10197–10260, 2020.

[22] Hangfeng He and Weijie J Su. A law of data separation in deep learning. *arXiv preprint arXiv:2210.17020*, 2022.

[23] Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Principled and efficient transfer learning of deep models via neural collapse. *arXiv preprint arXiv:2212.12206*, 2022.

[24] Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso Poggio. Feature learning in deep classifiers through intermediate neural collapse. Technical report, Center for Brains, Minds and Machines (CBMM), 2023.

[25] Shuo Xie, Jiahao Qiu, Ankita Pasad, Li Du, Qing Qu, and Hongyuan Mei. Hidden state variability of pretrained language models can guide computation reduction for transfer learning. *arXiv preprint arXiv:2210.10041*, 2022.

[26] Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. In *International Conference on Machine Learning*, pages 873–882. PMLR, 2018.

[27] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1924–1932. PMLR, 2018.

[28] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018.

[29] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.

[30] Hancheng Min, Salma Tarmoun, René Vidal, and Enrique Mallada. Convergence and implicit bias of gradient flow on overparametrized linear networks. *arXiv preprint arXiv:2105.06351*, 2022.

[31] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. *arXiv preprint arXiv:1909.12051*, 2019.

[32] Gal Vardi and Ohad Shamir. Implicit regularization in relu networks with the square loss. In *Conference on Learning Theory*, pages 4224–4258. PMLR, 2021.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[35] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17, 2012.

[36] Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[37] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[38] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

[39] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[40] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

[41] Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

[42] Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

[43] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

[44] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29, 2016.

[45] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[46] Qiuwei Li, Zhihui Zhu, and Gongguo Tang. The non-convex geometry of low-rank matrix optimization. *Information and Inference: A Journal of the IMA*, 8(1):51–96, 2019.

[47] Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

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

[49] Mahdi Soltanolkotabi, Dominik Stöger, and Changzhi Xie. Implicit balancing and regularization: Generalization and convergence guarantees for overparameterized asymmetric matrix sensing. *arXiv preprint arXiv:2303.14244*, 2023.

[50] Tom Tirer, Haoxiang Huang, and Jonathan Niles-Weed. Perturbation analysis of neural collapse. *arXiv preprint arXiv:2210.16658*, 2022.

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

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

[53] Kenji Kawaguchi. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

[54] Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

[55] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

[56] Armin Eftekhari. Training linear neural networks: Non-local convergence and complexity results. In *International Conference on Machine Learning*, pages 2836–2847. PMLR, 2020.

[57] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[58] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[59] Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.

[60] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

[61] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[62] Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43), 2021.

[63] Can Yaras, Peng Wang, Zhihui Zhu, Laura Balzano, and Qing Qu. Neural collapse with normalized features: A geometric analysis over the riemannian manifold. *arXiv preprint arXiv:2209.09211*, 2022.

[64] XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

[65] Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. *arXiv preprint arXiv:2203.01238*, 2022.

[66] Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[67] Peng Wang, Huikang Liu, Can Yaras, Laura Balzano, and Qing Qu. Linear convergence analysis of neural collapse with unconstrained features. In *OPT 2022: Optimization for Machine Learning* (*NeurIPS 2022 Workshop*), 2022.

[68] Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *Transactions on Machine Learning Research*, 2023.

[69] Depen Morwani, Jatin Batra, Prateek Jain, and Praneeth Netrapalli. Simplicity bias in 1-hidden layer neural networks. *arXiv preprint arXiv:2302.00457*, 2023.

[70] Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3420–3428. PMLR, 2019.

[71] Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations*, 2020.

[72] Chong You, Zhihui Zhu, Qing Qu, and Yi Ma. Robust recovery via implicit bias of discrepant learning rates for double over-parameterization. *Advances in Neural Information Processing Systems*, 33:17733–17744, 2020.

[73] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pages 14153–14172. PMLR, 2022.

[74] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. *arXiv preprint arXiv:1912.01198*, 2019.

[75] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[76] Hancheng Min, Salma Tarmoun, René Vidal, and Enrique Mallada. On the explicit role of initialization on the convergence and implicit bias of overparametrized linear networks. In *International Conference on Machine Learning*, pages 7760–7768. PMLR, 2021.

[77] Gilbert W Stewart and Ji-guang Sun. *Matrix perturbation theory*. Academic press, 1990.

Appendix

In the appendix, we present additional experiments in Appendix A, and provide complete proofs for the technical results of Sections 3 and 4 in Appendix B and Appendix C respectively. Before we proceed, we introduce some further notation. For simplicity, let $\ell(t) = \ell(\boldsymbol{\Theta}(t))$. We use $\sigma_{\max}(\boldsymbol{A})$ (or $\|\boldsymbol{A}\|$), $\sigma_i(\boldsymbol{A})$, and $\sigma_{\min}(\boldsymbol{A})$ to denote the largest, the i-th largest, and the smallest singular values of a matrix $\boldsymbol{A}$, respectively. Given weight matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L$, we denote $\boldsymbol{W}_{j:i} := \boldsymbol{W}_j \boldsymbol{W}_{j-1} \ldots \boldsymbol{W}_i$ if $1 \leq i \leq j \leq L$, and $\boldsymbol{W}_{0:1} := \boldsymbol{I}$ and $\boldsymbol{W}_{L+1:L} = \boldsymbol{I}$. We denote the Kronecker product by $\otimes$.

A Additional Experiments

In this section, we present additional experimental results to supplement those presented in the main paper. We note that all experiments in this work were carried out on a single NVIDIA Tesla V100 GPU.

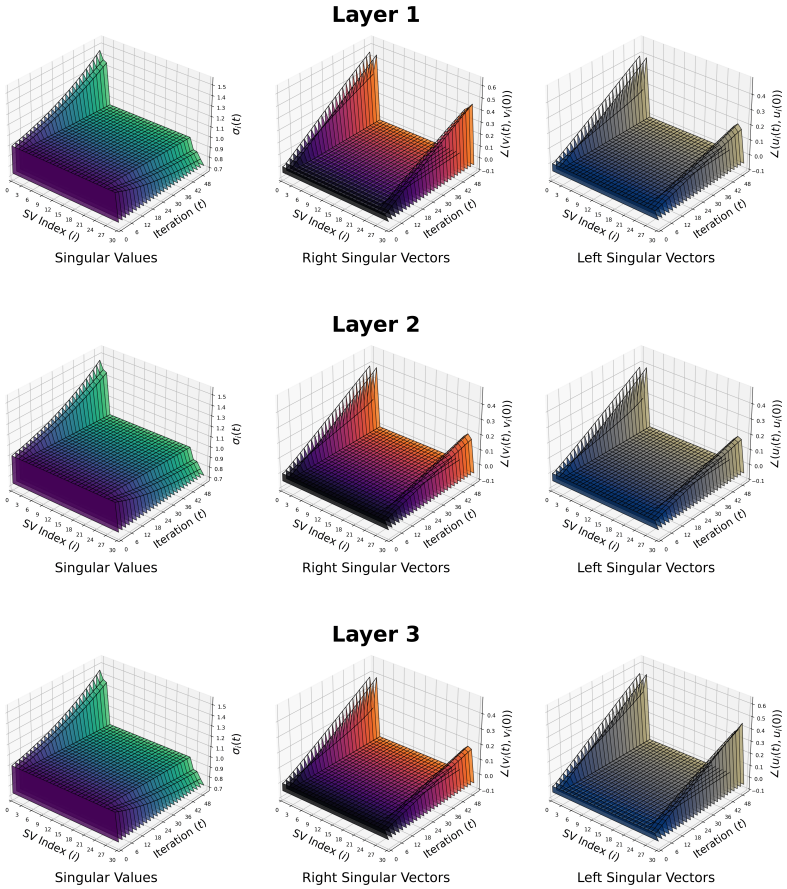


Figure 8: **Evolution of SVD of weight matrices for *all* layers.** We visualize the SVD of the weight matrix for each layer of an $L = 3$ layer deep linear network with $d_x = d_y = 30$ and $r = 3$ (**Case 1**) throughout GD without weight decay. *Left*: Magnitude of the i-th singular value $\sigma_i(t)$ at iteration t. *Middle*: Angle $\angle(\boldsymbol{v}_i(t), \boldsymbol{v}_i(0))$ between the i-th right singular vector at iteration t and initialization. *Right*: Angle $\angle(\boldsymbol{u}_i(t), \boldsymbol{u}_i(0))$ between the i-th left singular vector at iteration t and initialization.

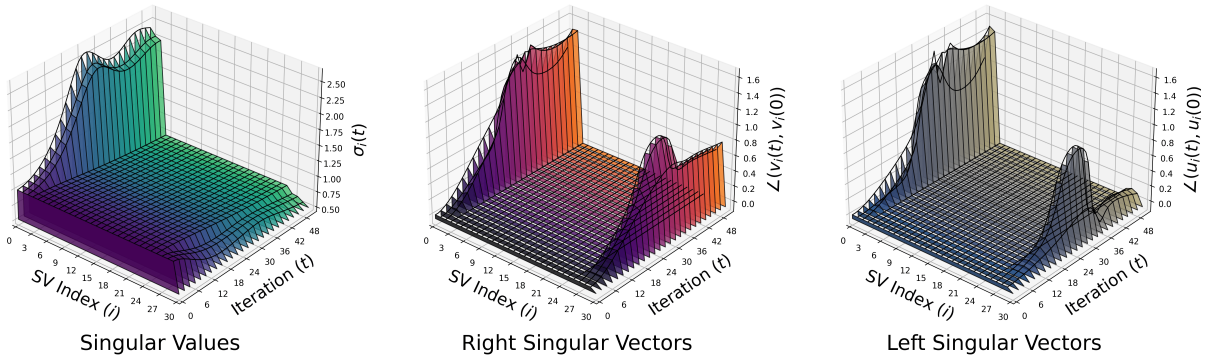


Figure 9: **Evolution of SVD of weight matrices *with momentum*.** We visualize the SVD of the first layer weight matrix of an $L = 3$ layer deep linear network with $d_x = d_y = 30$ and $r = 3$ (**Case 1**) throughout GD *with momentum* (16). *Left*: Magnitude of the i-th singular value $\sigma_i(t)$ at iteration t. *Middle*: Angle $\angle(\boldsymbol{v}_i(t), \boldsymbol{v}_i(0))$ between the i-th right singular vector at iteration t and initialization. *Right*: Angle $\angle(\boldsymbol{u}_i(t), \boldsymbol{u}_i(0))$ between the i-th left singular vector at iteration t and initialization.

First, we extend the visualization in Figure 3 to *all* layers, rather than simply the first layer – the results are shown in Figure 8. We see that the same parsimonious structures persist across all layers throughout GD, as implied by Theorem 1.

Next, we demonstrate that the law of parsimony generalizes to other variants of GD beyond weight decay, such as GD *with momentum*, i.e., we update all weights for $t = 2, 3, \dots$ as

$$\boldsymbol{W}_l(t) = \boldsymbol{W}_l(t-1) - \eta \nabla_{\boldsymbol{W}_l} \ell(\boldsymbol{\Theta}(t-1)) + \mu(\boldsymbol{W}_l(t-1) - \boldsymbol{W}_l(t-2)) \tag{16}$$

for all $l \in [L]$, where $0 \leq \mu < 1$ is the momentum parameter and $\eta > 0$ is the learning rate. We see in Figure 9 that introducing the momentum term still results in a low-dimensional trajectory along minimal singular subspaces. Using a similar approach to the analysis presented in this work, it should be fairly straightforward to extend our results to GD with momentum, but this is left as future work.

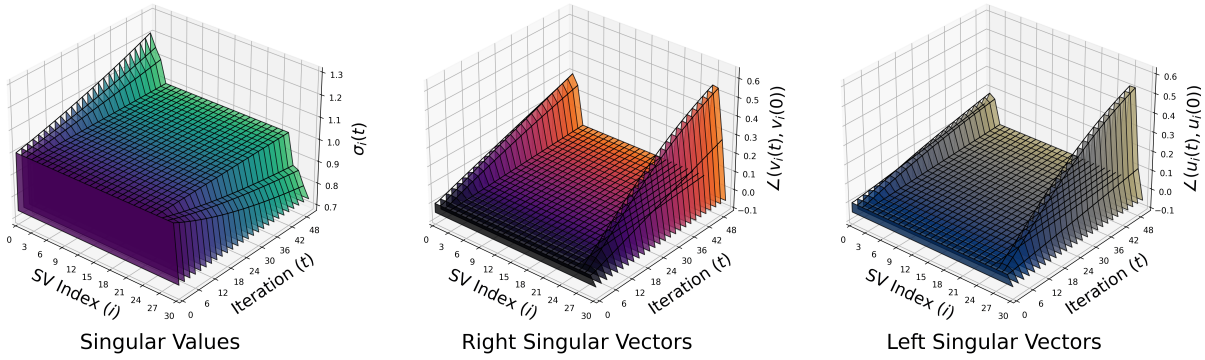


Figure 10: **Evolution of SVD of weight matrices *without whitened data*.** We visualize the SVD of the first layer weight matrix of an $L = 3$ layer deep linear network with $d_x = 30$, $d_y = 3$, and $N = 300$ throughout GD without weight decay. *Left*: Magnitude of the i-th singular value $\sigma_i(t)$ at iteration t. *Middle*: Angle $\angle(\boldsymbol{v}_i(t), \boldsymbol{v}_i(0))$ between the i-th right singular vector at iteration t and initialization. *Right*: Angle $\angle(\boldsymbol{u}_i(t), \boldsymbol{u}_i(0))$ between the i-th left singular vector at iteration t and initialization.

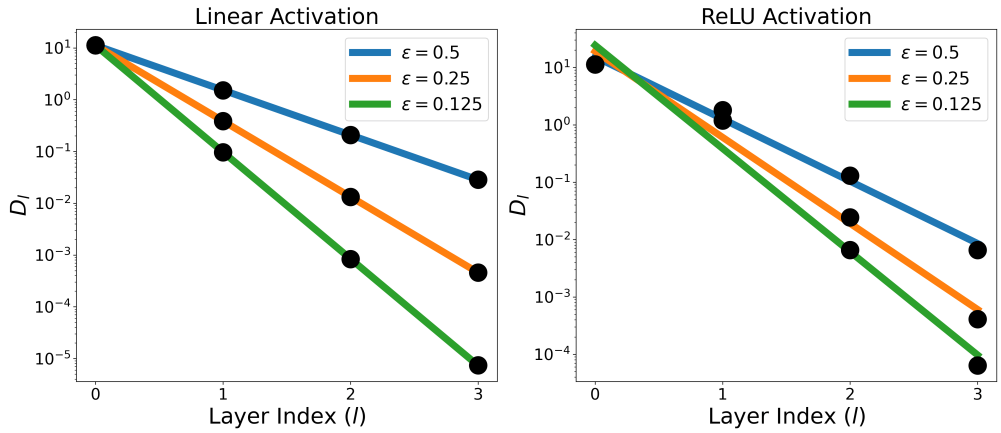


Figure 11: **Linear decay of feature separation in trained deep networks with varying initialization scale ε.** Varying the initialization scale ε for networks with both linear and ReLU activations for an $L = 4$ deep network, we plot the separation measure D_l in (13) at each layer l up to the penultimate layer $(L-1)$, along with the best log-linear fit line.

Besides the low-dimensional structure of $\boldsymbol{Y}\boldsymbol{X}^\top$, the other data assumption made in this work is that the input data is whitened, i.e., $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$. Here, we demonstrate that the law of parsimony continues to hold *approximately* when $\boldsymbol{X}$ is not whitened or preprocessed in any way – in particular, we draw the entries of $\boldsymbol{X} \in \mathbb{R}^{d_x \times N}$ i.i.d. from the standard normal distribution. The results are shown in Figure 10. We can see that most of the singular value remain unchanged from initialization as with before, and the corresponding singular subspaces evolve very little throughout optimization.

To conclude this section, we present several extensions to the experimental results in Section 4.2. First, to investigate the effect of initialization scale ε on the rate of linear progressive collapse, we fix the depth $L = 4$ and vary $\varepsilon \in \{0.5, 0.25, 0.125\}$, keeping all other aspects of the setup the same as in Section 4.2. The results are shown in Figure 11. As implied by (15), we see that decreasing the initialization scale ε leads to a steeper decay in the measure D_l (13) in both linear and nonlinear networks, demonstrating that there is indeed a necessary dependence on ε in the upper bound (15). Therefore, the initialization scale should be chosen carefully when training deep networks in practice, particularly due to the real implications of progressive collapse in transfer learning [23].

Throughout this work, we have assumed that the network weights are initialized *orthogonally* for the sake of analysis – we now empirically verify that the linear progressive collapse phenomenon is *independent* of the type of initialization, i.e., the distribution from which the weight matrices are drawn. With the same experimental set-up as in Section 4.2, we fix the depth $L = 8$ and vary the initialization type among orthogonal, normal, and uniform initializations. The results are shown in Figure 12. To keep the initialization scale comparable between different initialization schemes, for the normally distributed weights, we draw the entries independently from $\mathcal{N}\left(0, \frac{\varepsilon^2}{d}\right)$, and for the uniformly distributed weights, we draw the entries independently from $\mathcal{U}\left(-\frac{3\varepsilon}{d}, \frac{3\varepsilon}{d}\right)$. This guarantees that drawing $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ from any distribution gives $\mathrm{E}[\|\boldsymbol{W}\|_F^2] = d \cdot \varepsilon^2$. As a result, we see that progressive collapse occurs in both linear and nonlinear networks, and in fact decays at the same rate, regardless of the kind of initialization used.

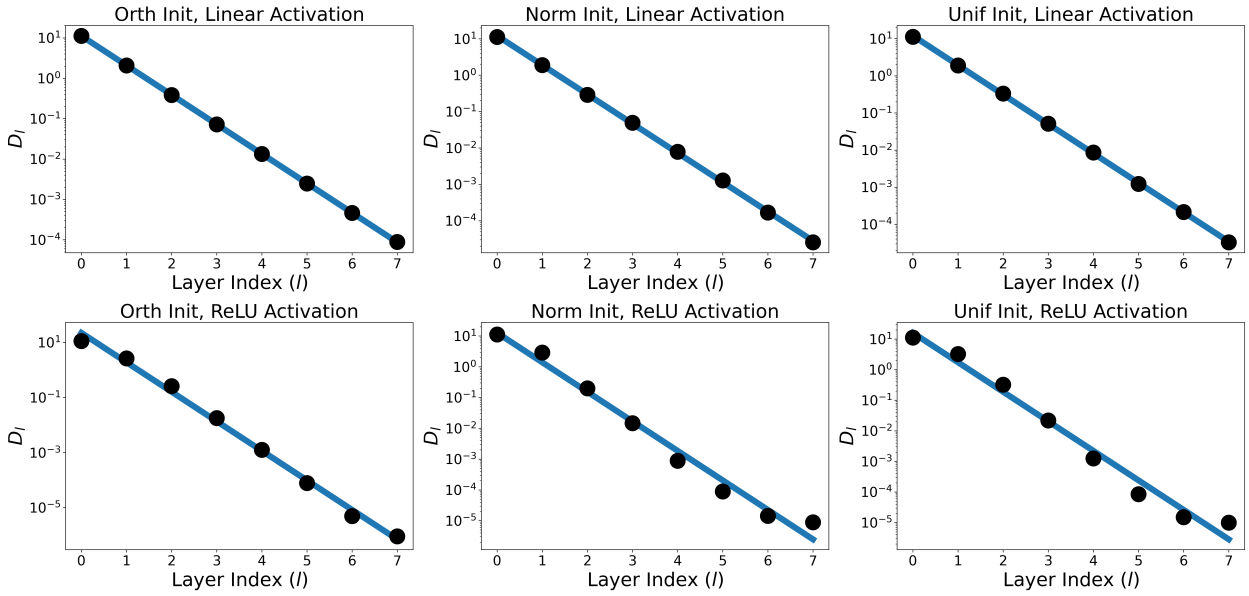


Figure 12: **Linear decay of feature separation in trained deep networks with different initialization types.** Varying the initialization type (orthogonal vs normal vs uniform) and activation type (Linear vs ReLU) for an $L = 8$ deep network, we plot the separation measure D_l in (13) at each layer l up to the penultimate layer $(L-1)$, along with the best log-linear fit line.

B Proofs in Section 3

In this section, we work towards proving Theorem 1 by establishing Lemmas 1 and 2, which imply the conclusions of **Case 1** and **Case 2** respectively. Before proceeding, we note that by the assumption that the data $\boldsymbol{X} \in \mathbb{R}^{d_x \times n}$ is whitened, i.e., $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}_{d_x}$, we have that $\boldsymbol{\Gamma}(t)$ in (4) takes the form

$$\boldsymbol{\Gamma}(t) = \boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}, \text{ where } \boldsymbol{\Phi} = \boldsymbol{Y}\boldsymbol{X}^\top \in \mathbb{R}^{d_y \times d_x} \tag{17}$$

In addition, we note that all statements quantified by i in this section implicity hold for all $i \in [m]$ (as defined in Theorem 1) for the sake of notational brevity.

Lemma 1. *Under the setting of Theorem 1 (**Case 1**), there exist orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m \subset \mathbb{R}^d$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m \subset \mathbb{R}^d$ for $l \in [L]$ satisfying $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$ such that the following hold for all $t \geq 0$:*

$$\begin{aligned}
\mathcal{A}(t) &: \boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} = \rho(t)\boldsymbol{u}_i^{(l)} \quad \forall l \in [L], \\
\mathcal{B}(t) &: \boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} = \rho(t)\boldsymbol{v}_i^{(l)} \quad \forall l \in [L], \\
\mathcal{C}(t) &: \boldsymbol{\Phi}^\top \boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} = \boldsymbol{0} \quad \forall l \in [L], \\
\mathcal{D}(t) &: \boldsymbol{\Phi}\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} = \boldsymbol{0} \quad \forall l \in [L],
\end{aligned}$$

where $\rho(t) = \rho(t-1)\left(1 - \eta\lambda - \eta \cdot \rho(t-1)^{2(L-1)}\right)$ for all $t \geq 1$ with $\rho(0) = \varepsilon$.

Proof. Define $\boldsymbol{\Psi} := \boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Phi}$. Since the rank of $\boldsymbol{\Phi}$ is at most r, we have that the rank of $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d}$ is at most r, which implies that $\dim \mathcal{N}(\boldsymbol{\Psi}) = \dim \mathcal{N}\left(\boldsymbol{\Psi}^\top\right) \geq d - r$. We define the subspace

$$\mathcal{S} := \mathcal{N}(\boldsymbol{\Psi}) \cap \mathcal{N}\left(\boldsymbol{\Psi}^\top \boldsymbol{W}_1(0)\right) \subset \mathbb{R}^d.$$

Since $\boldsymbol{W}_1(0) \in \mathbb{R}^{d\times d}$ is nonsingular, we have

$$\dim(\mathcal{S}) \geq 2(d-r) - d = m.$$

Let $\{\boldsymbol{v}_i^{(1)}\}_{i=1}^m$ denote an orthonormal basis for $\mathcal{S}$ and set $\boldsymbol{u}_i^{(1)} := \boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)}/\varepsilon$, where $\varepsilon > 0$ is the constant in (3) – since $\boldsymbol{W}_1(0)$ is orthogonal, $\{\boldsymbol{u}_i^{(1)}\}_{i=1}^m$ is also an orthonormal set. Then we trivially have $\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \varepsilon\boldsymbol{u}_i^{(1)}$, which, together with (3), implies $\boldsymbol{W}_1^\top(0)\boldsymbol{u}_i^{(1)} = \varepsilon\boldsymbol{v}_i^{(1)}$. It follows from $\boldsymbol{v}_i^{(1)} \in \mathcal{S}$ that $\boldsymbol{\Psi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ and $\boldsymbol{\Psi}^\top\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$, which is equivalent to $\boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ and $\boldsymbol{\Phi}^\top\boldsymbol{W}_{L:2}(0)\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \varepsilon\boldsymbol{\Phi}^\top\boldsymbol{W}_{L:2}(0)\boldsymbol{u}_i^{(1)} = \boldsymbol{0}$ respectively. Since $\boldsymbol{W}_{L:2}^\top(0)$ is full column rank, we further have that $\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$.

Now let $\mathcal{E}(l)$ be the event that we have orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m$ satisfying $\boldsymbol{W}_l(0)\boldsymbol{v}_i^{(l)} = \varepsilon\boldsymbol{u}_i^{(l)}$, $\boldsymbol{W}_l^\top(0)\boldsymbol{u}_i^{(l)} = \varepsilon\boldsymbol{v}_i^{(l)}$, $\boldsymbol{\Phi}^\top\boldsymbol{W}_{L:l+1}(0)\boldsymbol{u}_i^{(l)} = \boldsymbol{0}$, and $\boldsymbol{\Phi}\boldsymbol{W}_{l-1:1}^\top(0)\boldsymbol{v}_i^{(l)} = \boldsymbol{0}$. From the above arguments, we have that $\mathcal{E}(1)$ holds – now suppose $\mathcal{E}(k)$ holds for some $1 \leq k < L$. Set $\boldsymbol{v}_i^{(k+1)} := \boldsymbol{u}_i^{(k)}$ and $\boldsymbol{u}_i^{(k+1)} := \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon$. This, together with (3), implies that $\boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)} = \varepsilon\boldsymbol{u}_i^{(k+1)}$ and $\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)} = \varepsilon\boldsymbol{v}_i^{(k+1)}$. Moreover, we have

$$\begin{aligned}
\boldsymbol{\Phi}^\top\boldsymbol{W}_{L:(k+1)+1}(0)\boldsymbol{u}_i^{(k+1)} &= \boldsymbol{\Phi}^\top\boldsymbol{W}_{L:k+1}(0)\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)}/\varepsilon^2 \\
&= \boldsymbol{\Phi}^\top\boldsymbol{W}_{L:k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon \\
&= \boldsymbol{\Phi}^\top\boldsymbol{W}_{L:k+1}(0)\boldsymbol{u}_i^{(k)}/\varepsilon = \boldsymbol{0},
\end{aligned}$$

where the first two equalities follow from (3) and $\boldsymbol{u}_i^{(k+1)} = \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon$, and the last equality is due to $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$. Similarly, we have

$$\begin{aligned}
\boldsymbol{\Phi}\boldsymbol{W}_{(k+1)-1:1}^\top(0)\boldsymbol{v}_i^{(k+1)} &= \boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{v}_i^{(k+1)} \\
&= \boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} \\
&= \varepsilon\boldsymbol{\Phi}\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{v}_i^{(k)} = \boldsymbol{0},
\end{aligned}$$

where the second equality follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and the third equality is due to $\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} = \varepsilon\boldsymbol{v}_i^{(k)}$. Therefore $\mathcal{E}(k+1)$ holds, so we have $\mathcal{E}(l)$ for all $l \in [L]$. As a result, we have shown the base cases $\mathcal{A}(0)$, $\mathcal{B}(0)$, $\mathcal{C}(0)$, and $\mathcal{D}(0)$.

Now we proceed by induction on $t \geq 0$. Suppose that $\mathcal{A}(t)$, $\mathcal{B}(t)$, $\mathcal{C}(t)$, and $\mathcal{D}(t)$ hold for some $t \geq 0$. First, we show $\mathcal{A}(t+1)$ and $\mathcal{B}(t+1)$. We have

$$\begin{aligned}
\boldsymbol{W}_l(t+1)\boldsymbol{v}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l(t) - \eta\boldsymbol{W}_{L:l+1}^\top(t)\boldsymbol{\Gamma}(t)\boldsymbol{W}_{l-1:1}^\top(t)\right]\boldsymbol{v}_i^{(l)} \\
&= \left[(1-\eta\lambda)\boldsymbol{W}_l(t) - \eta\boldsymbol{W}_{L:l+1}^\top(t)\left(\boldsymbol{W}_{L:1}(t) - \boldsymbol{\Phi}\right)\boldsymbol{W}_{l-1:1}^\top(t)\right]\boldsymbol{v}_i^{(l)} \\
&= (1-\eta\lambda)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} - \eta\boldsymbol{W}_{L:l+1}^\top(t)\boldsymbol{W}_{L:1}(t)\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} \\
&= (1-\eta\lambda)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} - \eta\cdot\rho(t)^{2(L-1)}\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} \\
&= \rho(t)\left(1-\eta\lambda-\eta\cdot\rho(t)^{2(L-1)}\right)\boldsymbol{u}_i^{(l)} = \rho(t+1)\boldsymbol{u}_i^{(l)}
\end{aligned}$$

for all $l \in [L]$, where the first equality follows from (4), the second equality follows from (17), the third equality follows from $\mathcal{D}(t)$, and the fourth equality follows from $\mathcal{A}(t)$ and $\mathcal{B}(t)$ applied

repeatedly along with $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$, proving $\mathcal{A}(t+1)$. Similarly, we have

$$\begin{aligned}
\boldsymbol{W}_l^\top(t+1)\boldsymbol{u}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l^\top(t) - \eta \boldsymbol{W}_{l-1:1}(t)\boldsymbol{\Gamma}^\top(t)\boldsymbol{W}_{L:l+1}(t)\right]\boldsymbol{u}_i^{(l)} \\
&= \left[(1-\eta\lambda)\boldsymbol{W}_l^\top(t) - \eta \boldsymbol{W}_{l-1:1}(t)\left(\boldsymbol{W}_{L:1}^\top(t) - \boldsymbol{\Phi}^\top\right)\boldsymbol{W}_{L:l+1}(t)\right]\boldsymbol{u}_i^{(l)} \\
&= (1-\eta\lambda)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} - \eta \boldsymbol{W}_{l-1:1}(t)\boldsymbol{W}_{L:1}^\top(t)\boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} \\
&= (1-\eta\lambda)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} - \eta \cdot \rho(t)^{2(L-1)}\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} \\
&= \rho(t)\left(1-\eta\lambda-\eta\cdot\rho(t)^{2(L-1)}\right)\boldsymbol{v}_i^{(l)} = \rho(t+1)\boldsymbol{v}_i^{(l)}
\end{aligned}$$

for all $l \in [L]$, where the third equality follows from $\mathcal{C}(t)$, and the fourth equality follows from $\mathcal{A}(t)$ and $\mathcal{B}(t)$ applied repeatedly along with $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$, proving $\mathcal{B}(t+1)$. Now, we show $\mathcal{C}(t+1)$. For any $k \in [L-1]$, it follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and $\mathcal{A}(t+1)$ that

$$\boldsymbol{W}_{k+1}(t+1)\boldsymbol{u}_i^{(k)} = \boldsymbol{W}_{k+1}(t+1)\boldsymbol{v}_i^{(k+1)} = \rho(t+1)\boldsymbol{u}_i^{(k+1)}.$$

Repeatedly applying the above equality for $k = l, l+1, \ldots, L-1$, we obtain

$$\boldsymbol{\Phi}^\top \boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} = \rho(t+1)^{L-l}\boldsymbol{\Phi}^\top\boldsymbol{u}_i^{(L)} = \boldsymbol{0}$$

which follows from $\mathcal{C}(t)$, proving $\mathcal{C}(t+1)$. Finally, we show $\mathcal{D}(t+1)$. For any $k \in \{2, \ldots, L\}$, it follows from $\boldsymbol{v}_i^{(k)} = \boldsymbol{u}_i^{(k-1)}$ and $\mathcal{B}(t+1)$ that

$$\boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{v}_i^{(k)} = \boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{u}_i^{(k-1)} = \rho(t+1)\boldsymbol{v}_i^{(k-1)}.$$

Repeatedly applying the above equality for $k = l, l-1, \ldots, 2$, we obtain

$$\boldsymbol{\Phi}\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} = \boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$$

which follows from $\mathcal{D}(t)$. Thus we have proven $\mathcal{D}(t+1)$, concluding the proof. □

Lemma 2. *Under the setting of Theorem 1 (**Case 2**), there exist orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m \subset \mathbb{R}^d$ for $l \in [L-1]$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m \subset \mathbb{R}^d$ for $l \in [L]$ satisfying $\boldsymbol{v}_i^{(l+1)} = \boldsymbol{u}_i^{(l)}$ for all $l \in [L-1]$ such that the following hold for all $t \geq 0$:*

$$\begin{aligned}
&\mathcal{A}(t) : \boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} = \rho(t)\boldsymbol{u}_i^{(l)} && \forall l \in [L-1], \\
&\mathcal{B}(t) : \boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} = \rho(t)\boldsymbol{v}_i^{(l)} && \forall l \in [L-1], \\
&\mathcal{C}(t) : \boldsymbol{W}_{L:l+1}(t)\boldsymbol{u}_i^{(l)} = \boldsymbol{0} && \forall l \in [L-1], \\
&\mathcal{D}(t) : \boldsymbol{\Gamma}(t)\boldsymbol{W}_{l-1:1}^\top(t)\boldsymbol{v}_i^{(l)} = \boldsymbol{0} && \forall l \in [L],
\end{aligned}$$

where $\rho(t) = \rho(t-1)(1-\eta\lambda)$ for all $t \geq 1$ with $\rho(0) = \varepsilon$.

Proof. Since $\boldsymbol{\Gamma}(0) \in \mathbb{R}^{d_y \times d_x}$, the rank of

$$\nabla_{\boldsymbol{W}_1}\ell(0) = \boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Gamma}(0) \in \mathbb{R}^{d\times d} \tag{18}$$

is at most d_y, which implies that $\dim\mathcal{N}\left(\nabla_{\boldsymbol{W}_1}\ell(0)\right) = \dim\mathcal{N}\left(\nabla_{\boldsymbol{W}_1}^\top\ell(0)\right) \geq d - d_y$. We define the subspace

$$\mathcal{S} := \mathcal{N}\left(\nabla_{\boldsymbol{W}_1}\ell(0)\right) \cap \mathcal{N}\left(\nabla_{\boldsymbol{W}_1}^\top\ell(0)\boldsymbol{W}_1(0)\right) \subset \mathbb{R}^d.$$

Since $\boldsymbol{W}_1(0) \in \mathbb{R}^{d\times d}$ is nonsingular, we have

$$\dim(\mathcal{S}) \geq 2(d - d_y) - d = m.$$

Let $\{\boldsymbol{v}_i^{(1)}\}_{i=1}^m$ denote an orthonormal basis for $\mathcal{S}$ and set $\boldsymbol{u}_i^{(1)} := \boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)}/\varepsilon$, where $\varepsilon > 0$ is the constant in (3) – since $\boldsymbol{W}_1(0)$ is orthogonal, $\{\boldsymbol{u}_i^{(1)}\}_{i=1}^m$ is also an orthonormal set. Then we trivially have $\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \varepsilon\boldsymbol{u}_i^{(1)}$, which, together with (3), implies $\boldsymbol{W}_1^\top(0)\boldsymbol{u}_i^{(1)} = \varepsilon\boldsymbol{v}_i^{(1)}$. It follows from $\boldsymbol{v}_i^{(1)} \in \mathcal{S}$ that $\nabla_{\boldsymbol{W}_1}\ell(0)\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ and $\nabla_{\boldsymbol{W}_1}^\top\ell(0)\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$, which is equivalent to $\boldsymbol{W}_{L:2}^\top(0)\boldsymbol{\Gamma}(0)\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ and $\boldsymbol{\Gamma}^\top(0)\boldsymbol{W}_{L:2}(0)\boldsymbol{W}_1(0)\boldsymbol{v}_i^{(1)} = \varepsilon\boldsymbol{\Gamma}^\top(0)\boldsymbol{W}_{L:2}(0)\boldsymbol{u}_i^{(1)} = \boldsymbol{0}$ respectively by (18). Since $\boldsymbol{W}_{L:2}^\top(0)$ and $\boldsymbol{\Gamma}^\top(0)$ are full column rank, we have $\boldsymbol{\Gamma}(0)\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ and $\boldsymbol{W}_{L:2}(0)\boldsymbol{u}_i^{(1)} = \boldsymbol{0}$. It then follows that

$$\boldsymbol{0} = \boldsymbol{\Gamma}(0)\boldsymbol{v}_i^{(1)} = (\boldsymbol{W}_{L:1}(0) - \boldsymbol{\Phi})\,\boldsymbol{v}_i^{(1)} = \varepsilon\boldsymbol{W}_{L:2}(0)\boldsymbol{u}_i^{(1)} - \boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = -\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)}.$$

Therefore, we have $\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$.

Now let $\mathcal{E}(l)$ be the event that we have orthonormal sets $\{\boldsymbol{u}_i^{(l)}\}_{i=1}^m$ and $\{\boldsymbol{v}_i^{(l)}\}_{i=1}^m$ satisfying $\boldsymbol{W}_l(0)\boldsymbol{v}_i^{(l)} = \varepsilon\boldsymbol{u}_i^{(l)}$, $\boldsymbol{W}_l^\top(0)\boldsymbol{u}_i^{(l)} = \varepsilon\boldsymbol{v}_i^{(l)}$, $\boldsymbol{W}_{L:l+1}(0)\boldsymbol{u}_i^{(l)} = \boldsymbol{0}$, and $\boldsymbol{\Gamma}(0)\boldsymbol{W}_{l-1:1}^\top(0)\boldsymbol{v}_i^{(l)} = \boldsymbol{0}$. From the above arguments, we have that $\mathcal{E}(1)$ holds – now suppose $\mathcal{E}(k)$ holds for some $1 \leq k < L-1$. Set $\boldsymbol{v}_i^{(k+1)} := \boldsymbol{u}_i^{(k)}$ and $\boldsymbol{u}_i^{(k+1)} := \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon$. This, together with (3), implies that $\boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)} = \varepsilon\boldsymbol{u}_i^{(k+1)}$ and $\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)} = \varepsilon\boldsymbol{v}_i^{(k+1)}$. Moreover, we have

$$\begin{aligned}
\boldsymbol{W}_{L:(k+1)+1}(0)\boldsymbol{u}_i^{(k+1)} &= \boldsymbol{W}_{L:k+1}(0)\boldsymbol{W}_{k+1}^\top(0)\boldsymbol{u}_i^{(k+1)}/\varepsilon^2 \\
&= \boldsymbol{W}_{L:k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon \\
&= \boldsymbol{W}_{L:k+1}(0)\boldsymbol{u}_i^{(k)}/\varepsilon = \boldsymbol{0},
\end{aligned}$$

where the first two equalities follow from (3) and $\boldsymbol{u}_i^{(k+1)} = \boldsymbol{W}_{k+1}(0)\boldsymbol{v}_i^{(k+1)}/\varepsilon$, and the last equality is due to $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$. Similarly, we have

$$\begin{aligned}
\boldsymbol{\Gamma}(0)\boldsymbol{W}_{(k+1)-1:1}^\top(0)\boldsymbol{v}_i^{(k+1)} &= \boldsymbol{\Gamma}(0)\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{v}_i^{(k+1)} \\
&= \boldsymbol{\Gamma}(0)\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} \\
&= \varepsilon\boldsymbol{\Gamma}(0)\boldsymbol{W}_{k-1:1}^\top(0)\boldsymbol{v}_i^{(k)} = \boldsymbol{0},
\end{aligned}$$

where the second equality follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and the third equality is due to $\boldsymbol{W}_k^\top(0)\boldsymbol{u}_i^{(k)} = \varepsilon\boldsymbol{v}_i^{(k)}$. Therefore $\mathcal{E}(k+1)$ holds, so we have $\mathcal{E}(l)$ for all $l \in [L-1]$. Finally, setting $\boldsymbol{v}_i^{(L)} = \boldsymbol{u}_i^{(L-1)}$, we have

$$\begin{aligned}
\boldsymbol{\Gamma}(0)\boldsymbol{W}_{L-1:1}^\top(0)\boldsymbol{v}_i^{(L)} &= \boldsymbol{\Gamma}(0)\boldsymbol{W}_{L-2:1}^\top(0)\boldsymbol{W}_{L-1}^\top(0)\boldsymbol{v}_i^{(L)} \\
&= \boldsymbol{\Gamma}(0)\boldsymbol{W}_{L-1:1}^\top(0)\boldsymbol{W}_{L-1}^\top(0)\boldsymbol{u}_i^{(L-1)} \\
&= \varepsilon\boldsymbol{\Gamma}(0)\boldsymbol{W}_{L-1:1}^\top(0)\boldsymbol{v}_i^{(L-1)} = \boldsymbol{0}.
\end{aligned}$$

As a result, we have shown the base cases $\mathcal{A}(0)$, $\mathcal{B}(0)$, $\mathcal{C}(0)$, and $\mathcal{D}(0)$.

Now we proceed by induction on $t \geq 0$. Suppose that $\mathcal{A}(t)$, $\mathcal{B}(t)$, $\mathcal{C}(t)$, and $\mathcal{D}(t)$ hold for some $t \geq 0$. First, we show $\mathcal{A}(t+1)$ and $\mathcal{B}(t+1)$. We have

$$\begin{aligned}
\boldsymbol{W}_l(t+1)\boldsymbol{v}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l(t) - \eta\boldsymbol{W}_{L:l+1}^\top(t)\boldsymbol{\Gamma}(t)\boldsymbol{W}_{l-1:1}^\top(t)\right]\boldsymbol{v}_i^{(l)} \\
&= (1-\eta\lambda)\boldsymbol{W}_l(t)\boldsymbol{v}_i^{(l)} = \rho(t)(1-\eta\lambda)\boldsymbol{u}_i^{(l)} = \rho(t+1)\boldsymbol{u}_i^{(l)}
\end{aligned}$$

for all $l \in [L-1]$, where the first equality follows from (4), the second equality uses $\mathcal{D}(t)$, and the last equality is due to $\mathcal{A}(t)$, proving $\mathcal{A}(t+1)$. Similarly, we have

$$\begin{aligned}\boldsymbol{W}_l^\top(t+1)\boldsymbol{u}_i^{(l)} &= \left[(1-\eta\lambda)\boldsymbol{W}_l^\top(t) - \eta\boldsymbol{W}_{l-1:1}(t)\boldsymbol{\Gamma}^\top(t)\boldsymbol{W}_{L:l+1}(t)\right]\boldsymbol{u}_i^{(l)} \\ &= (1-\eta\lambda)\boldsymbol{W}_l^\top(t)\boldsymbol{u}_i^{(l)} = \rho(t)(1-\eta\lambda)\boldsymbol{v}_i^{(l)} = \rho(t+1)\boldsymbol{v}_i^{(l)}\end{aligned}$$

for all $l \in [L-1]$, where the second equality uses $\mathcal{C}(t)$, and the last equality is due to $\mathcal{B}(t)$, proving $\mathcal{B}(t+1)$. Now, we show $\mathcal{C}(t+1)$. For any $k \in [L-2]$, it follows from $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{u}_i^{(k)}$ and $\mathcal{A}(t+1)$ that

$$\boldsymbol{W}_{k+1}(t+1)\boldsymbol{u}_i^{(k)} = \boldsymbol{W}_{k+1}(t+1)\boldsymbol{v}_i^{(k+1)} = \rho(t+1)\boldsymbol{u}_i^{(k+1)}.$$

Repeatedly applying the above equality for $k = l, l+1, \ldots, L-2$, we obtain

$$\begin{aligned}\boldsymbol{W}_{L:l+1}(t+1)\boldsymbol{u}_i^{(l)} &= \rho(t+1)^{L-l-1}\boldsymbol{W}_L(t+1)\boldsymbol{u}_i^{(L-1)} \\ &= \rho(t+1)^{L-l-1}\left[(1-\eta\lambda)\boldsymbol{W}_L(t) - \eta\boldsymbol{\Gamma}(t)\boldsymbol{W}_{L-1:1}^\top(t)\right]\boldsymbol{u}_i^{(L-1)} = \boldsymbol{0}\end{aligned}$$

where $\boldsymbol{W}_L(t)\boldsymbol{u}_i^{(L-1)} = \boldsymbol{0}$ follows from $\mathcal{C}(t)$ and $\boldsymbol{\Gamma}(t)\boldsymbol{W}_{L-1:1}^\top(t)\boldsymbol{u}_i^{(L-1)} = \boldsymbol{0}$ follows from $\boldsymbol{u}_i^{(L-1)} = \boldsymbol{v}_i^{(L)}$ and $\mathcal{D}(t)$, proving $\mathcal{C}(t+1)$. Finally, we show $\mathcal{D}(t+1)$. For any $k \in \{2, \ldots, L\}$, it follows from $\boldsymbol{v}_i^{(k)} = \boldsymbol{u}_i^{(k-1)}$ and $\mathcal{B}(t+1)$ that

$$\boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{v}_i^{(k)} = \boldsymbol{W}_{k-1}^\top(t+1)\boldsymbol{u}_i^{(k-1)} = \rho(t+1)\boldsymbol{v}_i^{(k-1)}.$$

Repeatedly applying the above equality for $k = l, l-1, \ldots, 2$, we obtain

$$\begin{aligned}\boldsymbol{\Gamma}(t+1)\boldsymbol{W}_{l-1:1}^\top(t+1)\boldsymbol{v}_i^{(l)} &= \rho(t+1)^{l-1}\boldsymbol{\Gamma}(t+1)\boldsymbol{v}_i^{(1)} \\ &= \rho(t+1)^{l-1}(\boldsymbol{W}_{L:1}(t+1) - \boldsymbol{\Phi})\boldsymbol{v}_i^{(1)} = \boldsymbol{0}\end{aligned}$$

where the last line follows from $\boldsymbol{\Phi}\boldsymbol{v}_i^{(1)} = \boldsymbol{0}$ as well as

$$\boldsymbol{W}_{L:1}(t+1)\boldsymbol{v}_i^{(1)} = \boldsymbol{W}_{L:2}(t+1)\boldsymbol{W}_1(t+1)\boldsymbol{v}_i^{(1)} = \rho(t+1)\boldsymbol{W}_{L:2}(t+1)\boldsymbol{u}_i^{(1)} = \boldsymbol{0}$$

by $\mathcal{A}(t+1)$ and $\mathcal{C}(t+1)$. Thus we have proven $\mathcal{D}(t+1)$, concluding the proof. □

Proof of Theorem 1. We show the result for **Case 1** – the proof is nearly identical for **Case 2**.

By $\mathcal{A}(t)$ and $\mathcal{B}(t)$ of Lemma 1, there exists orthonormal matrices $\{\boldsymbol{U}_{l,2}\} \subset \mathbb{R}^{d\times m}$ and $\{\boldsymbol{V}_{l,2}\} \subset \mathbb{R}^{d\times m}$ for $l \in [L]$ satisfying $\boldsymbol{U}_{l+1,2} = \boldsymbol{V}_{l,2}$ for all $l \in [L-1]$ as well as

$$\boldsymbol{W}_l(t)\boldsymbol{V}_{l,2} = \rho(t)\boldsymbol{U}_{l,2} \quad \text{and} \quad \boldsymbol{W}_l(t)^\top\boldsymbol{U}_{l,2} = \rho(t)\boldsymbol{V}_{l,2} \tag{19}$$

for all $l \in [L]$ and $t \geq 0$, where $\rho(t)$ satisfies (6) for $t \geq 1$ with $\rho(0) = \varepsilon$. First, complete $\boldsymbol{V}_{1,2}$ to an orthonormal basis for $\mathbb{R}^d$ as $\boldsymbol{V}_1 = \begin{bmatrix}\boldsymbol{V}_{1,1} & \boldsymbol{V}_{1,2}\end{bmatrix} \in \mathcal{O}^d$. Then for each $l \in [L-1]$, set $\boldsymbol{U}_l = \begin{bmatrix}\boldsymbol{U}_{l,1} & \boldsymbol{U}_{l,2}\end{bmatrix} \in \mathcal{O}^d$ where $\boldsymbol{U}_{l,1} = \boldsymbol{W}_l(0)\boldsymbol{V}_{l,1}/\varepsilon$ and $\boldsymbol{V}_{l+1} = \begin{bmatrix}\boldsymbol{V}_{l+1,1} & \boldsymbol{V}_{l+1,2}\end{bmatrix} \in \mathcal{O}^d$ where $\boldsymbol{V}_{l+1,1} = \boldsymbol{U}_{l,1}$, and finally set $\boldsymbol{U}_L = \begin{bmatrix}\boldsymbol{U}_{L,1} & \boldsymbol{U}_{L,2}\end{bmatrix} \in \mathcal{O}^d$ where $\boldsymbol{U}_{L,1} = \boldsymbol{W}_L(0)\boldsymbol{V}_{L,1}/\varepsilon$. We note that $\boldsymbol{V}_{l+1} = \boldsymbol{U}_l$ for each $l \in [L-1]$. Then we have

$$\boldsymbol{U}_{l,1}^\top\boldsymbol{W}_l(t)\boldsymbol{V}_{l,2} = \rho(t)\boldsymbol{U}_{l,1}^\top\boldsymbol{U}_{l,2} = \boldsymbol{0} \tag{20}$$

for all $l \in [L]$, where the first equality follows from (19). Similarly, we also have

$$\boldsymbol{U}_{l,2}^\top\boldsymbol{W}_l(t)\boldsymbol{V}_{l,1} = \rho(t)\boldsymbol{V}_{l,2}^\top\boldsymbol{V}_{l,1} = \boldsymbol{0} \tag{21}$$

for all $l \in [L]$, where the first equality also follows from (19). Therefore, combining (19), (20), and (21) yields

$$\boldsymbol{U}_l^\top \boldsymbol{W}_l(t) \boldsymbol{V}_l = \begin{bmatrix} \boldsymbol{U}_{l,1} & \boldsymbol{U}_{l,2} \end{bmatrix}^\top \boldsymbol{W}_l(t) \begin{bmatrix} \boldsymbol{V}_{l+1,1} & \boldsymbol{V}_{l+1,2} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{W}}_l(t) & \boldsymbol{0} \\ \boldsymbol{0} & \rho(t) \boldsymbol{I}_m \end{bmatrix}$$

for all $l \in [L]$, where $\widetilde{\boldsymbol{W}}_l(0) = \varepsilon \boldsymbol{I}_{2r}$ by construction of $\boldsymbol{U}_{l,1}$. This directly implies (5), completing the proof. □

C Proofs in Section 4

Suppose that $\boldsymbol{\Theta} = \{\boldsymbol{W}_l\}_{l=1}^L$ satisfies
(i) Global Optimality:

$$\boldsymbol{W}_{L:1} \boldsymbol{X} = \boldsymbol{Y}. \tag{22}$$

(ii) Balancedness: There exists a positive constant $\varepsilon \leq \left\{ \frac{n^{1/2L}}{\sqrt{30}L \sqrt[4]{d-K}}, \frac{(n/2)^{1/4L}}{\sqrt[4]{d-K}}, \frac{1}{\sqrt{2(\sqrt{K}+1)}} \right\}$ such that

$$\boldsymbol{W}_{l+1}^\top \boldsymbol{W}_{l+1} = \boldsymbol{W}_l \boldsymbol{W}_l^\top, \forall l \in [L-2], \ \| \boldsymbol{W}_L^\top \boldsymbol{W}_L - \boldsymbol{W}_{L-1} \boldsymbol{W}_{L-1}^\top \|_F \leq \varepsilon^2 \sqrt{d-K}. \tag{23}$$

(iii) Unchanged Spectrum: There exists a positive constant $\varepsilon > 0$ and an index set $\mathcal{A} \subseteq [d]$ with $|\mathcal{A}| = d - 2K$ such that for all $l \in [L-1]$ that

$$\sigma_i(\boldsymbol{W}_l) = \varepsilon, \ \forall i \in \mathcal{A}. \tag{24}$$

For ease of exposition, we introduce some additional notation. In our analysis, we can assume $\boldsymbol{Y} = \boldsymbol{I}_K \otimes \boldsymbol{1}_n^\top$ without loss of generality. This, together with (22) and $\boldsymbol{X}$ is full rank, yields that the rank of $\boldsymbol{W}_l$ is at least K for all $l \in [L]$. For all $l \in [l-1]$, let

$$\boldsymbol{W}_l = \boldsymbol{U}_l \boldsymbol{\Sigma}_l \boldsymbol{V}_l^\top = \begin{bmatrix} \boldsymbol{U}_{l,1} & \boldsymbol{U}_{l,2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{l,1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{l,2} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{l,1}^\top \\ \boldsymbol{V}_{l,2}^\top \end{bmatrix} = \boldsymbol{U}_{l,1} \boldsymbol{\Sigma}_{l,1} \boldsymbol{V}_{l,1}^\top + \boldsymbol{U}_{l,2} \boldsymbol{\Sigma}_{l,2} \boldsymbol{V}_{l,2}^\top, \tag{25}$$

be a singular value decomposition (SVD) of $\boldsymbol{W}_l$, where $\boldsymbol{\Sigma}_l \in \mathbb{R}^{d \times d}$ is diagonal, $\boldsymbol{\Sigma}_{l,1} = \operatorname{diag}(\sigma_{l,1}, \ldots, \sigma_{l,K})$ with $\sigma_{l,1} \geq \cdots \geq \sigma_{l,K} > 0$ being the singular values, and $\boldsymbol{\Sigma}_{l,2} = \operatorname{diag}(\sigma_{l,K+1}, \ldots, \sigma_{l,d})$ with $\sigma_{l,K+1} \geq \cdots \geq \sigma_{l,d} \geq 0$ being the remaining singular values of $\boldsymbol{W}_l$; $\boldsymbol{U}_l \in \mathcal{O}^d$ with $\boldsymbol{U}_{l,1} \in \mathbb{R}^{d \times K}$, $\boldsymbol{U}_{l,2} \in \mathbb{R}^{d \times (d-K)}$; $\boldsymbol{V}_l \in \mathcal{O}^d$ with $\boldsymbol{V}_{l,1} \in \mathbb{R}^{d \times K}$, $\boldsymbol{V}_{l,2} \in \mathbb{R}^{d \times (d-K)}$. Noting that $\boldsymbol{W}_L \in \mathbb{R}^{K \times d}$, let

$$\boldsymbol{W}_L = \boldsymbol{U}_L \boldsymbol{\Sigma}_L \boldsymbol{V}_L^\top = \boldsymbol{U}_L \begin{bmatrix} \boldsymbol{\Sigma}_{L,1} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{L,1}^\top \\ \boldsymbol{V}_{L,2}^\top \end{bmatrix} = \boldsymbol{U}_L \boldsymbol{\Sigma}_{L,1} \boldsymbol{V}_{L,1}^\top \tag{26}$$

be its singular value decomposition (SVD), where $\boldsymbol{\Sigma}_L \in \mathbb{R}^{K \times d}$, and $\boldsymbol{\Sigma}_{L,1} = \operatorname{diag}(\sigma_{L,1}, \ldots, \sigma_{L,K})$ with $\sigma_{L,1} \geq \cdots \geq \sigma_{L,K} > 0$ being the singular values; $\boldsymbol{U}_L \in \mathcal{O}^K$, and $\boldsymbol{V}_L \in \mathcal{O}^d$ with $\boldsymbol{V}_{L,1} \in \mathbb{R}^{d \times K}$, $\boldsymbol{V}_{L,2} \in \mathbb{R}^{d \times (d-K)}$.

Before we prove Theorem 2, we need some preliminary results. We first show that when $\{\boldsymbol{W}_l\}_{l=1}^L$ satisfy (22) and (23), we can bound the leading K singular values of $\boldsymbol{W}_l$ for all $l \in [L]$. Throughout this section, let

$$\delta := \varepsilon^2 \sqrt{d-K} \leq \frac{n^{1/L}}{30L^2}, \tag{27}$$

where the inequality follows from

$$\varepsilon \leq \min\left\{\frac{n^{1/2L}}{\sqrt{30}L\sqrt[4]{d-K}}, \frac{(n/2)^{1/4L}}{\sqrt[4]{d-K}}, \frac{1}{\sqrt{2(\sqrt{K}+1)}}\right\}. \tag{28}$$

Lemma 3. *Suppose that the weights $\{\boldsymbol{W}_l\}_{l=1}^L$ satisfy (22) and (23). Then, it holds that*

$$\left(\sqrt{\frac{n}{2}}\right)^{1/L} \leq \sigma_K\left(\boldsymbol{W}_l\right) \leq \sigma_1\left(\boldsymbol{W}_l\right) \leq \left(\sqrt{2n}\right)^{1/L}, \ \forall l \in [L]. \tag{29}$$

Proof of Lemma 3. It follows from (22) and the fact that $\boldsymbol{X} \in \mathbb{R}^{d\times d}$ is orthogonal that $\boldsymbol{W}_{L:1} = \boldsymbol{Y}\boldsymbol{X}^\top$. This, together with $\boldsymbol{Y} = \boldsymbol{I}_K \otimes \mathbf{1}_n^\top$ and $\boldsymbol{X} \in \mathcal{O}^d$, yields that $\boldsymbol{W}_{L:1}$ is of rank K and

$$\sigma_i(\boldsymbol{W}_{L:1}) = \sqrt{n}, \ \forall i \in [K]. \tag{30}$$

This, together with [55, Lemma 6] and (27), implies $\sigma_1(\boldsymbol{W}_l) \leq \left(\sqrt{2n}\right)^{1/L}$. Using $\boldsymbol{W}_{l+1}^T\boldsymbol{W}_{l+1} = \boldsymbol{W}_l\boldsymbol{W}_l^T$ for all $l \in [L-2]$ in (23), we obtain

$$\begin{aligned}
\|\boldsymbol{W}_{L:1}^\top\boldsymbol{W}_{L:1} - \left(\boldsymbol{W}_1^\top\boldsymbol{W}_1\right)^L\|_F &= \|\boldsymbol{W}_{L:1}^\top\boldsymbol{W}_{L:1} - \boldsymbol{W}_{L-1:1}^\top\boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^\top\boldsymbol{W}_{L-1:1}\|_F \\
&= \|\boldsymbol{W}_{L-1:1}^\top\left(\boldsymbol{W}_L^\top\boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^\top\right)\boldsymbol{W}_{L-1:1}\|_F \\
&\leq \|\boldsymbol{W}_L^\top\boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^\top\|_F \prod_{l=1}^{L-1}\|\boldsymbol{W}_l\|^2 \\
&\leq \delta\left(2n\right)^{(L-1)/L} \leq \frac{n}{15L^2},
\end{aligned}$$

where the second inequality uses (23) and $\|\boldsymbol{W}_l\| \leq (\sqrt{2n})^{1/L}$ for all $l \in [L-1]$, and the last inequality follows from (27). Using (30) and Weyl's inequality, we obtain

$$\begin{aligned}
\sigma_K\left(\left(\boldsymbol{W}_1^\top\boldsymbol{W}_1\right)^L\right) &\geq \sigma_K\left(\boldsymbol{W}_{L:1}^\top\boldsymbol{W}_{L:1}\right) - \|\boldsymbol{W}_{L:1}^\top\boldsymbol{W}_{L:1} - \left(\boldsymbol{W}_1^\top\boldsymbol{W}_1\right)^L\| \\
&\geq n - \|\boldsymbol{W}_{L:1}^\top\boldsymbol{W}_{L:1} - \left(\boldsymbol{W}_1^\top\boldsymbol{W}_1\right)^L\|_F \geq \left(1 - \frac{1}{15L^2}\right)n.
\end{aligned}$$

Therefore, we have $\sigma_K(\boldsymbol{W}_1) \geq \left(\left(1-\frac{1}{15L^2}\right)n\right)^{1/2L}$. Using this and (23), we obtain $\sigma_K(\boldsymbol{W}_l) \geq \left(\left(1-\frac{1}{15L^2}\right)n\right)^{1/2L}$ for all $l \in [L-1]$. This, together with Weyl's inequality and $\|\boldsymbol{W}_L^T\boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^T\|_F \leq \delta$ in (23), yields

$$\begin{aligned}
\sigma_K\left(\boldsymbol{W}_L^T\boldsymbol{W}_L\right) &\geq \sigma_K\left(\boldsymbol{W}_{L-1}^T\boldsymbol{W}_{L-1}\right) - \|\boldsymbol{W}_L^T\boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^T\| \\
&\geq \left(1 - \frac{1}{15L^2}\right)^{1/L} n^{1/L} - \delta \geq \left(\frac{n}{2}\right)^{1/L},
\end{aligned}$$

where the last inequality follows from (27) and

$$1 - \frac{1}{15L^2} \geq \left(\frac{1}{30L^2} + \left(\frac{1}{2}\right)^{1/L}\right)^L, \ \forall L \geq 1,$$

which implies $\sigma_K(\boldsymbol{W}_L) \geq \left(\frac{n}{2}\right)^{1/2L}$. Since $1 - 1/15L^2 \geq 1/2$ for all $L \geq 1$, we also have $\sigma_K(\boldsymbol{W}_l) \geq \left(\frac{n}{2}\right)^{1/2L}$ for all $l \in [L-1]$. Combining all this together yields (29). □

Then, we show that if the weight matrices $\{\boldsymbol{W}_l\}_{l=1}^L$ satisfy (22) and (23), the right singular vectors of $\boldsymbol{W}_l$ are equal or close to the left singular vectors of $\boldsymbol{W}_{l+1}$ for all $l \in [L-1]$.

Lemma 4. *Suppose that $\{\boldsymbol{W}_l\}_{l=1}^L$ satisfy (22) and (23) with δ satisfying (27), and admit the SVD in (25) and (26). Then, it holds that*

$$\boldsymbol{\Sigma}_{l+1} = \boldsymbol{\Sigma}_l,\ \boldsymbol{V}_{l+1}^T\boldsymbol{U}_l = \boldsymbol{I}_d,\ \forall l \in [L-2],\ \|\boldsymbol{\Sigma}_{l,2}^2\|_F \le \delta,\ \forall l \in [L-1], \tag{31}$$

$$\|\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F \le \frac{2\sqrt{\delta}\sqrt[4]{K}}{n^{1/2L}},\ \sigma_{\min}(\boldsymbol{V}_{L,1}^\top\boldsymbol{U}_{L-1,1}) \ge 1 - \frac{2\sqrt{\delta}\sqrt[4]{K}}{n^{1/2L}}. \tag{32}$$

Proof. It follows from $\boldsymbol{W}_{l+1}^\top\boldsymbol{W}_{l+1} = \boldsymbol{W}_l\boldsymbol{W}_l^\top$ for all $l \in [L-2]$ and (25) that

$$\boldsymbol{V}_{l+1}\boldsymbol{\Sigma}_{l+1}^\top\boldsymbol{\Sigma}_{l+1}\boldsymbol{V}_{l+1}^\top = \boldsymbol{U}_l\boldsymbol{\Sigma}_l\boldsymbol{\Sigma}_l^\top\boldsymbol{U}_l^\top.$$

This, together with the argument in [17, Proof of Theorem 1], implies (31). Using $\|\boldsymbol{W}_L^\top\boldsymbol{W}_L - \boldsymbol{W}_{L-1}\boldsymbol{W}_{L-1}^\top\|_F \le \delta$, (25), and (26), we have

$$\begin{aligned}\delta &\ge \|\boldsymbol{V}_L\boldsymbol{\Sigma}_L^\top\boldsymbol{\Sigma}_L\boldsymbol{V}_L^\top - \boldsymbol{U}_{L-1}\boldsymbol{\Sigma}_{L-1}\boldsymbol{\Sigma}_{L-1}^\top\boldsymbol{U}_{L-1}^\top\|_F \\ &= \|\boldsymbol{V}_{L,1}\boldsymbol{\Sigma}_{L,1}^2\boldsymbol{V}_{L,1}^T - \boldsymbol{U}_{L-1,1}\boldsymbol{\Sigma}_{L-1,1}^2\boldsymbol{U}_{L-1,1}^T - \boldsymbol{U}_{L-1,2}\boldsymbol{\Sigma}_{L-1,2}^2\boldsymbol{U}_{L-1,2}^T\|_F.\end{aligned} \tag{33}$$

Obviously, we have $\|\boldsymbol{\Sigma}_L^T\boldsymbol{\Sigma}_L - \boldsymbol{V}_L^T\boldsymbol{U}_{L-1}\boldsymbol{\Sigma}_{L-1}^2\boldsymbol{U}_{L-1}^T\boldsymbol{V}_L\|_F \le \delta$ due to $\boldsymbol{V}_L \in \mathcal{O}^d$. This, together with [55, Lemma 4], implies

$$\|\boldsymbol{\Sigma}_L^T\boldsymbol{\Sigma}_L - \boldsymbol{\Sigma}_{L-1}^2\|_F \le \delta.$$

This, together with (31), implies $\|\boldsymbol{\Sigma}_{l,2}^2\|_F \le \delta$ for all $l \in [L-1]$. Using this and the structures of $\boldsymbol{\Sigma}_L$ and $\boldsymbol{\Sigma}_{L-1}$, we obtain

$$\|\boldsymbol{\Sigma}_{L,1}^2 - \boldsymbol{\Sigma}_{L-1,1}^2\|_F \le \delta,\ \|\boldsymbol{\Sigma}_{L-1,2}^2\|_F \le \delta. \tag{34}$$

Using (33), $\|\boldsymbol{U}^\top\boldsymbol{A}\boldsymbol{U}\|_F \le \|\boldsymbol{A}\|_F$ for any $\boldsymbol{U} \in \mathcal{O}^{d\times(d-K)}$, we further obtain

$$\|\boldsymbol{U}_{L-1,2}^T\boldsymbol{V}_{L,1}\boldsymbol{\Sigma}_{L,1}^2\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2} - \boldsymbol{\Sigma}_{L-1,2}^2\|_F \le \delta.$$

This, together with (34), yields

$$\|\boldsymbol{U}_{L-1,2}^T\boldsymbol{V}_{L,1}\boldsymbol{\Sigma}_{L,1}^2\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F \le 2\delta.$$

It follows from this and Lemma 5 that

$$\|\boldsymbol{\Sigma}_{L,1}\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F \le \sqrt{2\delta}\sqrt[4]{K}. \tag{35}$$

Noting that $\|\boldsymbol{\Sigma}_{L,1}\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F \ge \sigma_{\min}(\boldsymbol{\Sigma}_{L,1})\|\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F$, we have

$$\|\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F \le \frac{\|\boldsymbol{\Sigma}_{L,1}\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,2}\|_F}{\sigma_{\min}(\boldsymbol{\Sigma}_{L,1})} \le \frac{\sqrt{2\delta}\sqrt[4]{K}}{\left(\frac{n}{2}\right)^{1/2L}} \le \frac{2\sqrt{\delta}\sqrt[4]{K}}{n^{1/2L}}, \tag{36}$$

where the second inequality follows from (35) and Lemma 3. Then, we compute

$$\begin{aligned}\sigma_{\min}^2(\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,1}) &= \min_{\|\boldsymbol{x}\|=1}\|\boldsymbol{U}_{L-1,1}^T\boldsymbol{V}_{L,1}\boldsymbol{x}\|^2 = \min_{\|\boldsymbol{x}\|=1}\boldsymbol{x}^T\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1,1}\boldsymbol{U}_{L-1,1}^T\boldsymbol{V}_{L,1}\boldsymbol{x} \\ &= \min_{\|\boldsymbol{x}\|=1}\boldsymbol{x}^T\boldsymbol{V}_{L,1}^T\left(\boldsymbol{I} - \boldsymbol{U}_{L-1,2}\boldsymbol{U}_{L-1,2}^T\right)\boldsymbol{V}_{L,1}\boldsymbol{x} = 1 - \max_{\|\boldsymbol{x}\|=1}\|\boldsymbol{U}_{L-1,2}^T\boldsymbol{V}_{L,1}\boldsymbol{x}\|^2 \\ &\ge 1 - \sigma_{\max}^2(\boldsymbol{U}_{L-1,2}^T\boldsymbol{V}_{L,1}) \ge 1 - \|\boldsymbol{U}_{L-1,2}^T\boldsymbol{V}_{L,1}\|_F^2,\end{aligned}$$

where the third equality follows from $\boldsymbol{U}_{L-1} \in \mathcal{O}^d$. This, together with (36), implies

$$\sigma_{\min}(\boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,1}) \geq \sqrt{1 - \|\boldsymbol{U}_{L-1,2}^T \boldsymbol{V}_{L,1}\|_F^2} \geq 1 - \|\boldsymbol{U}_{L-1,2}^T \boldsymbol{V}_{L,1}\|_F \geq 1 - \frac{2\sqrt{\delta}\sqrt[4]{K}}{n^{1/2L}}.$$

□

Recall that $\bar{\boldsymbol{x}}_k = \sum_{i=1}^n \boldsymbol{x}_{k,i}/n$ and $\bar{\boldsymbol{x}} = \sum_{k=1}^K \bar{\boldsymbol{x}}_k/K$. Let

$$\boldsymbol{\Delta}_W = [\boldsymbol{\delta}_{1,1}, \ldots, \boldsymbol{\delta}_{1,n}, \ldots, \boldsymbol{\delta}_{K,1}, \ldots, \boldsymbol{\delta}_{K,n}] \in \mathbb{R}^{d \times N}, \text{ where } \boldsymbol{\delta}_{k,i} = \boldsymbol{x}_{k,i} - \bar{\boldsymbol{x}}_k, \ \forall k, i, \tag{37}$$

$$\boldsymbol{\Delta}_B = \left[\bar{\boldsymbol{\delta}}_1, \ldots, \bar{\boldsymbol{\delta}}_K\right] \in \mathbb{R}^{d \times K}, \text{ where } \bar{\boldsymbol{\delta}}_k = \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}, \ \forall k \in [K]. \tag{38}$$

This, together with (12), (14), and (13), yields

$$D_0 = \frac{K}{N} \frac{\|\boldsymbol{\Delta}_W\|_F^2}{\|\boldsymbol{\Delta}_B\|_F^2}, \quad D_l = \frac{K}{N} \frac{\|\boldsymbol{W}_{l:1} \boldsymbol{\Delta}_W\|_F^2}{\|\boldsymbol{W}_{l:1} \boldsymbol{\Delta}_B\|_F^2}. \tag{39}$$

Proof of Theorem 2. Using (31) in Lemma 4, there exists a diagonal matrix $\tilde{\boldsymbol{\Sigma}} = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$ with $\sigma_1 \geq \cdots \geq \sigma_d > 0$ such that

$$\boldsymbol{\Sigma}_l = \tilde{\boldsymbol{\Sigma}}, \ \forall l \in [L-1]. \tag{40}$$

According to Lemma 3, we have $\sigma_K(\boldsymbol{W}_l) \geq \left(\sqrt{n/2}\right)^{1/L} \geq \delta$, where the last inequality follows from (27) and (28). It follows from this and (24) that $\mathcal{A} \subseteq \{K+1, \ldots, d\}$. For simplicity, we write $\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{\Sigma}}_2 \end{bmatrix}$ satisfying $\tilde{\boldsymbol{\Sigma}}_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_K)$ and $\tilde{\boldsymbol{\Sigma}}_2 = \operatorname{diag}(\sigma_{K+1}, \ldots, \sigma_d)$. Using (32) and (24), we have

$$\sum_{i=K+1}^{d} \sigma_i^4 \leq \delta^2 = \varepsilon^4 (d-K).$$

Using (24) and letting $\mathcal{A}^c = \{K+1, \ldots, d\} \setminus \mathcal{A}$, we have

$$\sum_{i \in \mathcal{A}^c} \sigma_i^4 + (d - 2K)\varepsilon^4 \leq \varepsilon^4 (d-K),$$

which implies $\sigma_i \leq \sqrt[4]{K}\varepsilon$ for all $i \in \mathcal{A}^c$. Therefore, we have

$$\sigma_i \leq \sqrt[4]{K}\varepsilon, \ i = K+1, \ldots, d. \tag{41}$$

It follows from (31) in Lemma 4 that

$$\boldsymbol{W}_l \ldots \boldsymbol{W}_1 = \boldsymbol{U}_l \tilde{\boldsymbol{\Sigma}}^l \boldsymbol{V}_1^T, \ \forall l \in [L-1]. \tag{42}$$

It follows from (22), (37), and $\boldsymbol{Y} = \boldsymbol{I}_K \otimes \boldsymbol{1}_n^T$ that

$$\begin{aligned} \boldsymbol{W}_{L:1} \boldsymbol{\Delta}_W = \boldsymbol{W}_{L:1} \begin{bmatrix} \boldsymbol{x}_{1,1} & \ldots & \boldsymbol{x}_{1,n} & \ldots & \boldsymbol{x}_{K,1} & \ldots & \boldsymbol{x}_{K,n} \end{bmatrix} \\ - \boldsymbol{W}_{L:1} \begin{bmatrix} \bar{\boldsymbol{x}}_1 & \ldots & \bar{\boldsymbol{x}}_1 & \ldots & \bar{\boldsymbol{x}}_K & \ldots & \bar{\boldsymbol{x}}_K \end{bmatrix} = \boldsymbol{0}. \end{aligned} \tag{43}$$

Substituting (25), (31), and (40) into (43) yields

$$\mathbf{0} = \boldsymbol{W}_L \boldsymbol{U}_{L-1} \tilde{\boldsymbol{\Sigma}}^{L-1} \boldsymbol{V}_1^T \boldsymbol{\Delta}_W = \boldsymbol{W}_L \left(\boldsymbol{U}_{L-1,1} \tilde{\boldsymbol{\Sigma}}_1^{L-1} \boldsymbol{V}_{1,1}^T + \boldsymbol{U}_{L-1,2} \tilde{\boldsymbol{\Sigma}}_2^{L-1} \boldsymbol{V}_{1,2}^T \right) \boldsymbol{\Delta}_W.$$

As a result, we obtain for all $l \in [L-1]$,

$$\begin{aligned}
\|\boldsymbol{W}_L \boldsymbol{U}_{L-1,1} \tilde{\boldsymbol{\Sigma}}_1^{L-1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F &= \|\boldsymbol{U}_L \boldsymbol{\Sigma}_{L,1} \boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,2} \tilde{\boldsymbol{\Sigma}}_2^{L-1} \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F \\
&\le \|\boldsymbol{\Sigma}_{L,1}\| \|\boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,2}\| \|\tilde{\boldsymbol{\Sigma}}_2^{L-1-l}\| \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F \\
&\le (2n)^{1/2L} \frac{2\sqrt{\delta} \sqrt[4]{K}}{n^{1/2L}} \left(\sqrt[4]{K} \varepsilon \right)^{L-1-l} \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F \\
&\le 2\sqrt{2} K^{(L-l)/4} \sqrt[4]{d-K} \varepsilon^{L-l} \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F, \qquad (44)
\end{aligned}$$

where the second inequality follows from (29), (32), and (41), and the last inequality uses (27). Next, we compute

$$\begin{aligned}
\sigma_{\min}(\boldsymbol{W}_L \boldsymbol{U}_{L-1,1}) &= \sigma_{\min}(\boldsymbol{U}_L \boldsymbol{\Sigma}_{L,1} \boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,1}) = \sigma_{\min}(\boldsymbol{\Sigma}_{L,1} \boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,1}) \\
&\ge \sigma_{\min}(\boldsymbol{V}_{L,1}^T \boldsymbol{U}_{L-1,1}) \sigma_{\min}(\boldsymbol{\Sigma}_{L,1}) \ge \left(1 - \frac{2\sqrt{\delta} \sqrt[4]{K}}{n^{1/2L}} \right) \left(\frac{n}{2} \right)^{1/2L} \\
&\ge \frac{1}{2} \left(\frac{n}{2} \right)^{1/2L}, \qquad (45)
\end{aligned}$$

where the second inequality follows from (29) and (32), and the last inequality uses (27) and (28). Then, we compute

$$\begin{aligned}
\|\boldsymbol{W}_L \boldsymbol{U}_{L-1,1} \tilde{\boldsymbol{\Sigma}}_1^{L-1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F &\ge \sigma_{\min}(\boldsymbol{W}_L \boldsymbol{U}_{L-1,1} \tilde{\boldsymbol{\Sigma}}_1^{L-l-2}) \|\tilde{\boldsymbol{\Sigma}}_1^{l+1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F \\
&\ge \sigma_{\min}(\boldsymbol{W}_L \boldsymbol{U}_{L-1,1}) \sigma_{\min}(\tilde{\boldsymbol{\Sigma}}_1^{L-l-2}) \|\tilde{\boldsymbol{\Sigma}}_1^{l+1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F \\
&\ge \frac{1}{2} \left(\frac{n}{2} \right)^{(L-l-1)/2L} \|\tilde{\boldsymbol{\Sigma}}_1^{l+1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F, \qquad (46)
\end{aligned}$$

where the last inequality follows from (29) and (45). This, together with (44), implies for $l \in [L-2]$,

$$\frac{\|\tilde{\boldsymbol{\Sigma}}_1^{l+1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F}{\|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F} \le \kappa_1 \varepsilon^{L-l}, \text{ where } \kappa_1 := \frac{4\sqrt{2} K^{(L-l)/4} \sqrt[4]{d-K}}{(n/2)^{(L-l-1)/2L}}. \qquad (47)$$

It follows from (25), (40), and (42) that for all $l \in [L-1]$,

$$\|\boldsymbol{W}_{l:1} \boldsymbol{\Delta}_W\|_F^2 = \|\boldsymbol{U}_l \tilde{\boldsymbol{\Sigma}}^l \boldsymbol{V}_1^T \boldsymbol{\Delta}_W\|_F^2 = \|\tilde{\boldsymbol{\Sigma}}_1^l \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F^2.$$

This, together with (41), yields

$$\begin{aligned}
\frac{\|\boldsymbol{W}_{l+1:1} \boldsymbol{\Delta}_W\|_F^2}{\|\boldsymbol{W}_{l:1} \boldsymbol{\Delta}_W\|_F^2} &= \frac{\|\tilde{\boldsymbol{\Sigma}}_1^{l+1} \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^{l+1} \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F^2}{\|\tilde{\boldsymbol{\Sigma}}_1^l \boldsymbol{V}_{1,1}^T \boldsymbol{\Delta}_W\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F^2} \\
&\le \frac{\kappa_1 \varepsilon^{L-l} \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F + \sqrt{K} \varepsilon^2 \|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F^2}{\|\tilde{\boldsymbol{\Sigma}}_2^l \boldsymbol{V}_{1,2}^T \boldsymbol{\Delta}_W\|_F^2} \\
&\le \kappa_1 \varepsilon^{L-l} + \sqrt{K} \varepsilon^2 \le (\sqrt{K} + 1) \varepsilon^2, \qquad (48)
\end{aligned}$$

where the first inequality follows from (41) and (47). Suppose that we have

$$\|\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F \le \|\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F. \tag{49}$$

Then, we have

$$\|\tilde{\boldsymbol{\Sigma}}_2^{l+1}\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F \le \|\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F \le \|\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F \le \|\tilde{\boldsymbol{\Sigma}}_1^{l+1}\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F,$$

where the first inequality follows from $\sigma_i \le 1$ for all $i = K+1, \ldots, d$ due to (41) and (28), the second inequality uses (49), and the last inequality holds by (29). Using (42) and a similar argument as above, we compute

$$\begin{aligned}
\frac{\|\boldsymbol{W}_{l:1}\boldsymbol{\Delta}_B\|_F^2}{\|\boldsymbol{W}_{l+1:1}\boldsymbol{\Delta}_B\|_F^2} &= \frac{\|\tilde{\boldsymbol{\Sigma}}_1^l\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^l\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F^2}{\|\tilde{\boldsymbol{\Sigma}}_1^{l+1}\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^{l+1}\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F^2} \\
&\le \frac{\|\tilde{\boldsymbol{\Sigma}}_1^l\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2 + \|\tilde{\boldsymbol{\Sigma}}_2^l\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F^2}{\|\tilde{\boldsymbol{\Sigma}}_1^{l+1}\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2} \le \frac{2\|\tilde{\boldsymbol{\Sigma}}_1^l\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2}{\|\tilde{\boldsymbol{\Sigma}}_1^{l+1}\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2} \\
&\le \frac{2\|\tilde{\boldsymbol{\Sigma}}_1^l\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2}{\sigma_{\min}(\tilde{\boldsymbol{\Sigma}})\|\tilde{\boldsymbol{\Sigma}}_1^l\boldsymbol{V}_{1,1}^T\boldsymbol{\Delta}_B\|_F^2} \le 2,
\end{aligned}$$

where the last inequality uses $\sigma_{\min}(\tilde{\boldsymbol{\Sigma}}) \ge 1$ due to (29). This, together with (39) and (48), implies (15).

The rest of the proof is devoted to showing (49). For ease of exposition, let $\boldsymbol{V} := \boldsymbol{X}\boldsymbol{Y}^T/\sqrt{n} \in \mathbb{R}^{d\times K}$. One can verify $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_K$. In addition, we can compute

$$\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{\Delta}_B = \boldsymbol{\Delta}_B, \tag{50}$$

where the equality follows from (38) and $\boldsymbol{X} \in \mathcal{O}^d$. Using (22) and $\boldsymbol{X} \in \mathcal{O}^d$, we have

$$\boldsymbol{W}_{L:1} = \boldsymbol{Y}\boldsymbol{X}^T = \sqrt{n}\boldsymbol{V}^T. \tag{51}$$

This, together with (26) and (42), yields

$$\boldsymbol{U}_L\boldsymbol{\Sigma}_{L,1}\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1}\tilde{\boldsymbol{\Sigma}}^{L-1}\boldsymbol{V}_1^T = \sqrt{n}\boldsymbol{V}^T.$$

Therefore, we obtain

$$\begin{aligned}
n\boldsymbol{V}\boldsymbol{V}^T &= \boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{L-1}\boldsymbol{U}_{L-1}^T\boldsymbol{V}_{L,1}\boldsymbol{\Sigma}_{L,1}^2\boldsymbol{V}_{L,1}^T\boldsymbol{U}_{L-1}\tilde{\boldsymbol{\Sigma}}^{L-1}\boldsymbol{V}_1^T \\
&= \left(\boldsymbol{V}_{1,1}\tilde{\boldsymbol{\Sigma}}_1^{L-1}\boldsymbol{U}_{L-1,1}^T + \boldsymbol{V}_{1,2}\tilde{\boldsymbol{\Sigma}}_2^{L-1}\boldsymbol{U}_{L-1,2}^T\right)\boldsymbol{V}_{L,1}\boldsymbol{\Sigma}_{L,1}^2\boldsymbol{V}_{L,1}^T\left(\boldsymbol{U}_{L-1,1}\tilde{\boldsymbol{\Sigma}}_1^{L-1}\boldsymbol{V}_{1,1}^T + \boldsymbol{U}_{L-1,2}\tilde{\boldsymbol{\Sigma}}_2^{L-1}\boldsymbol{V}_{1,2}^T\right).
\end{aligned}$$

Using (32) and Davis-Kahan Theorem [77, Theorem V.3.6], we have

$$\|\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{V}_{1,1}\boldsymbol{V}_{1,1}^T\|_F \le \frac{1}{2}. \tag{52}$$

Then, we compute

$$\begin{aligned}
\|\boldsymbol{V}_{1,2}^T\boldsymbol{\Delta}_B\|_F &= \|(\boldsymbol{I} - \boldsymbol{V}_{1,1}\boldsymbol{V}_{1,1}^T)\boldsymbol{\Delta}_B\|_F = \|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^T)\boldsymbol{\Delta}_B + (\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{V}_{1,1}\boldsymbol{V}_{1,1}^T)\boldsymbol{\Delta}_B\|_F \\
&= \|(\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{V}_{1,1}\boldsymbol{V}_{1,1}^T)\boldsymbol{\Delta}_B\|_F \le \frac{1}{2}\|\boldsymbol{\Delta}_B\|_F.
\end{aligned}$$

where the third equality uses (50), and the inequality follows from (52). This, together with $\boldsymbol{V}_1 \in \mathcal{O}^d$, directly implies (49). □

D Auxiliary Results

Lemma 5. *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ of rank $r \geq 0$, we have*

$$\|\boldsymbol{A}^T\boldsymbol{A}\|_F \leq \|\boldsymbol{A}\|_F^2 \leq \sqrt{r}\|\boldsymbol{A}^T\boldsymbol{A}\|_F. \tag{53}$$

Proof. Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be a singular value decomposition of $\boldsymbol{A}$, where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \ \tilde{\boldsymbol{\Sigma}} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r),$$

$\boldsymbol{U} \in \boldsymbol{O}^m$, and $\boldsymbol{V} \in \boldsymbol{O}^n$. Then we compute

$$\|\boldsymbol{A}^T\boldsymbol{A}\|_F^2 = \|\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\|_F^2 = \sum_{i=1}^r \sigma_i^4, \ \|\boldsymbol{A}\|_F^2 = \|\boldsymbol{\Sigma}\|_F^2 = \sum_{i=1}^r \sigma_i^2,$$

which, together with the AM-QM inequality, directly implies (53). □