The Impact of Anisotropic Covariance Structure on the Training Dynamics and Generalization Error of Linear Networks

Anonymous authors
Paper under double-blind review

Abstract

The success of deep neural networks largely depends on the statistical structure of the training data. While learning dynamics and generalization on isotropic data are well-established, the impact of pronounced anisotropy on these crucial aspects is not yet fully understood. We examine the impact of data anisotropy, represented by a spiked covariance structure, a canonical yet tractable model, on the learning dynamics and generalization error of a two-layer linear network in a linear regression setting. Our analysis reveals that the learning dynamics proceed in two distinct phases, governed initially by the input-output correlation and subsequently by other principal directions of the data structure. Furthermore, we derive an analytical expression for the generalization error, quantifying how the alignment of the spike structure of the data with the learning task improves performance. Our findings offer deep theoretical insights into how data anisotropy shapes the learning trajectory and final performance, providing a foundation for understanding complex interactions in more advanced network architectures.

1 Introduction

Deep learning has achieved remarkable success in diverse fields, ranging from image recognition to natural language processing, over the past decade. A key factor contributing to the high performance of deep learning is the availability of vast amounts of training data. While deep networks perform well at learning complex patterns from large data volumes, their performance is inherently sensitive to the underlying statistical structure of the data. Therefore, understanding how the data structure influences learning dynamics and generalization performance remains a fundamental problem in deep learning research.

The inherent non-linearity of deep networks makes a theoretical analysis of their learning process difficult. Consequently, simplified models, such as linear networks, are often employed. Although linear networks are less expressive than non-linear networks, they provide a tractable framework for analyzing important aspects of deep learning, including convergence (Arora et al., 2018; Du & Hu, 2019), learning dynamics (Saxe et al., 2013; 2019; Arora et al., 2018; Tarmoun et al., 2021; Min et al., 2021; Atanasov et al., 2022; Braun et al., 2022; Dominé et al., 2024; Kunin et al., 2024), and implicit bias of gradient descent (Arora et al., 2019; Woodworth et al., 2020; Pesme et al., 2021; Kunin et al., 2023; Varre et al., 2023).

The learning dynamics of linear networks are well-understood when the input data is isotropic. Under this condition, we can derive exact expressions for learning trajectories under certain initial conditions. However, real-world data is rarely isotropic, as many datasets exhibit complex anisotropic structures. Extending these theories to anisotropic data settings is notably challenging, even for these simple linear models (Tarmoun et al., 2021). Thus, a deeper theoretical understanding of how data anisotropy shapes learning trajectories and affects generalization performance is crucial.

In this study, we investigate the impact of data anisotropy on the learning dynamics and generalization performance of linear networks. We consider a linear regression problem where the input data is generated from a spiked covariance model. This model introduces controlled anisotropy by featuring one large eigenvalue

(the spike) and small remaining eigenvalues. Specifically, we examine how spike magnitude and spike-target alignment affect the learning dynamics and generalization.

Our approach yields several important results. First, we identify distinct learning phases determined by the anisotropic data structure. To analyze the weight dynamics in the learning phases, we introduce a basis that captures the anisotropic structure of the input data. By effectively reducing the dimensionality of the dynamics, this framework allows us to simplify the description of the dynamics and consistently explain the overall weight evolution. Second, we derive an analytical expression for the generalization error of linear models trained on spiked covariance data. These findings provide a deeper theoretical understanding of learning under anisotropic data, establishing a crucial perspective for analyzing the behavior of more complex neural network architectures.

Our contributions are as follows:

- We demonstrate that two-layer linear networks trained on spiked covariance data exhibit distinct two-phase learning dynamics. We show that data anisotropy is essential for the existence of the later phase, as in its absence the dynamics reduce to the behavior in isotropic settings analyzed in previous studies (Section 3).
- We develop an analytical framework that reduces the high-dimensional dynamics to a tractable system of scalar differential equations. This approach clearly identifies the mechanisms governing weight evolution and the existence of the two learning phases (Section 3.2).
- We derive an analytical expression for the generalization error, explicitly quantifying its dependence on data anisotropy, namely the magnitude and alignment of the spike with the target vector (Section 4).

1.1 Related Work

While a linear network is representationally equivalent to a single-layer model, its multi-layer structure introduces non-linear learning dynamics. Consequently, linear networks have been widely used to understand the learning dynamics of deep neural networks. Numerous important insights have been obtained for linear networks, including their loss landscape (Baldi & Hornik, 1989; Fukumizu, 1998), convergence properties (Arora et al., 2018; Du & Hu, 2019), generalization capabilities (Lampinen & Ganguli, 2018; Poggio et al., 2018; Advani et al., 2020; Huh, 2020; Gunasekar et al., 2018), and the implicit bias of gradient descent (Arora et al., 2019; Woodworth et al., 2020; Pesme et al., 2021; Kunin et al., 2023; Varre et al., 2023).

The analytical tractability of linear network dynamics is significantly enhanced when the input data is whitened. Under this assumption, exact solutions for the learning dynamics have been derived in several cases (Fukumizu, 1998; Saxe et al., 2013; 2019; Arora et al., 2018; Tarmoun et al., 2021; Min et al., 2021; Atanasov et al., 2022; Braun et al., 2022; Dominé et al., 2024; Kunin et al., 2024). For instance, Saxe et al. (2013; 2019) demonstrated that for deep linear networks with small or specific spectral initialization, the learning process decouples into independent modes governed by the singular values of the weight matrices. In another analysis with specific balanced initialization, the dynamics can be described by matrix Riccati equations, which have closed-form solutions (Fukumizu, 1998; Braun et al., 2022; Tarmoun et al., 2021; Dominé et al., 2024).

In anisotropic settings, the symmetries that simplify analysis in the isotropic case, such as the rotational invariance of the input data distribution, are absent. This absence generally makes the derivation of exact closed-form solutions for weight evolution intractable, even for linear networks. For example, Tarmoun et al. (2021) suggest that obtaining exact dynamics for general anisotropic data is broadly unachievable. Consequently, research often turns to alternative approaches, such as perturbation analysis (Gidel et al., 2019) and fixed-point analysis (Zhang et al., 2024).

To describe the learning dynamics of linear networks under general data in more detail, Advani et al. (2020) provided a set of dynamical equations for multi-layer linear networks under small initialization, linking their behavior to that of linear networks in which the weight matrix has a rank-one structure. This observation has

been used to analyze various phenomena in linear networks (Shan & Bordelon, 2021; Ji & Telgarsky, 2018; Marion & Chizat, 2024). However, their contribution centered on deriving these reduced equations rather than providing explicit solutions for general anisotropic data. More recently, Atanasov et al. (2022) made significant progress by showing that under small initialization, linear networks initially align their weights with the input-output correlation direction and grow along it. However, their analysis primarily captured this early alignment phase, leaving the subsequent learning process largely unexplored. This paper aims to address these remaining open questions, investigating how components of the solution are learned over time under a spiked covariance model and what mechanisms govern the later learning phases.

Beyond training dynamics, understanding the generalization performance of neural networks, especially in overparameterized regimes, is a crucial challenge. Gradient-based optimization in such models often exhibits an implicit bias towards solutions with specific properties, such as minimum-norm solutions in linear models with small initialization (Gunasekar et al., 2017; 2018; Arora et al., 2019; Min et al., 2021; Yun et al., 2021). The generalization error of these solutions, particularly in high-dimensional settings, has been intensely studied, with random matrix theory proving to be a powerful tool for precise characterization (Bartlett et al., 2020; Belkin et al., 2020; Mei & Montanari, 2022; Atanasov et al., 2024). Random matrix theory has yielded key insights, enabling exact calculations of generalization error for models like ridgeless and kernel regression, and elucidating dependencies on dimensionality, sample size, and noise, including the "double descent" phenomenon (Adlam & Pennington, 2020; Hastie et al., 2022; Canatar et al., 2021). These advancements allow for a more detailed understanding of how data anisotropies influence generalization (Wu & Xu, 2020; Richards et al., 2021; Mel & Ganguli, 2021; Mel & Pennington, 2021; Wei et al., 2022). Utilizing these theoretical foundations, our work provides a precise analysis of the generalization error for the spiked covariance model. We derive an analytical expression that explicitly quantifies the dependence of the error on two key aspects of the anisotropy, namely, the magnitude and alignment of the spike with the target vector.

2 Preliminaries

In this study, we investigate a standard high-dimensional linear regression problem with structured data, using linear networks trained via gradient descent.

2.1 Generative model

We consider a dataset of n input-output pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The input data $\boldsymbol{x}_i$ is drawn from a multivariate Gaussian distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ given by

$$\boldsymbol{\Sigma} = \sigma^2(\boldsymbol{I} + \rho\boldsymbol{\mu}\boldsymbol{\mu}^\top), \tag{1}$$

where $\|\boldsymbol{\mu}\| = 1$. The covariance matrix $\boldsymbol{\Sigma}$ has one eigenvalue $\sigma^2(1+\rho)$ corresponding to the eigenvector $\boldsymbol{\mu}$ (the spike direction), and $d-1$ eigenvalues equal to σ^2 in the subspace orthogonal to $\boldsymbol{\mu}$. This model is known as the spiked covariance model (Johnstone, 2001). The output $y_i \in \mathbb{R}$ is given by

$$y_i = \boldsymbol{\beta}^\top \boldsymbol{x}_i, \tag{2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is the target vector with $\|\boldsymbol{\beta}\| = 1$. We define the spike-target alignment as $A = \boldsymbol{\mu}^\top\boldsymbol{\beta}$. Without loss of generality, we can assume $A \geq 0$, as the sign of $\boldsymbol{\mu}$ can be flipped without changing the covariance matrix $\boldsymbol{\Sigma}$. Using this target vector, the input-output correlation is given by

$$\boldsymbol{\Sigma}_{xy} = \mathbb{E}[y\boldsymbol{x}] = \boldsymbol{\Sigma}\boldsymbol{\beta}. \tag{3}$$

In this paper, we investigate how the dynamics and generalization error of a linear network depend on the spike magnitude ρ and the spike-target alignment A.

2.2 Two-layer linear network

A two-layer linear network of width m is defined as

$$f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{W}\boldsymbol{x}. \tag{4}$$

Here $\boldsymbol{x} \in \mathbb{R}^d$ is the input, $\boldsymbol{W} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{a} \in \mathbb{R}^m$ are the weight parameters for the first and second layers, respectively. In particular, we consider the case where $m \geq d$.

For analytical tractability, the network is initialized with small random weights and is trained using full-batch gradient descent with learning rate η (or time constant $\tau = 1/2\eta$) to minimize the squared error on the training dataset:

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{a}) = \frac{1}{n} \sum_{i=1}^{n} \left(f(\boldsymbol{x}_i) - y_i \right)^2 . \tag{5}$$

In the limit of an infinitesimal learning rate, the learning dynamics of gradient descent are well approximated by the following gradient flow:

$$\tau \frac{d}{dt} \boldsymbol{W} = \boldsymbol{a} \left(\hat{\boldsymbol{\Sigma}}_{xy}^\top - \boldsymbol{a}^\top \boldsymbol{W} \hat{\boldsymbol{\Sigma}} \right), \tag{6}$$

$$\tau \frac{d}{dt} \boldsymbol{a}^\top = \left(\hat{\boldsymbol{\Sigma}}_{xy}^\top - \boldsymbol{a}^\top \boldsymbol{W} \hat{\boldsymbol{\Sigma}} \right) \boldsymbol{W}^\top, \tag{7}$$

where $\hat{\boldsymbol{\Sigma}}_{xy}$ denotes the empirical input-output correlation, and $\hat{\boldsymbol{\Sigma}}$ denotes the empirical input covariance:

$$\hat{\boldsymbol{\Sigma}}_{xy} = \frac{1}{n} \sum_{i=1}^{n} y_i \boldsymbol{x}_i, \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top . \tag{8}$$

3 Learning Dynamics in Two-Layer Linear Network under Spiked Covariance Model

3.1 Empirical Observation

We begin by empirically examining the learning dynamics of a two-layer linear network using numerical simulations. To capture the essential behavior of the weight parameters, we show the learning trajectory in a two-dimensional phase plane defined by the coordinates $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{\mu}$ and $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{\mu}_\perp$, where $\hat{\boldsymbol{r}}_1$ is a fixed vector and $\boldsymbol{\mu}_\perp$ denotes a vector orthogonal to $\boldsymbol{\mu}$. Figure 1 shows this trajectory, together with orthogonal lines whose directions are defined by the vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ (precise definitions of $\hat{\boldsymbol{r}}_1$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_2$ are provided in the subsequent subsection). The learning trajectory clearly exhibits two distinct phases. It initially evolves primarily along the direction of $\boldsymbol{v}_1$, then makes a sharp turn and develops along $\boldsymbol{v}_2$ until it converges to the final solution.

These two-phase learning dynamics are also evident in the time evolution of the training loss. In Figure 2, we plot the loss together with the temporal evolution of the $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ components of $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}$, namely, $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{v}_1$ and $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{v}_2$. Initially, the training loss remains on a plateau before dropping sharply. This drop corresponds to the rapid growth of the $\boldsymbol{v}_1$ component of $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}$, while the $\boldsymbol{v}_2$ component remains negligible. Subsequently, the loss decreases smoothly, driven by the growth of the $\boldsymbol{v}_2$ component of $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}$ and adjustments to the $\boldsymbol{v}_1$ component.

Based on this empirical observation, we define two distinct phases of learning: an *early phase*, characterized by the rapid growth of the $\boldsymbol{v}_1$ component of $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}$, and a *later phase*, characterized by the subsequent growth of its $\boldsymbol{v}_2$ component. In the following, we provide a theoretical analysis of these observed dynamics.

3.2 Theoretical Analysis

To analyze the two-phase behavior, we consider the limit $n \to \infty$ with a finite input dimension d, where the empirical covariance matrices $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Sigma}}_{xy}$ are replaced by their expectations, $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_{xy} = \boldsymbol{\Sigma}\boldsymbol{\beta}$, respectively. Additionally, we introduce a basis adapted to the data structure, designed to capture the principal axes of learning and thus allowing for a clear characterization of each phase. We construct this basis, denoted by the matrix $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d) \in \mathbb{R}^{d \times d}$, as follows:

- The first basis vector is defined as $\boldsymbol{v}_1 = \boldsymbol{\Sigma}_{xy} / \|\boldsymbol{\Sigma}_{xy}\|$, to capture the weight growth along the input-output correlation direction in the early phase.

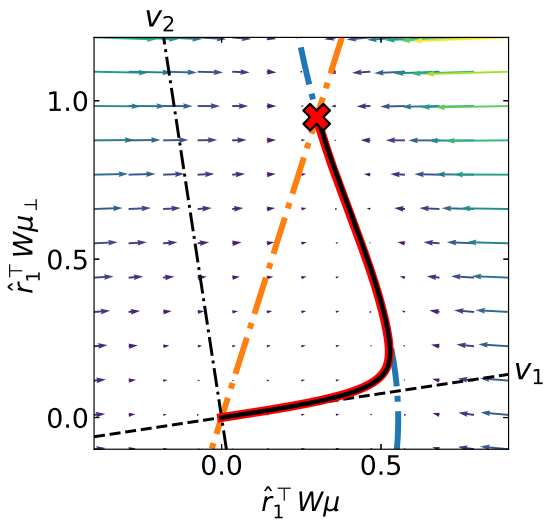


Figure 1: Learning trajectories in a phase plane with the spike direction $\boldsymbol{\mu}$ as the horizontal axis and the perpendicular direction $\boldsymbol{\mu}_\perp = (\boldsymbol{I} - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta}/\|(\boldsymbol{I} - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta}\|$ as the vertical axis. The vector field and nullclines (blue: $dw_1/dt = 0$, orange: $dw_2/dt = 0$) are from the reduced scalar dynamics (Equations (18) and (19)). The red trajectory is projected from a simulation, while the black trajectory is from the reduced scalar dynamics (Equations (18) and (19)). The basis directions, $\boldsymbol{v}_1$ (dashed) and $\boldsymbol{v}_2$ (dash-dotted), are also shown. The close match between trajectories validates the dimensionality reduction analysis. The parameters are set to $A = 0.3$, $\rho = 20$, and $\sigma^2 = 1$, with the number of training samples $n = 10000$. The network parameters are $d = 30$ and $m = 50$ and the initialization is performed as $W_{ij} \sim \mathcal{N}(0, s^2/d)$ and $a_i \sim \mathcal{N}(0, s^2/m)$ with $s = 10^{-5}$.

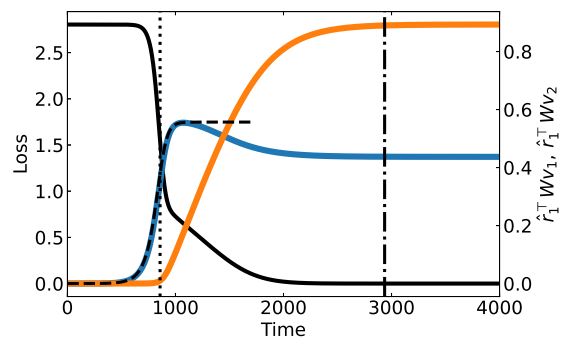


Figure 2: Time evolution of the training loss and magnitudes of weight components. The plotted values are the numerically obtained training loss (black) and the projections of the weight vectors $\boldsymbol{w}_1$ (blue) and $\boldsymbol{w}_2$ (orange) onto the initial growth direction $\hat{\boldsymbol{r}}_1$, namely, $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}\boldsymbol{v}_1$ and $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W}\boldsymbol{v}_2$, respectively. The black dashed line is the analytical approximation for the early phase, initialized by $u(0) = \hat{\boldsymbol{r}}_1^\top(\boldsymbol{a}(0)+\boldsymbol{w}_1(0))/2$ (Equation (17)). Vertical dotted lines and vertical dash-dotted lines indicate characteristic timescales for the early and later phases, respectively, with the latter defined in Equation (20) using $\delta = 3$. The parameters are the same as those in Figure 1.

- The second basis vector, $\boldsymbol{v}_2$, is defined based on the alignment of $\boldsymbol{\Sigma}_{xy}$ and $\boldsymbol{\beta}$. If they are not parallel, $\boldsymbol{v}_2$ is defined as the normalized vector in the plane spanned by $\{\boldsymbol{\Sigma}_{xy}, \boldsymbol{\beta}\}$ that is orthogonal to $\boldsymbol{v}_1$. In this case, it captures the adjustment of the weights toward $\boldsymbol{\beta}$. The sign of $\boldsymbol{v}_2$ will be specified just below for analytical convenience. In the parallel case, where no such adjustment occurs, $\boldsymbol{v}_2$ is simply chosen as any normalized vector orthogonal to $\boldsymbol{v}_1$ to complete the basis.

- The remaining vectors, $\{\boldsymbol{v}_3, \ldots, \boldsymbol{v}_d\}$, form an orthonormal basis for the subspace orthogonal to the span of $\{\boldsymbol{\Sigma}_{xy}, \boldsymbol{\beta}\}$.

Using the basis $\boldsymbol{V}$, we can represent the weight matrix as $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d)\boldsymbol{V}^\top$. By substituting this expression into the gradient flow equation (Equation (6)), we obtain the following dynamics for each

component vector:

$$\tau \frac{d}{dt}\boldsymbol{w}_1 = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{a} - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)\boldsymbol{a}, \tag{9}$$

$$\tau \frac{d}{dt}\boldsymbol{w}_2 = -(-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2)\boldsymbol{a}, \tag{10}$$

$$\tau \frac{d}{dt}\boldsymbol{w}_i = -\sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \boldsymbol{a}, \quad (i = 3, \ldots, d). \tag{11}$$

Similarly, from Equation (7), the dynamics for the second-layer weight $\boldsymbol{a}$ are given by:

$$\tau \frac{d}{dt}\boldsymbol{a} = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{w}_1 - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)\boldsymbol{w}_1 - (-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2)\boldsymbol{w}_2 - \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \boldsymbol{w}_i, \tag{12}$$

where $\lambda_1 = \boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_1$, $\lambda_2 = \boldsymbol{v}_2^\top \boldsymbol{\Sigma} \boldsymbol{v}_2$, and $\nu = -\boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_2$. Here, we choose the direction of $\boldsymbol{v}_2$ so that ν is positive. Explicit forms of $\|\boldsymbol{\Sigma}_{xy}\|$, λ_1, λ_2, and ν are given in Appendix A.

3.2.1 Early Phase

In the early phase of learning, for a two-layer linear network initialized with small random weights drawn from a distribution with a small standard deviation $s \ll 1$, its dynamics can be approximated by linear differential equations by neglecting higher-order terms:

$$\tau \frac{d}{dt}\boldsymbol{W} = \boldsymbol{a}\boldsymbol{\Sigma}_{xy}^\top + O(s^3), \quad \tau \frac{d}{dt}\boldsymbol{a} = \boldsymbol{W}\boldsymbol{\Sigma}_{xy} + O(s^3). \tag{13}$$

Expressing $\boldsymbol{W}$ in the basis $\boldsymbol{V}$, we obtain:

$$\tau \frac{d}{dt}\boldsymbol{w}_1 = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{a} + O(s^3), \quad \tau \frac{d}{dt}\boldsymbol{a} = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{w}_1 + O(s^3) \tag{14}$$

while for $i = 2, \ldots, d$, $\tau d\boldsymbol{w}_i/dt = O(s^3)$. Since the basis vector is given by $\boldsymbol{v}_1 = \boldsymbol{\Sigma}_{xy}/\|\boldsymbol{\Sigma}_{xy}\|$, the first-layer weights $\boldsymbol{W}$ grow primarily along the direction of $\boldsymbol{v}_1$, while components in other directions $\boldsymbol{v}_i$ for $i \geq 2$ remain negligible at the initial scale $O(s)$. Therefore, Equation (14) for $\boldsymbol{w}_1$ and $\boldsymbol{a}$ can be solved with errors of $O(s)$ for $t \ll \frac{\tau}{2\|\boldsymbol{\Sigma}_{xy}\|} \log(\|\boldsymbol{\Sigma}_{xy}\|/(\lambda_1 s^2))$, as:

$$\boldsymbol{w}_1(t) = \boldsymbol{r}_1 e^{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau} t} + O(s), \quad \boldsymbol{a}(t) = \boldsymbol{r}_1 e^{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau} t} + O(s) \tag{15}$$

where $\boldsymbol{r}_1 = (\boldsymbol{w}_1(0) + \boldsymbol{a}(0))/2$. Atanasov et al. (2022) identified this alignment direction using a linearized multivariate system. In this formulation, however, the specific temporal evolution of the weight magnitude in anisotropic settings was not considered.

To understand how these weights grow and eventually saturate, we must return to the full non-linear dynamics (Equations (9) to (12)). The key insight for this analysis lies in a symmetry between the dynamics of $\boldsymbol{w}_1$ and $\boldsymbol{a}$ in the small initialization limit. While not apparent in the original gradient flow (Equations (6) and (7)), the decomposed equations reveal that the dynamics for $\boldsymbol{w}_1$ (Equation (9)) and $\boldsymbol{a}$ (Equation (12)) become symmetric under exchange, provided that the other negligible components $\boldsymbol{w}_2, \ldots, \boldsymbol{w}_d$ are ignored. This symmetry allows us to assume $\boldsymbol{a}(t) \approx \boldsymbol{w}_1(t) = u(t)\hat{\boldsymbol{r}}_1$, where $u(t)$ is a scalar magnitude and $\hat{\boldsymbol{r}}_1 = \boldsymbol{r}_1/\|\boldsymbol{r}_1\|$ is their common growth direction. Substituting this into the dynamics for $\boldsymbol{w}_1$ (Equation (9)) yields a self-contained scalar differential equation for $u(t)$:

$$\tau \frac{d}{dt}u(t) = \|\boldsymbol{\Sigma}_{xy}\|u(t) - \lambda_1 u(t)^3 + O(s). \tag{16}$$

The solution to this equation describes a sigmoidal growth for the magnitude of $\boldsymbol{w}_1$ (See Appendix B.1 for details):

$$\boldsymbol{w}_1(t) = \sqrt{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\left(1 - e^{-2\|\boldsymbol{\Sigma}_{xy}\|t/\tau} + \frac{\|\boldsymbol{\Sigma}_{xy}\|}{\lambda_1} u(0)^{-2} e^{-2\|\boldsymbol{\Sigma}_{xy}\|t/\tau}\right)\lambda_1}}\hat{\boldsymbol{r}}_1 + O(s). \tag{17}$$

The term $\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\lambda_1} u(0)^{-2} e^{-2\|\boldsymbol{\Sigma}_{xy}\|t/\tau}$ in the denominator initially dominates the expression due to the small initial value $u(0) = O(s)$. As the exponential factor decays, this term diminishes rapidly, causing $u(t)$ to transition sharply from its initial $O(s)$ scale to its $O(1)$ saturation value, approximately $\sqrt{\|\boldsymbol{\Sigma}_{xy}\|/\lambda_1}$. During this brief transition, the other components $\boldsymbol{w}_i(t)$ (for $i \geq 2$) remain at their initial $O(s)$ scale. This justifies the effective decoupling of the dynamics of $\boldsymbol{w}_1$ during the early phase.

Equation (17) reveals that, up to $O(s)$ corrections, the magnitude evolves according to the same sigmoidal law governing isotropic data. This implies that the anisotropy primarily determines the alignment direction, while the growth dynamics of the magnitude in this phase effectively follow the isotropic case along that direction.

This sigmoidal shape and its characteristic timescale, $\frac{\tau}{2\|\boldsymbol{\Sigma}_{xy}\|} \log(\|\boldsymbol{\Sigma}_{xy}\|/(\lambda_1 s^2))$, are visually confirmed in Figure 2. This implies that learning in this phase becomes faster as $\|\boldsymbol{\Sigma}_{xy}\| = \sigma^2\sqrt{1 + ((1+\rho)^2 - 1)A^2}$ increases, which occurs with a larger spike magnitude ρ and greater alignment A between the spike direction and the target vector.

3.2.2 Later Phase

Next, we consider the phase after $\boldsymbol{w}_1$ and $\boldsymbol{a}$ have reached $O(1)$ scale. In general, when the data is anisotropic and the target vector $\boldsymbol{\beta}$ and the spike direction $\boldsymbol{\mu}$ are neither perfectly aligned nor orthogonal, the network must modify its weights to capture the target using a component other than $\boldsymbol{w}_1$. We characterize this later phase by reducing the system to a lower-dimensional one.

To this end, we first focus on the dynamics of weight components $\boldsymbol{w}_i$ for $i \geq 3$. Equation (11) shows that the projection of each $\boldsymbol{w}_i$ onto the direction of $\boldsymbol{a}$ decays, while the components orthogonal to $\boldsymbol{a}$ do not grow. Since $\boldsymbol{w}_i(0)$ is of scale $O(s)$, these components remain at $O(s)$ throughout the training. Thus, the overall dynamics are effectively confined to the subspace spanned by $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{a}$.

We then consider the directions of these three vectors. At the beginning of this phase, $\boldsymbol{w}_1(t)$ and $\boldsymbol{a}(t)$ are already aligned with a common direction $\hat{\boldsymbol{r}}_1$ (up to $O(s)$ corrections), while $\boldsymbol{w}_2$ remains at the $O(s)$ scale. The evolution equations reveal a tight coupling among these vectors that preserves and reinforces this alignment. Specifically, the time derivatives of both $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are proportional to $\boldsymbol{a}$ (Equations (9) and (10)), indicating that they grow along the direction of $\boldsymbol{a}$. In turn, $\boldsymbol{a}$ evolves within the subspace spanned by $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ (Equation (12)). Given the initial alignment of $\boldsymbol{w}_1$ and $\boldsymbol{a}$, these reciprocal dynamics constrain all three vectors to evolve nearly along the single common direction $\hat{\boldsymbol{r}}_1$. This observation holds exactly in the limit $s \to 0$, as we formally prove in Appendix C.

Crucially, this alignment of the vectors simplifies the system's dynamics by allowing us to describe the evolution of these vectors using only their scalar magnitudes: $\boldsymbol{a}(t) \approx a(t)\hat{\boldsymbol{r}}_1$, $\boldsymbol{w}_1(t) \approx w_1(t)\hat{\boldsymbol{r}}_1$, and $\boldsymbol{w}_2(t) \approx w_2(t)\hat{\boldsymbol{r}}_1$. This effectively reduces the dimensionality of the problem and implies that the first-layer weight matrix can be approximated as $\boldsymbol{W} \approx \hat{\boldsymbol{r}}_1 \boldsymbol{w}(t)^\top \boldsymbol{V}^\top$, where $\boldsymbol{w}(t) = (w_1(t), w_2(t), 0, \ldots, 0)^\top$. This rank-one structure of the weight matrix is consistent with findings in previous research (Advani et al., 2020; Atanasov et al., 2022; Shan & Bordelon, 2021; Ji & Telgarsky, 2018; Marion & Chizat, 2024).

The description can be simplified further by employing a conservation law $\frac{d}{dt}(\boldsymbol{a}\boldsymbol{a}^\top - \sum_{i=1}^d \boldsymbol{w}_i \boldsymbol{w}_i^\top) = 0$, which can be derived from the fact that the time derivatives of $\boldsymbol{a}\boldsymbol{a}^\top$ and $\boldsymbol{W}\boldsymbol{W}^\top$ are identical under the gradient flow dynamics (Equations (6) and (7)). Considering the initialization, this implies $\boldsymbol{a}\boldsymbol{a}^\top - \sum_{i=1}^d \boldsymbol{w}_i \boldsymbol{w}_i^\top = O(s^2)$ for all time. Given that $\boldsymbol{w}_i$ for $i \geq 3$ remain at $O(s)$, this simplifies to $\boldsymbol{a}\boldsymbol{a}^\top - \boldsymbol{w}_1\boldsymbol{w}_1^\top - \boldsymbol{w}_2\boldsymbol{w}_2^\top = O(s^2)$. Taking the trace of this expression yields the important relation between the magnitudes $a(t)^2 \approx w_1(t)^2 + w_2(t)^2$. This allows us to express $a(t)$ in terms of $w_1(t)$ and $w_2(t)$, reducing the system to two variables.

By substituting these scalar representations into the vector dynamics of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ (Equations (9) and (10)), and then applying the conservation law through the relation $a(t)^2 \approx w_1(t)^2 + w_2(t)^2$, we reduce the dynamics

to the following system of two scalar differential equations:

$$\tau \frac{d}{dt} w_1 = \|\boldsymbol{\Sigma}_{xy}\| \sqrt{w_1^2 + w_2^2} - \lambda_1 w_1 (w_1^2 + w_2^2) + \nu w_2 (w_1^2 + w_2^2), \tag{18}$$

$$\tau \frac{d}{dt} w_2 = \nu w_1 (w_1^2 + w_2^2) - \lambda_2 w_2 (w_1^2 + w_2^2). \tag{19}$$

Figure 1 validates these reduced scalar dynamics by showing close agreement with the full network simulation.

The growth of the second component, w_2, is driven by the coupling term ν, which is non-zero only when the data is anisotropic ($\rho > 0$) and the spike direction is neither perfectly aligned nor orthogonal to the target vector ($0 < A < 1$). Conversely, if these conditions are not met (i.e., if $A = 0$, $A = 1$, or $\rho = 0$), the term ν vanishes, and as seen from Equation (19), $\boldsymbol{w}_2$ does not grow beyond its initial $O(s)$ magnitude. In such simpler cases, the learning dynamics are fully described by the early phase, a scenario that has been extensively studied (Saxe et al., 2013; Fukumizu, 1998; Atanasov et al., 2022).

It is noteworthy that these scalar equations become exact descriptions of the magnitude dynamics if the network is initialized such that components $\boldsymbol{w}_i(0)$ for $i \geq 3$ are zero and the vectors $\boldsymbol{a}(0)$, $\boldsymbol{w}_1(0)$, and $\boldsymbol{w}_2(0)$ are perfectly aligned. Under these conditions, the relation $a^2 = w_1^2 + w_2^2$ holds exactly.

While the coupled differential equations for w_1 and w_2 are analytically intractable, we can estimate the time required for w_2 to grow to a substantial fraction of its final value. We assume that $w_1(t) \geq w_{1*}$ and $w_2(t) \leq w_{2*}$ during the later phase, where $w_{1*} = \lim_{t\to\infty} w_1(t)$ and $w_{2*} = \lim_{t\to\infty} w_2(t)$. Let t_1 be the time when the later phase begins ($w_2(t_1) \approx 0$), and t_2 be the time when $w_2(t_2)$ reaches $(1 - e^{-\delta}) w_{2*}$. We can then establish the following upper bound on the time interval $t_2 - t_1$ in the $s \to 0$ limit (See Appendix B.2 for details):

$$t_2 - t_1 \leq \tau \frac{1}{\lambda_2} \left[\frac{\nu}{\lambda_2} \arctan\left(\frac{\nu}{\lambda_2} (1 - e^{-\delta}) \right) + \frac{1}{2} \log\left(1 + \left(\frac{\nu}{\lambda_2} (1 - e^{-\delta}) \right)^2 \right) + \delta \right]. \tag{20}$$

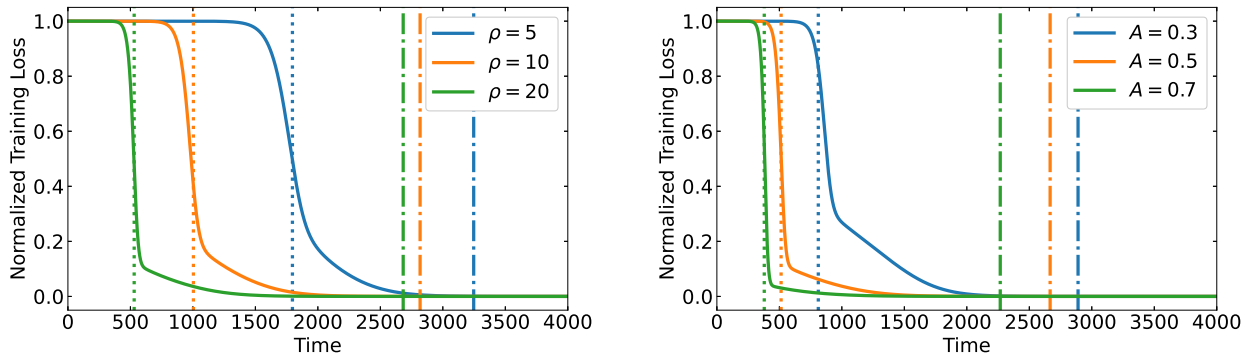
This bound is tight if w_1 changes little during the growth of w_2. The analysis of the timescale highlights that the speed of the later phase is governed by $1/\lambda_2$, representing the inverse of data variance along the $\boldsymbol{v}_2$ direction.

Figure 2 also shows the later-phase dynamics, as the projection $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{v}_2$ (corresponding to w_2 in the reduced dynamics) begins to grow after $\hat{\boldsymbol{r}}_1^\top \boldsymbol{W} \boldsymbol{v}_1$ (corresponding to w_1) has reached its peak. Furthermore, the characteristic timescale t_2 derived in Equation (20), marked by a vertical dash-dotted line, coincides well with the saturation of the w_2 component, validating the theoretical analysis.

The effects of spike magnitude ρ and spike-target alignment A on the training loss are summarized in Figure 3. As shown in the figure, the learning process indeed consists of two phases, with the acceleration in the early phase being the dominant factor shaping the overall training trajectory. This dominance of the early phase, which is more pronounced for a larger spike magnitude ρ and stronger spike-target alignment A, leads to a substantial initial loss reduction and keeps the training loss consistently lower throughout the process. Conversely, when these parameters are small, the loss decreases more slowly, and it becomes difficult to distinguish between the two phases. The theory developed here explains these observations by showing how anisotropic data structures, namely, spike magnitude ρ and spike-target alignment A, influence the two learning phases. A larger spike magnitude ρ primarily accelerates the early phase by increasing the magnitude of the input-output correlation $\|\boldsymbol{\Sigma}_{xy}\|$, while a stronger spike-target alignment A achieves this by slowing the convergence of the later phase through a reduction of λ_2, i.e., the data variance along the direction of $\boldsymbol{v}_2$.

4 Generalization Error under Spiked Covariance Model

We now investigate how the spiked covariance structure affects the generalization performance. While the behavior of generalization error is well-established for isotropic data, its precise dependence on the key parameters of an anisotropic structure, in particular the spike magnitude ρ and spike-target alignment A,



(a) Varying ρ while keeping $A = 0.5$ fixed.

(b) Varying A while keeping $\rho = 20$ fixed.

Figure 3: Evolution of the normalized training loss for different settings of (a) spike magnitude ρ and (b) spike-target alignment A. The loss $\mathcal{L}$ is normalized by the empirical second moment of the labels, $\frac{1}{n}\sum_{i=1}^{n} y_i^2$. Vertical dotted and dash-dotted lines indicate the characteristic timescales for the early and later learning phases, respectively, with the latter defined in Equation (20) using $\delta = 3$. Other parameters are the same as in Figure 1.

has not been fully characterized. This section provides a quantitative analysis of this relationship by deriving a precise analytical expression for the generalization error. Our result explicitly demonstrates this connection, offering a clear theoretical explanation for how these structural properties of data influence generalization.

We focus on the high-dimensional regime where the number of samples n and the data dimension d are large, specifically when $n, d \to \infty$ such that their ratio $\gamma = d/n > 1$. Under this condition, the regression problem admits infinitely many solutions that achieve the optimal training loss. It is known that for high-dimensional regression, linear networks trained with gradient flow from a small initialization converge to a solution close to the minimum ℓ_2-norm solution given by:

$$\hat{\boldsymbol{\beta}}_0 = \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top)^+ \boldsymbol{y}, \tag{21}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is the design matrix whose rows are the input data $\boldsymbol{x}_i$, and $\boldsymbol{y} \in \mathbb{R}^n$ is the vector of corresponding outputs y_i (Min et al., 2021; Yun et al., 2021).

This solution is also identical to that obtained by ridgeless regression, which corresponds to taking the limit $\lambda \to 0^+$ in ridge regression:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_\lambda &= \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i)^2 + \lambda\|\boldsymbol{\beta}\|^2 \\ &= (\boldsymbol{X}^\top\boldsymbol{X} + \lambda n \boldsymbol{I})^{-1}\boldsymbol{X}^\top \boldsymbol{y}. \end{aligned} \tag{22}$$

Therefore, the generalization error of the solution $R_0 = \mathbb{E}[(\hat{\boldsymbol{\beta}}_0^\top \boldsymbol{x} - \boldsymbol{\beta}^\top \boldsymbol{x})^2]$ can be obtained as the limit $\lambda \to 0^+$ of the ridge regression error

$$R_\lambda = \mathbb{E}[(\hat{\boldsymbol{\beta}}_\lambda^\top \boldsymbol{x} - \boldsymbol{\beta}^\top \boldsymbol{x})^2],$$

which converges to

$$\hat{R}_\lambda = \frac{\partial \kappa(\lambda)}{\partial \lambda}\kappa(\lambda)^2 \sum_{i=1}^{d} \frac{\lambda_i}{(\kappa(\lambda)+\lambda_i)^2}(\boldsymbol{\beta}^\top \boldsymbol{u}_i)^2, \tag{23}$$

in the limit $d, n \to \infty$ with $d/n \to \gamma$, as shown in recent advances in random matrix theory (Wei et al., 2022; Hastie et al., 2022; Canatar et al., 2021; Wu & Xu, 2020; Richards et al., 2021; Mel & Ganguli, 2021; Atanasov et al., 2024). Here, $\boldsymbol{u}_i$ and λ_i are the eigenvectors and eigenvalues of the covariance matrix of the input data, $\boldsymbol{\Sigma} = \sum_{i=1}^{d} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$, and $\kappa(\lambda)$ is the unique positive solution of:

$$1 = \frac{\lambda}{\kappa(\lambda)} + \frac{1}{n}\sum_{i=1}^{d}\frac{\lambda_i}{\kappa(\lambda)+\lambda_i}. \tag{24}$$

For our spiked covariance model, the principal eigenvalue corresponding to $\boldsymbol{u}_1 = \boldsymbol{\mu}$ is $\lambda_1 = \sigma^2(1+\rho)$, and the remaining eigenvalues are $\lambda_i = \sigma^2$ for $i = 2, \ldots, d$. Hence, the empirical spectral distribution of $\boldsymbol{\Sigma}$ is $\frac{1}{d}\delta_{\sigma^2(1+\rho)} + \frac{d-1}{d}\delta_{\sigma^2}$. Consequently, in the limit $n, d \to \infty$ with $d/n = \gamma$, the contribution of the single spike to the sum $\frac{1}{n}\sum_{i=1}^{d} \frac{\lambda_i}{\kappa(\lambda)+\lambda_i} = \frac{\gamma}{d}\sum_{i=1}^{d} \frac{\lambda_i}{\kappa(\lambda)+\lambda_i}$ vanishes, because its weight in the empirical spectral distribution becomes negligible as $d \to \infty$. This sum is thus dominated by the $d-1$ isotropic eigenvalues σ^2, and the equation for $\kappa(\lambda)$ reduces to:

$$1 = \frac{\lambda}{\kappa(\lambda)} + \gamma \frac{\sigma^2}{\kappa(\lambda) + \sigma^2}. \tag{25}$$

Solving this for $\kappa(\lambda)$ and its derivative with respect to λ, we obtain:

$$\lim_{\lambda \to 0^+} \kappa(\lambda) = \sigma^2(\gamma - 1) \quad \text{and} \quad \lim_{\lambda \to 0^+} \frac{\partial \kappa(\lambda)}{\partial \lambda} = \frac{\gamma}{\gamma - 1}. \tag{26}$$

Now, since $A = \boldsymbol{\mu}^\top \boldsymbol{\beta}$, we can decompose $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = A\boldsymbol{\mu} + \sqrt{1 - A^2}\boldsymbol{u}_2$, where $\boldsymbol{u}_2 = (\boldsymbol{I} - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta} / \|(\boldsymbol{I} - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta}\|$ is one of the eigenvectors of $\boldsymbol{\Sigma}$.

Substituting this decomposition, along with Equation (26), into Equation (23) gives the generalization error,

$$\hat{R}_0 = \sigma^2 \left(1 - \frac{1}{\gamma}\right) \left[1 - A^2 \left(1 - \frac{1+\kappa}{(1+\kappa/\gamma)^2}\right)\right]. \tag{27}$$

Note that this expression with $\rho = 0$ recovers previous results for isotropic data (Mei & Montanari, 2022; Belkin et al., 2020). To better assess the performance of the model relative to the signal strength, we evaluate the normalized generalization error, which is the error $\hat{R}_0$ divided by the variance of the true signal, $\mathbb{E}[y^2] = \boldsymbol{\beta}^\top \boldsymbol{\Sigma} \boldsymbol{\beta} = \sigma^2(1 + \rho A^2)$. Our analysis predicts that this normalized error $\hat{R}_0 / \mathbb{E}[y^2]$ is minimized when the spike magnitude ρ is large and spike-target alignment A is strong, indicating that a pronounced anisotropy of the data structure significantly improves generalization if the anisotropy, represented by the spike, is suitably aligned with the direction of the target vector of the task. Conversely, when the target signal is orthogonal to the spike ($A = 0$), the spike provides no benefit to generalization performance. This overall theoretical dependence of the normalized error on both spike magnitude and spike-target alignment is shown in Figure 4.

To validate these theoretical predictions, we conducted numerical simulations of a two-layer linear network with small initialization, illustrated in Figure 5. The results show a close match between the empirical normalized generalization errors and the theoretical predictions across a range of parameters.

5 Discussion and Conclusion

In this work, we investigated the learning dynamics of two-layer linear networks trained on data with a spiked covariance structure and analyzed the resulting generalization error. Theoretical analysis revealed several key insights into how data anisotropy shapes the learning process and generalization.

First, we characterized the learning dynamics, revealing two distinct phases. The early phase involves rapid weight growth in the direction of the input-output correlation $\boldsymbol{v}_1$. The speed of this phase, proportional to $\|\boldsymbol{\Sigma}_{xy}\|$, increases with both a larger spike magnitude and a stronger spike-target alignment. Once this component saturates, the later phase begins. During this phase, learning along the direction $\boldsymbol{v}_2$, which is defined within the span of $\{\boldsymbol{\Sigma}_{xy}, \boldsymbol{\beta}\}$ and orthogonal to $\boldsymbol{v}_1$, becomes significant. Its growth rate is governed by $\lambda_2 = \boldsymbol{v}_2^\top \boldsymbol{\Sigma} \boldsymbol{v}_2$ under the condition that the component in the $\boldsymbol{v}_1$ direction decreases during the later phase. The analysis of the dynamics in this data-adapted basis reveals a mechanism behind these two phases. By decomposing weight evolution along these axes, we obtain a reduced-dimensionality model that clearly explains how the network learns features in phases.

Second, we derived an analytical expression for the generalization error in the high-dimensional regime where $d, n \to \infty$ with $d/n = \gamma > 1$. The expression (Equation (27)) quantifies the dependence of the error on the spike magnitude and the spike-target alignment. Our analysis shows that a larger, well-aligned spike leads to lower generalization error, confirming the benefit of prominent, task-relevant data features. This result

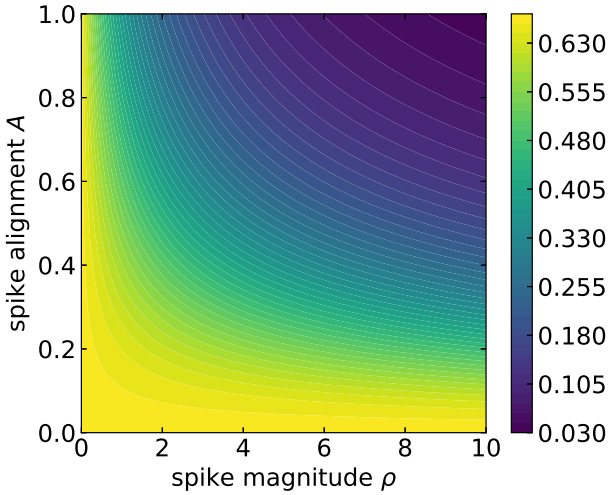
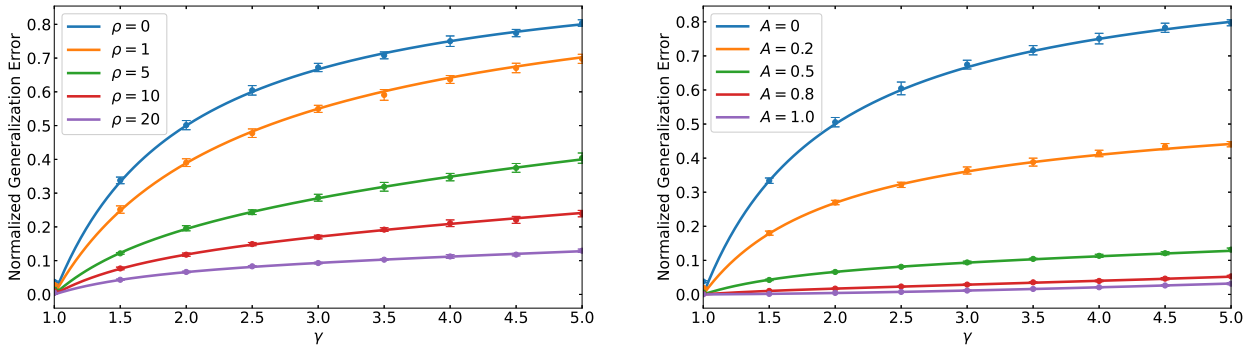


Figure 4: Normalized generalization error $\hat{R}_0/\boldsymbol{\beta}^\top\boldsymbol{\Sigma}\boldsymbol{\beta}$ as a function of spike-target alignment A and spike magnitude ρ for a fixed ratio $\gamma = 3$.



(a) Varying ρ while keeping $A = 0.5$ fixed.

(b) Varying A while keeping $\rho = 20$ fixed.

Figure 5: Normalized generalization error $\hat{R}_0/\boldsymbol{\beta}^\top\boldsymbol{\Sigma}\boldsymbol{\beta}$ as a function of the ratio $\gamma = d/n$. Solid lines represent the theoretical prediction (Equation (27)) and points are results of numerical simulations. The empirical results are averaged over 10 independent trials, with error bars indicating the standard deviation. All simulations were performed with a two-layer linear network of input dimension $d = 3000$ and width $m = 3000$, with $\sigma^2 = 1$. The initialization procedure is the same as in Figure 1.

extends previous work on isotropic data by quantifying the impact of the spiked covariance structure, a fundamental model of data anisotropy.

Our findings connect to and extend several lines of prior research. While the dynamics of linear networks on isotropic data are well-understood (Fukumizu, 1998; Saxe et al., 2013; 2019; Braun et al., 2022; Dominé et al., 2024), and the initial alignment phase under small initialization has been previously identified (Atanasov et al., 2022), our work investigates the later phases of learning under a specific anisotropic model. This provides a more complete picture of how different components of the solution are learned over time. The generalization error analysis complements previous works (Hastie et al., 2022; Wu & Xu, 2020; Richards et al., 2021; Mel & Ganguli, 2021) by providing concrete expressions for the spiked model, making the effects of anisotropy explicit.

However, our study has several limitations. The analysis is restricted to two-layer linear networks, and the spiked covariance model represents only a specific form of anisotropy. Additionally, the theory developed here employs small initialization, leaving the effects of varying initialization scales between layers unexplored.

These limitations suggest several directions for future research. A direct extension of our work would involve generalizing the analysis to non-linear activations and deeper architectures. Based on our findings, another valuable direction would be to investigate learning dynamics under more complex anisotropic structures, such as multiple spikes or power-law spectral densities that are ubiquitous in natural datasets. Moreover, reconciling our dynamics and generalization analyses within a unified theoretical framework remains a significant problem. Finally, a crucial next step is to explore practical strategies, such as initialization schemes, learning rate schedules, and optimization algorithms, that explicitly exploit data anisotropy to achieve faster convergence and lower generalization error.

In conclusion, this work provides a theoretical perspective on the relationship between data anisotropy, learning dynamics, and generalization in linear networks. By analyzing how a simple anisotropic structure influences learning phases and the final performance, we contribute to a deeper understanding of the mechanisms that shape learning in structured data environments.

References

Ben Adlam and Jeffrey Pennington. The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*, pp. 74–84. PMLR, 2020.

Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

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.

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

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022.

Alexander Atanasov, Jacob A Zavatone-Veth, and Cengiz Pehlevan. Scaling and renormalization in high-dimensional regression. *arXiv preprint arXiv:2405.00592*, 2024.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Advances in Neural Information Processing Systems*, 35:6615–6629, 2022.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1): 2914, 2021.

Clémentine CJ Dominé, Nicolas Anguita, Alexandra M Proca, Lukas Braun, Daniel Kunin, Pedro AM Mediano, and Andrew M Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. *arXiv preprint arXiv:2409.14623*, 2024.

Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In *International Conference on Machine Learning*, pp. 1655–1664. PMLR, 2019.

Kenji Fukumizu. Effect of batch learning in multilayer neural networks. In *International Conference on Neural Information Processing*, 1998.

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.

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.

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.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

Dongsung Huh. Curvature-corrected learning dynamics in deep neural networks. In *International Conference on Machine Learning*, pp. 4552–4560. PMLR, 2020.

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

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of statistics*, 29(2):295–327, 2001.

Daniel Kunin, Atsushi Yamamura, Chao Ma, and Surya Ganguli. The asymmetric maximum margin bias of quasi-homogeneous neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Daniel Kunin, Allan Raventos, Clémentine Carla Juliette Dominé, Feng Chen, David Klindt, Andrew M Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

Pierre Marion and Lénaïc Chizat. Deep linear networks for regression are implicitly regularized towards flat minima. *arXiv preprint arXiv:2405.13456*, 2024.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

Gabriel Mel and Surya Ganguli. A theory of high dimensional regression with arbitrary correlations between input features and target functions: sample complexity, multiple descent curves and a hierarchy of phase transitions. In *International Conference on Machine Learning*, pp. 7578–7587. PMLR, 2021.

Gabriel Mel and Jeffrey Pennington. Anisotropic random feature regression in high dimensions. In *International Conference on Learning Representations*, 2021.

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*, pp. 7760–7768. PMLR, 2021.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34: 29218–29230, 2021.

Tomaso Poggio, Qianli Liao, Brando Miranda, Andrzej Banburski, Xavier Boix, and Jack Hidary. Theory iiib: Generalization in deep networks. *arXiv preprint arXiv:1806.11379*, 2018.

Dominic Richards, Jaouad Mourtada, and Lorenzo Rosasco. Asymptotics of ridge (less) regression under general source condition. In *International Conference on Artificial Intelligence and Statistics*, pp. 3889–3897. PMLR, 2021.

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.

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.

Haozhe Shan and Blake Bordelon. A theory of neural tangent kernel alignment and its influence on training. *arXiv preprint arXiv:2105.14301*, 2021.

Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, pp. 10153–10161. PMLR, 2021.

Aditya Vardhan Varre, Maria-Luiza Vladarean, Loucas Pillaud-Vivien, and Nicolas Flammarion. On the spectral bias of two-layer linear networks. *Advances in Neural Information Processing Systems*, 36:64380–64414, 2023.

Alexander Wei, Wei Hu, and Jacob Steinhardt. More than a toy: Random matrix models predict how real-world neural representations generalize. In *International conference on machine learning*, pp. 23549–23588. PMLR, 2022.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Denny Wu and Ji Xu. On the optimal weighted ℓ_2 regularization in overparameterized linear regression. *Advances in Neural Information Processing Systems*, 33:10112–10123, 2020.

Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *International Conference on Learning Representations*, 2021.

Yedi Zhang, Peter E. Latham, and Andrew M Saxe. Understanding unimodal bias in multimodal deep linear networks. In *Forty-first International Conference on Machine Learning*, 2024.

A Derivation of Input-Output Correlation and Input Covariance Components

In this section, we provide detailed derivations for the quantities related to the correlation between the spike and the target signal, discussed in the main text. We use the notation $A = \boldsymbol{\mu}^\top \boldsymbol{\beta}$.

First, we calculate the norm of the input-output correlation $\boldsymbol{\Sigma}_{xy}$. Using Equation (1) and Equation (3), we can derive it as follows:

$$\begin{aligned} \|\boldsymbol{\Sigma}_{xy}\| &= \|\sigma^2(\boldsymbol{I} + \rho\boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta}\| \\ &= \sigma^2\sqrt{1 + 2\rho A^2 + \rho^2 A^2} \\ &= \sigma^2\sqrt{1 + ((1+\rho)^2 - 1)A^2}. \end{aligned} \tag{28}$$

Next, we calculate the components of the input covariance matrix $\boldsymbol{\Sigma}$ in the basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$. These components are the diagonal terms $\lambda_1 = \boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_1$ and $\lambda_2 = \boldsymbol{v}_2^\top \boldsymbol{\Sigma} \boldsymbol{v}_2$, and the off-diagonal term $\nu = -\boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_2$. To compute them, we first need the inner products of the spike direction $\boldsymbol{\mu}$ with the basis vectors.

The inner product between $\boldsymbol{\mu}$ and $\boldsymbol{v}_1$ is computed as follows:

$$\begin{aligned}\boldsymbol{\mu}^\top \boldsymbol{v}_1 &= \frac{1}{\|\boldsymbol{\Sigma}_{xy}\|}\sigma^2 \boldsymbol{\mu}^\top (\boldsymbol{I} + \rho \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{\beta} \\ &= \frac{(1+\rho)A}{\sqrt{1+((1+\rho)^2-1)A^2}}.\end{aligned} \tag{29}$$

Since $\|\boldsymbol{\mu}\| = 1$ and $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ form an orthonormal basis for the relevant subspace, we have $(\boldsymbol{\mu}^\top \boldsymbol{v}_1)^2 + (\boldsymbol{\mu}^\top \boldsymbol{v}_2)^2 = 1$. The inner product of $\boldsymbol{\mu}$ with $\boldsymbol{v}_2$ is then calculated as:

$$\begin{aligned}\boldsymbol{\mu}^\top \boldsymbol{v}_2 &= -\sqrt{1-(\boldsymbol{\mu}^\top \boldsymbol{v}_1)^2} \\ &= -\sqrt{1-\frac{(1+\rho)^2 A^2}{1+((1+\rho)^2-1)A^2}}.\end{aligned} \tag{30}$$

Using these inner products, we can now calculate the components of the covariance matrix. The first diagonal component, λ_1, is:

$$\begin{aligned}\lambda_1 &= \boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_1 \\ &= \sigma^2 \boldsymbol{v}_1^\top (\boldsymbol{I} + \rho \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{v}_1 \\ &= \sigma^2 (1 + \rho (\boldsymbol{\mu}^\top \boldsymbol{v}_1)^2) \\ &= \sigma^2 \left[1 + \rho \frac{(1+\rho)^2 A^2}{1+((1+\rho)^2-1)A^2}\right].\end{aligned} \tag{31}$$

Similarly, the second diagonal component, λ_2, is:

$$\begin{aligned}\lambda_2 &= \boldsymbol{v}_2^\top \boldsymbol{\Sigma} \boldsymbol{v}_2 \\ &= \sigma^2 \boldsymbol{v}_2^\top (\boldsymbol{I} + \rho \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{v}_2 \\ &= \sigma^2 (1 + \rho (\boldsymbol{\mu}^\top \boldsymbol{v}_2)^2) \\ &= \sigma^2 \left[1 + \rho \left(1 - \frac{(1+\rho)^2 A^2}{1+((1+\rho)^2-1)A^2}\right)\right].\end{aligned} \tag{32}$$

The off-diagonal term, ν, is calculated as follows:

$$\begin{aligned}\nu &= -\boldsymbol{v}_1^\top \boldsymbol{\Sigma} \boldsymbol{v}_2 \\ &= -\sigma^2 \boldsymbol{v}_1^\top (\boldsymbol{I} + \rho \boldsymbol{\mu}\boldsymbol{\mu}^\top)\boldsymbol{v}_2 \\ &= -\sigma^2 (\boldsymbol{v}_1^\top \boldsymbol{v}_2 + \rho (\boldsymbol{\mu}^\top \boldsymbol{v}_1)(\boldsymbol{\mu}^\top \boldsymbol{v}_2)) \\ &= -\sigma^2 \rho (\boldsymbol{\mu}^\top \boldsymbol{v}_1)(\boldsymbol{\mu}^\top \boldsymbol{v}_2) \\ &= \sigma^2 \rho \frac{(1+\rho)A\sqrt{1-A^2}}{1+((1+\rho)^2-1)A^2}.\end{aligned} \tag{33}$$

B Derivation of the Dynamics

B.1 Early Phase

In this subsection, we provide the derivation for the dynamics of the weight vectors in the early phase Equation (17) in the main text.

At the beginning of the early phase of learning, the dynamics of a two-layer linear network initialized with small random weights can be approximated by Equation (14):

$$\tau \frac{d}{dt}\boldsymbol{w}_1 = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{a}, \quad \tau \frac{d}{dt}\boldsymbol{a} = \|\boldsymbol{\Sigma}_{xy}\|\boldsymbol{w}_1. \tag{34}$$

Solving these equations, we obtain

$$\boldsymbol{w}_1(t) = \frac{\boldsymbol{w}_1(0)+\boldsymbol{a}(0)}{2} e^{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau}t} + \frac{\boldsymbol{w}_1(0)-\boldsymbol{a}(0)}{2} e^{-\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau}t}, \quad \boldsymbol{a}(t) = \frac{\boldsymbol{w}_1(0)+\boldsymbol{a}(0)}{2} e^{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau}t} - \frac{\boldsymbol{w}_1(0)-\boldsymbol{a}(0)}{2} e^{-\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\tau}t}. \tag{35}$$

The second term in each equation is of order $O(s)$ and the first terms cause the weights to grow in the direction of $\boldsymbol{r}_1 = (\boldsymbol{w}_1(0)+\boldsymbol{a}(0))/2$. By defining the unit vector $\hat{\boldsymbol{r}}_1 = \boldsymbol{r}_1/\|\boldsymbol{r}_1\|$, we can write $\boldsymbol{w}_1(t) = u(t)\hat{\boldsymbol{r}}_1 + O(s)$ and $\boldsymbol{a}(t) = u(t)\hat{\boldsymbol{r}}_1 + O(s)$ for some positive scalar function $u(t)$. This approximation holds as long as the other components $\boldsymbol{w}_2, \ldots, \boldsymbol{w}_d$ are negligible. Substituting this expression into the differential equation for $\boldsymbol{w}_1$ in Equation (9) and neglecting the $O(s)$ terms yields the following equation for the magnitude $u(t)$:

$$\tau \frac{d}{dt} u = \|\boldsymbol{\Sigma}_{xy}\| u - \lambda_1 u^3. \tag{36}$$

Solving this Bernoulli differential equation gives the solution:

$$u(t) = \sqrt{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\lambda_1}\left(1 + \left(\frac{\|\boldsymbol{\Sigma}_{xy}\|/\lambda_1}{u(0)^2} - 1\right) e^{-\frac{2\|\boldsymbol{\Sigma}_{xy}\|}{\tau}t}\right)^{-1}}. \tag{37}$$

Substituting this solution back into the expression for $\boldsymbol{w}_1(t)$ and rearranging the terms, we obtain the analytical form in Equation (17):

$$\boldsymbol{w}_1(t) = \sqrt{\frac{\|\boldsymbol{\Sigma}_{xy}\|}{\left(1 - e^{-2\|\boldsymbol{\Sigma}_{xy}\|t/\tau} + \frac{\|\boldsymbol{\Sigma}_{xy}\|}{\lambda_1} u(0)^{-2} e^{-2\|\boldsymbol{\Sigma}_{xy}\|t/\tau}\right)\lambda_1}}\hat{\boldsymbol{r}}_1 + O(s). \tag{38}$$

B.2 Later Phase

In this subsection, we provide the derivation for the upper bound of the timescale in the later phase in the main text.

Let $w_{1*} = \lim_{t\to\infty} w_1(t)$ and $w_{2*} = \lim_{t\to\infty} w_2(t)$. We assume that $w_1 \geq w_{1*}$ and $w_2 \leq w_{2*}$ during the later phase $t \geq t_1$. Using this assumption, we can establish a lower bound on the derivative term

$$\tau \frac{dw_2}{dt} = (w_1^2 + w_2^2)(\nu w_1 - \lambda_2 w_2) \geq (w_{1*}^2 + w_2^2)(\nu w_{1*} - \lambda_2 w_2). \tag{39}$$

Therefore,

$$\begin{aligned}
t - t_1 &= \tau \int_{w_2(t_1)}^{w_2(t)} \frac{dw_2}{(w_1^2 + w_2^2)(\nu w_1 - \lambda_2 w_2)} \\
&\leq \tau \int_{w_2(t_1)}^{w_2(t)} \frac{dw_2}{(w_{1*}^2 + w_2^2)(\nu w_{1*} - \lambda_2 w_2)} \\
&= \tau \frac{1}{\lambda_2 w_{1*}^2 \left(1 + \left(\frac{\nu}{\lambda_2}\right)^2\right)} \int_{w_2(t_1)}^{w_2(t)} \left(\frac{w_2 + \frac{\nu}{\lambda_2} w_{1*}}{w_2^2 + w_{1*}^2} - \frac{1}{w_2 - \frac{\nu}{\lambda_2} w_{1*}}\right) dw_2 \\
&= \tau \frac{1}{\lambda_2 w_{1*}^2 \left(1 + \left(\frac{\nu}{\lambda_2}\right)^2\right)} \left[\frac{\nu}{\lambda_2} \arctan\left(\frac{w_2}{w_{1*}}\right) + \frac{1}{2}\log(w_2^2 + w_{1*}^2) - \log\left|w_2 - \frac{\nu}{\lambda_2} w_{1*}\right|\right]_{w_2(t_1)}^{w_2(t)}.
\end{aligned} \tag{40}$$

Now, let $a_* = \lim_{t\to\infty} a(t)$. Then, as the initialization scale $s \to 0$, we have $a_* = \sqrt{w_{1*}^2 + w_{2*}^2}$, and $\lim_{t\to\infty} \|\boldsymbol{a}^\top \boldsymbol{W}\| = a_* \sqrt{w_{1*}^2 + w_{2*}^2} = w_{1*}^2 + w_{2*}^2$. Since this value must equal $\|\boldsymbol{\beta}\| = 1$, we obtain $w_{1*}^2 + w_{2*}^2 = 1$. Furthermore, from the fixed-point condition $w_{1*} = \lambda_2 w_{2*}/\nu$, it follows that $w_{2*} = \nu w_{1*}/\lambda_2$. This implies

$$w_{1*}^2 + w_{2*}^2 = w_{1*}^2 \left(1 + \left(\frac{\nu}{\lambda_2}\right)^2\right) = 1. \tag{41}$$

Using this result to evaluate the time t_2 at which $w_2(t_2) = (1 - e^{-\delta})w_{2*} = (1 - e^{-\delta})\nu w_{1*}/\lambda_2$, we can obtain the bound in Equation (20):

$$t_2 - t_1 \leq \tau \frac{1}{\lambda_2} \left[\frac{\nu}{\lambda_2} \arctan\left(\frac{\nu}{\lambda_2}(1 - e^{-\delta}) \right) + \frac{1}{2} \log\left(1 + \left(\frac{\nu}{\lambda_2}(1 - e^{-\delta}) \right)^2 \right) + \delta \right]. \tag{42}$$

C Proof of Weight Vector Alignment

In this section, we prove that in the small initialization limit ($s \to 0$), the weight vectors $\boldsymbol{a}(t)$, $\boldsymbol{w}_1(t)$, and $\boldsymbol{w}_2(t)$ remain aligned with the initial growth direction $\hat{\boldsymbol{r}}_1$ throughout training. Specifically, we show that the components of each weight vector orthogonal to $\hat{\boldsymbol{r}}_1$ vanish as $s \to 0$.

To formalize this, we decompose each weight vector into components parallel and orthogonal to $\hat{\boldsymbol{r}}_1$. The orthogonal components, which we term deviation vectors, are defined as $\Delta\boldsymbol{a}(t) = \boldsymbol{P}_\perp \boldsymbol{a}(t)$ and $\Delta\boldsymbol{w}_i(t) = \boldsymbol{P}_\perp \boldsymbol{w}_i(t)$, where $\boldsymbol{P}_\perp = \boldsymbol{I} - \hat{\boldsymbol{r}}_1 \hat{\boldsymbol{r}}_1^\top$ is the projection matrix onto the subspace orthogonal to $\hat{\boldsymbol{r}}_1$.

The dynamics of the deviation vectors are derived by applying the projection $\boldsymbol{P}_\perp$ to the gradient flow equations (Equations (9), (10) and (12)). Applying $\boldsymbol{P}_\perp$ to the dynamics of $\boldsymbol{w}_1$ in Equation (9) gives:

$$\begin{aligned} \tau \frac{d}{dt} \Delta\boldsymbol{w}_1 &= \boldsymbol{P}_\perp \left(\|\boldsymbol{\Sigma}_{xy}\| \boldsymbol{a} - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2) \boldsymbol{a} \right) \\ &= (\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)) \Delta\boldsymbol{a}. \end{aligned} \tag{43}$$

Similarly, for $\boldsymbol{w}_2$ in Equation (10):

$$\begin{aligned} \tau \frac{d}{dt} \Delta\boldsymbol{w}_2 &= \boldsymbol{P}_\perp \left(-(-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2) \boldsymbol{a} \right) \\ &= -(-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2) \Delta\boldsymbol{a}, \end{aligned} \tag{44}$$

and for $\boldsymbol{a}$ in Equation (12):

$$\begin{aligned} \tau \frac{d}{dt} \Delta\boldsymbol{a} &= \boldsymbol{P}_\perp \left(\|\boldsymbol{\Sigma}_{xy}\| \boldsymbol{w}_1 - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2) \boldsymbol{w}_1 - (-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2) \boldsymbol{w}_2 - \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \boldsymbol{w}_i \right) \\ &= (\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)) \Delta\boldsymbol{w}_1 - (-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2) \Delta\boldsymbol{w}_2 + \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \Delta\boldsymbol{w}_i. \end{aligned} \tag{45}$$

To show that the deviation vectors remain small, we define

$$E(t) = \frac{1}{2}(\|\Delta\boldsymbol{a}\|^2 + \|\Delta\boldsymbol{w}_1\|^2 + \|\Delta\boldsymbol{w}_2\|^2). \tag{46}$$

By substituting Equations (43) to (45) into the time derivative of $E(t)$, we obtain:

$$\begin{aligned} \tau \frac{dE}{dt} &= \Delta\boldsymbol{a}^\top \tau \frac{d\Delta\boldsymbol{a}}{dt} + \Delta\boldsymbol{w}_1^\top \tau \frac{d\Delta\boldsymbol{w}_1}{dt} + \Delta\boldsymbol{w}_2^\top \tau \frac{d\Delta\boldsymbol{w}_2}{dt} \\ &= 2(\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)) \Delta\boldsymbol{a}^\top \Delta\boldsymbol{w}_1 - 2(-\nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2) \Delta\boldsymbol{a}^\top \Delta\boldsymbol{w}_2 + \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \Delta\boldsymbol{a}^\top \Delta\boldsymbol{w}_i. \end{aligned} \tag{47}$$

Applying the Cauchy-Schwarz inequality, $\boldsymbol{x}^\top \boldsymbol{y} \leq \|\boldsymbol{x}\|\|\boldsymbol{y}\|$, and the inequality $2\|\boldsymbol{x}\|\|\boldsymbol{y}\| \leq \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2$, we bound the derivative:

$$\begin{aligned}
\tau \frac{dE}{dt} &\leq 2|(\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2))| \, \|\Delta \boldsymbol{a}\| \|\Delta \boldsymbol{w}_1\| + 2| - \nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2| \, \|\Delta \boldsymbol{a}\| \|\Delta \boldsymbol{w}_2\| + \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \|\Delta \boldsymbol{a}\| \|\Delta \boldsymbol{w}_i\| \\
&\leq |\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)|(\|\Delta \boldsymbol{a}\|^2 + \|\Delta \boldsymbol{w}_1\|^2) + | - \nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2|(\|\Delta \boldsymbol{w}_2\|^2 + \|\Delta \boldsymbol{a}\|^2) \\
&\quad + \frac{1}{2} \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i (\|\Delta \boldsymbol{a}\|^2 + \|\Delta \boldsymbol{w}_i\|^2) \\
&\leq C(t) E(t) + O(s^3),
\end{aligned} \tag{48}$$

where

$$C(t) = |\|\boldsymbol{\Sigma}_{xy}\| - (\lambda_1 \boldsymbol{a}^\top \boldsymbol{w}_1 - \nu \boldsymbol{a}^\top \boldsymbol{w}_2)| + | - \nu \boldsymbol{a}^\top \boldsymbol{w}_1 + \lambda_2 \boldsymbol{a}^\top \boldsymbol{w}_2| + \frac{1}{2} \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i. \tag{49}$$

The $O(s^3)$ term arises from the sum $\frac{1}{2} \sum_{i=3}^{d} \sigma^2 \boldsymbol{a}^\top \boldsymbol{w}_i \|\Delta \boldsymbol{w}_i\|^2$. For $i \geq 3$, the weights $\boldsymbol{w}_i$ remain of order $O(s)$ throughout the dynamics. This implies $\boldsymbol{a}^\top \boldsymbol{w}_i = O(s)$ and $\|\Delta \boldsymbol{w}_i\| \leq \|\boldsymbol{w}_i\| = O(s)$.

Applying Grönwall's inequality to the differential inequality in Equation (48) gives the following bound on $E(t)$:

$$E(t) \leq \left(E(0) + \int_0^t O(s^3) dt' \right) \exp \left(\int_0^t \frac{C(t')}{\tau} dt' \right). \tag{50}$$

To evaluate this bound, we first show that $C(t)$ is bounded by examining the time derivative of the loss function $\mathcal{L}$:

$$\frac{d\mathcal{L}}{dt} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}}^\top \frac{d\boldsymbol{a}}{dt} + \sum_{i=1}^{d} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i}^\top \frac{d\boldsymbol{w}_i}{dt} = -\frac{1}{\tau} \left(\left\| \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}} \right\|^2 + \sum_{i=1}^{d} \left\| \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i} \right\|^2 \right) \leq 0. \tag{51}$$

This inequality shows that the loss is non-increasing, so $\mathcal{L}(t) \leq \mathcal{L}(0)$ for all $t \geq 0$. Additionally, the definition of $\mathcal{L}$ in Equation (5) implies that $\mathcal{L}(t) \geq 0$. Since the loss is bounded, the weights $\boldsymbol{a}(t)$ and $\boldsymbol{w}_i(t)$ must also remain within a bounded region. Consequently, $C(t)$, which is a function of these weights, is bounded for all time.

Since $C(t)$ is bounded, the exponential term in Equation (50) is bounded for any finite time t. $E(0) = O(s^2)$ and the integral term involving $O(s^3)$ does not change this order for any finite time t. Thus, the bound shows that $E(t)$ remains of order $O(s^2)$. This result shows that the norms of the deviation vectors $\|\Delta \boldsymbol{a}\|$, $\|\Delta \boldsymbol{w}_1\|$, and $\|\Delta \boldsymbol{w}_2\|$ stay of order $O(s)$ throughout training. In the $s \to 0$ limit, these deviations vanish, meaning that the vectors $\boldsymbol{a}(t), \boldsymbol{w}_1(t), \boldsymbol{w}_2(t)$ become perfectly co-aligned with $\hat{\boldsymbol{r}}_1$.