Towards a Statistical Understanding of Neural Networks: Beyond the Neural Tangent Kernel Theories

Abstract

A primary advantage of neural networks lies in their feature learning characteristics, which is challenging to theoretically analyze due to the complexity of their training dynamics. We propose a new paradigm for studying feature learning and the resulting benefits in generalizability. After reviewing the neural tangent kernel (NTK) theory and recent results in kernel regression, which address the generalization issue of sufficiently wide neural networks, we examine limitations and implications of the fixed kernel theory (as the NTK theory) and review recent theoretical advancements in feature learning. Moving beyond the fixed kernel/feature theory, we consider neural networks as adaptive feature models. Finally, we propose an over-parameterized Gaussian sequence model as a prototype model to study the feature learning characteristics of neural networks.

Keywords: neural networks, generalization ability, feature learning, kernel regression, over-parameterized Gaussian sequence model

Mathematics Subject Classification (2020): 62G05

1 Introduction

In recent years, deep neural networks have made remarkable accomplishments in many application areas, whereas their theoretical understanding has lagged far behind. Although neural network modeling has a long history, recent developments in data availability, computing resources, and network architecture designs were believed to be the key to their somewhat mysteriously outstanding performance.

Numerous topics have been explored regarding neural network theory, such as those discussed in recent review papers and references therein (Fan et al., 2021; Bartlett et al., 2021; Belkin, 2021; Suh and Cheng, 2024). In this paper, we consider the generalization ability of neural networks within the nonparametric regression framework. Let a compact set $\mathcal{X} \subset \mathbb{R}^d$ be the input space, let $\mathcal{Y} \subseteq \mathbb{R}$ be the output space, and let μ be the marginal distribution of $\mathcal{X}$. Suppose that the n i.i.d. draws $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are sampled from the model

$$y = f^*(\boldsymbol{x}) + \epsilon, \quad \boldsymbol{x} \sim \mu, \tag{1}$$

where $f^* \in L^2(\mathcal{X}, \mu)$ is the true function, and the noise ϵ has zero mean and variance $\sigma_0^2 > 0$. Throughout the paper, we denote $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{R}^{d \times n}$, and $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$. For any estimator $\hat{f}$ (which can be generated by a neural network, a kernel regression model, etc.), we

will investigate the L^2-norm generalization error (or excess risk) given by

$$\mathcal{R}(\hat{f}) = \left\|\hat{f} - f^*\right\|_2^2 \triangleq \mathbb{E}_{\boldsymbol{x}\sim\mu}\left[\left(\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x})\right)^2\right]. \tag{2}$$

It is natural to ask the following questions about neural networks:

(i) How can the generalization ability of neural networks be characterized from a statistical viewpoint? For instance, can we establish convergence rates of the generalization error with respect to sample sizes? Do we know the minimax optimality of a neural network method?

(ii) Why do neural networks outperform other existing methods in many applications? Can we understand when and how?

To answer the first question, the line of work of Bauer and Kohler (2019), Schmidt-Hieber (2017), and Suzuki (2018) considered the "algorithm-independent controls" of the generalization error, a name coined by Fan et al. (2021). By carefully selecting the candidate function class represented by a specific neural network architecture, they obtained generalization error bounds for the empirical risk minimization estimator over the chosen function class. For comparison, our current paper focuses on *algorithm-dependent controls* (Fan et al., 2021), which take into account the training dynamics of neural networks. In practice, training dynamics are usually difficult to understand and can lead to drastically different generalization behaviors. This makes the theoretical analysis of training dynamics one of the most intriguing and challenging tasks.

Our starting point is the seminal *neural tangent kernel* (NTK) theory (Jacot et al., 2018; Arora et al., 2019b; Lee et al., 2019), which establishes a connection between the training dynamics of sufficiently wide neural networks and the kernel gradient flow (Section 2.1). Due to the success of the NTK theory, we have experienced a renaissance of kernel regression. We will review some important topics about the generalization ability of kernel regression, addressing both fixed-dimensional (Section 2.3) and high-dimensional settings (Section 2.4).

The answer to the second question is even more complicated. It is believed that the superiority of neural networks stems from their feature learning characteristics, which cannot be addressed by the NTK theory. Therefore, another theme of this paper is to go beyond the NTK theory and propose a new paradigm to analyze feature learning characteristics of neural networks. Specifically, in Section 3, we discuss limitations and implications of the NTK theory (as the fixed kernel regression) and review some recent advances in feature learning theory. In Section 4, we propose an *adaptive feature model* to emulate the feature learning process of neural networks. Furthermore, we introduce the *over-parameterized Gaussian sequence model* as a viable prototype of the adaptive feature model. This model offers a convenient way for us to qualitatively analyze both the training dynamics and the feature learning characteristics of the adaptive feature model. Finally, we discuss some future questions in Section 5. We hope that our review and the proposed paradigms can offer new insights and directions for developing theoretical understandings of neural networks.

Notations: The asymptotic notations, $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$ and $\Theta(\cdot)$, will be used throughout the paper: We say $a_n = \Omega(b_n)$ if $b_n = O(a_n)$; $a_n = \Theta(b_n)$ if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$. We

also write $a_n \asymp b_n$ for $a_n = \Theta(b_n)$, and $a_n \lesssim b_n$ for $a_n = O(b_n)$. The corresponding probabilistic versions of the asymptotic notations will also be used: $O_{\mathbb{P}}(\cdot), o_{\mathbb{P}}(\cdot), \Omega_{\mathbb{P}}(\cdot)$ and $\Theta_{\mathbb{P}}(\cdot)$. For example, we say that random variables X_n, Y_n satisfy $X_n = O_{\mathbb{P}}(Y_n)$ if $\forall\ \varepsilon > 0$, there exist constants C_ε and N_ε such that $\mathbb{P}\left(|X_n| \geq C_\varepsilon |Y_n|\right) \leq \varepsilon$ for $n > N_\varepsilon$. For $\alpha > 0$, we write $\lfloor \alpha \rfloor$ as the largest integer not exceeding α. We use boldface letters to denote column vectors and matrices; $\|\boldsymbol{x}\|_2$ is the ℓ^2-norm of vector $\boldsymbol{x}$.

2 Neural tangent kernel theory & kernel regression

2.1 Neural tangent kernel theory

The neural tangent kernel (NTK) theory (Jacot et al., 2018; Arora et al., 2019b; Lee et al., 2019) is one of the most successful tools for analyzing neural networks' training dynamics and generalization ability. A major challenge in studying neural networks is the highly non-convex nature of the target function. The basic idea of the NTK theory is that when a neural network's width is sufficiently large, the network's training dynamics can be approximated well by the kernel gradient flow for the corresponding neural tangent kernel, which is a convex problem. Numerous papers discuss various topics within the NTK theory in different settings, e.g. Lai et al. (2023), Du et al. (2019), Huang et al. (2020), Arora et al. (2019a), Li et al. (2024c), and Nitanda and Suzuki (2020).

A purpose of our introduction of the NTK theory is to emphasize the importance of studying kernel regression, which prepares us for proposing the adaptive feature approach in subsequent sections. Therefore, here we only review one result from Lai et al. (2023) to establish the connection between neural networks and kernel gradient flow. Specifically, Lai et al. (2023) considered the following two-layer fully connected ReLU neural network with width m:

$$f_{\boldsymbol{\theta}}^m(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(\boldsymbol{w}_r^\top \boldsymbol{x} + b_r\right) + b, \tag{3}$$

where $\boldsymbol{\theta} = \mathrm{vec}(\{a_r, \boldsymbol{w}_r, b_r, b, r = 1, \ldots, m\})$ and $\sigma(\cdot)$ is the ReLU function. The neural network is trained through the gradient flow of parameters $\boldsymbol{\theta}(t)$ to minimize the mean-squared loss:

$$\mathcal{L}\left(f_{\boldsymbol{\theta}}^m\right) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - f_{\boldsymbol{\theta}}^m\left(\boldsymbol{x}_i\right)\right)^2,$$

i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}(t) = -\nabla_{\boldsymbol{\theta}} \mathcal{L}\left(f_{\boldsymbol{\theta}(t)}^m\right).$$

Then, $f_{\boldsymbol{\theta}(t)}^m$ is the neural network estimator at time t (see Lai et al. 2023, Section 3 for more details). Additionally, denote $\hat{f}_t^{\mathrm{NTK}}$ as the kernel gradient flow estimator (defined later in Definition 2.2.2) with the corresponding NTK. Under certain initialization, Lai et al. (2023), Proposition 2, showed that for any $\varepsilon > 0$, if the width m is sufficiently large, then

$$\sup_{t \geq 0} \sup_{\boldsymbol{x} \in \mathcal{X}} \left| f_{\boldsymbol{\theta}(t)}^m(\boldsymbol{x}) - \hat{f}_t^{\mathrm{NTK}}(\boldsymbol{x}) \right| \leq \varepsilon,$$

and

$$\sup_{t\geq 0}\left|\mathcal{R}\left(f^m_{\boldsymbol{\theta}(t)}\right)-\mathcal{R}\left(\hat{f}^{\mathrm{NTK}}_t\right)\right|\leq\varepsilon,$$

hold with probability at least $1-o_m(1)$, where the randomness comes from the initialization of the parameters. In other words, as the width m tends to infinity, the neural network estimator $f^m_{\boldsymbol{\theta}(t)}$ uniformly converges to $\hat{f}^{\mathrm{NTK}}_t$, and the corresponding generalization error $\mathcal{R}(f^m_{\boldsymbol{\theta}(t)})$ is well approximated by $\mathcal{R}(\hat{f}^{\mathrm{NTK}}_t)$. Similar results have also been derived for other neural network architectures (Arora et al., 2019b; Tirer et al., 2022, etc.), making the kernel gradient flow a reasonable alternative for sufficiently wide neural networks and leading to a resurgence in the study of kernel regression.

2.2 Preliminaries of kernel regression

In kernel regression (or kernel method), we are given a pre-specified kernel function $k(\cdot,\cdot):\mathcal{X}\times\mathcal{X}\to\mathbb{R}$, which is supposed to be positive-definite, symmetric, and continuous. Without loss of generality, we assume that the kernel function is bounded by 1, that is, $\sup_{\boldsymbol{x}\in\mathcal{X}}k(\boldsymbol{x},\boldsymbol{x})\leq 1$. Then, there exists a corresponding function space $\mathcal{H}\subset L^2(\mathcal{X},\mu)$, called the reproducing kernel Hilbert space (RKHS), with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$. Mercer's theorem (see Steinwart and Christmann 2008, Theorem 4.49) shows that there exists a non-increasing summable sequence $\{\lambda_j\}_{j=1}^{\infty}\subset(0,\infty)$ and a family of functions $\{\psi_j\}_{j=1}^{\infty}\subset\mathcal{H}$, such that

$$k\left(\boldsymbol{x},\boldsymbol{x}'\right)=\sum_{j=1}^{\infty}\lambda_j\psi_j(\boldsymbol{x})\psi_j\left(\boldsymbol{x}'\right),\quad \boldsymbol{x},\boldsymbol{x}'\in\mathcal{X},\tag{4}$$

where the convergence is absolute and uniform. $\{\psi_j\}_{j=1}^{\infty}$ and $\{\lambda_j\}_{j=1}^{\infty}$ are the eigenfunctions and eigenvalues, respectively, of the kernel function k and the RKHS $\mathcal{H}$. The set of functions $\{\psi_j\}_{j=1}^{\infty}$ can be assumed to be an orthonormal basis of $L^2(\mathcal{X},\mu)$ without loss of generality. The RKHS $\mathcal{H}$ can be expressed as

$$\mathcal{H}=\left\{\sum_{j=1}^{\infty}a_j\lambda_j^{\frac{1}{2}}\psi_j(\cdot):\ (a_j)_{j=1}^{\infty}\in\ell^2\right\},$$

equipped with the norm

$$\left\|\sum_{j=1}^{\infty}a_j\lambda_j^{\frac{1}{2}}\psi_j(\cdot)\right\|_{\mathcal{H}}=\left(\sum_{j=1}^{\infty}a_j^2\right)^{\frac{1}{2}}.$$

For more details on RKHS, see Steinwart and Christmann (2008) and Steinwart and Scovel (2012).

The basic idea of kernel regression is to estimate f^* using candidate functions from $\mathcal{H}$ (Cucker and Smale, 2001; Kohler and Krzyżak, 2001; Steinwart and Christmann, 2008). A large class of kernel regression estimators is collectively introduced as spectral algorithms (Rosasco et al., 2005; Caponnetto, 2006; Gerfo et al., 2008). The two most widely studied are the *kernel ridge regression* and *kernel gradient flow*. For a kernel function k, we denote

$$\mathbb{K}(\boldsymbol{X},\boldsymbol{X})=\left(k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\right)_{n\times n},\qquad \mathbb{K}(\boldsymbol{x},\boldsymbol{X})=\left(k\left(\boldsymbol{x},\boldsymbol{x}_1\right),\ldots,k\left(\boldsymbol{x},\boldsymbol{x}_n\right)\right).$$

Definition 2.2.1 (Kernel ridge regression, KRR)**.** *For a given kernel function k, corresponding RKHS $\mathcal{H}$ and any $0 < \lambda < \infty$, kernel ridge regression constructs an estimator $\hat{f}_\lambda$ by solving the penalized least square problem*

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\boldsymbol{x}_i\right) \right)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right),$$

where λ is the regularization parameter. The explicit expression of $\hat{f}_\lambda$ is

$$\hat{f}_\lambda(\boldsymbol{x}) = \mathbb{K}(\boldsymbol{x}, \boldsymbol{X}) \left(\mathbb{K}(\boldsymbol{X}, \boldsymbol{X}) + n\lambda \mathbf{I}_n \right)^{-1} \boldsymbol{Y}.$$

Definition 2.2.2 (Kernel gradient flow, KGF)**.** *For a given kernel function k, corresponding RKHS $\mathcal{H}$, and any $0 < t < \infty$, kernel gradient flow constructs an estimator $\hat{f}_t$ by solving the differential equation*

$$\frac{\mathrm{d}}{\mathrm{d}t} \hat{f}_t(\boldsymbol{x}) = -\frac{1}{n} \mathbb{K}(\boldsymbol{x}, \boldsymbol{X}) \left(\hat{f}_t(\boldsymbol{X}) - \boldsymbol{Y} \right), \tag{5}$$

and let $\hat{f}_0(\boldsymbol{x}) \equiv 0$. For any $t > 0$, when the matrix $\mathbb{K}(\boldsymbol{X}, \boldsymbol{X})$ is strictly positive definite, the explicit expression of $\hat{f}_t$ is

$$\hat{f}_t(\boldsymbol{x}) = \mathbb{K}(\boldsymbol{x}, \boldsymbol{X}) \mathbb{K}(\boldsymbol{X}, \boldsymbol{X})^{-1} \left(\mathbf{I}_n - e^{-\frac{1}{n}\mathbb{K}(\boldsymbol{X}, \boldsymbol{X})t} \right) \boldsymbol{Y}. \tag{6}$$

In fact, the differential equation (5) is a gradient flow of $\hat{f}_t \in \mathcal{H}$ minimizing the training loss $\|\hat{f}_t(\boldsymbol{X}) - \boldsymbol{Y}\|_2^2/n$. If we assume the training time $t \to \infty$, it results in the *kernel interpolation estimator* (the same as kernel ridge regression with $\lambda \to 0$):

$$\hat{f}_{\text{inter}}(\boldsymbol{x}) = \mathbb{K}(\boldsymbol{x}, \boldsymbol{X}) \mathbb{K}(\boldsymbol{X}, \boldsymbol{X})^{-1} \boldsymbol{Y}, \tag{7}$$

which will be of independent interest.

t plays the same role as the regularization parameter λ in KRR, and the generalization property of $\hat{f}_\lambda$ is actually almost the same as that of $\hat{f}_t$ with $t \asymp \lambda^{-1}$ (except in the case of saturation effect of KRR, which will be introduced later). In the definition of general spectral algorithms, both t and λ can be unified as a regularization parameter in the spectral algorithm's filter function (see, e.g., Zhang et al. 2024a, Definition 1). Additionally, the kernel gradient flow is a continuous version of the kernel gradient descent, which has similar theoretical properties and is used more frequently in practice.

In the following, we focus on kernel ridge regression and kernel gradient flow. For studies of general spectral algorithms, see Lin et al. (2018), Blanchard and Mücke (2018), Lin and Cevher (2020), Zhang et al. (2024a), and Li et al. (2024a).

2.3 Kernel regression in fixed dimensions

2.3.1 Assumptions

In fixed dimensions, we disregard the constants' dependence on the input dimension d. The first commonly used assumption is the eigenvalue decay rate of the eigenvalues $\{\lambda_j\}_{j=1}^{\infty}$.

Assumption 1 (Eigenvalue decay rate, EDR)**.** *Suppose that the eigenvalue decay rate (EDR) of $\mathcal{H}$ is $\beta > 1$. That is, there exist positive constants c and C such that*

$$cj^{-\beta} \leq \lambda_j \leq Cj^{-\beta}, \quad \forall j = 1, 2, \ldots.$$

Assumption 1 holds for many kernels, e.g., the Laplacian kernel, Matérn kernel, neural tangent kernel, etc., and is also closely related to the effective dimension or the capacity condition of RKHS (Caponnetto and de Vito, 2007).

The second widely adopted assumption is the source condition, which characterizes the relative smoothness of f^* with respect to $\mathcal{H}$. To introduce this, we need the following definition of the *interpolation space* (or power space) $[\mathcal{H}]^s$ for any $s \geq 0$, which is defined as

$$[\mathcal{H}]^s = \left\{ \sum_{j=1}^{\infty} a_j \lambda_j^{\frac{s}{2}} \psi_j(\cdot) : (a_j)_{j=1}^{\infty} \in \ell^2 \right\}, \tag{8}$$

equipped with the norm

$$\left\| \sum_{j=1}^{\infty} a_j \lambda_j^{\frac{s}{2}} \psi_j(\cdot) \right\|_{[\mathcal{H}]^s} = \left(\sum_{j=1}^{\infty} a_j^2 \right)^{\frac{1}{2}}.$$

Specifically, we have $[\mathcal{H}]^1 = \mathcal{H}$. For $0 < s_1 < s_2$, the embeddings $[\mathcal{H}]^{s_2} \hookrightarrow [\mathcal{H}]^{s_1} \hookrightarrow [\mathcal{H}]^0$ exist and are compact (Fischer and Steinwart, 2020). The functions in $[\mathcal{H}]^s$ with smaller s are less "smooth", which are harder for a kernel regression algorithm to estimate. In fact, another equivalent definition of $[\mathcal{H}]^s$ is through the *real interpolation* in functional analysis (Tartar, 2007; Sawano, 2018; Steinwart and Scovel, 2012).

Assumption 2 (Source condition)**.** *Suppose that for some $s \geq 0$, there is a constant $R > 0$ such that $f^* \in [\mathcal{H}]^s$ and*

$$\|f^*\|_{[\mathcal{H}]^s} \leq R.$$

We refer to s as the source condition of f^*.

2.3.2 Learning curve results

The first question of interest is the minimax optimality of kernel regression. In the framework of eigenvalue decay rate and source conditions, the minimax optimality was first established in the well-specified case ($f^* \in \mathcal{H}$, or the source condition $s \geq 1$) in Caponnetto (2006); Caponnetto and de Vito (2007). Then, extensive subsequent literature (Steinwart et al., 2009; Fischer and Steinwart, 2020; Zhang et al., 2023; Pillaud-Vivien et al., 2018; Celisse and Wahl, 2020; Dieuleveut and Bach, 2016; Wang and Jing, 2022; Li et al., 2024b) studied mis-specified case (source condition $0 < s < 1$). Among them, Fischer and Steinwart (2020) firstly considered the embedding property of $\mathcal{H}$: we say that $\mathcal{H}$ has an embedding property of order $\alpha \in (0, 1]$ if $[\mathcal{H}]^\alpha$ can be continuously embedded into $L^\infty(\mathcal{X}, \mu)$, i.e., the operator norm of the embedding satisfies

$$\|[\mathcal{H}]^\alpha \hookrightarrow L^\infty(\mathcal{X}, \mu)\|_{\text{op}} = M_\alpha < \infty.$$

This embedding property was later summarized as an embedding index assumption (see, e.g., Li et al. 2024b, Assumption 2) and led to the minimax optimality of spectral algorithms for $0 < s < 1$ (Zhang et al., 2024a). The embedding index assumption postulates that $\alpha_0 = 1/\beta$, where α_0 is defined as

$$\alpha_0 = \inf \left\{ \alpha \in [1/\beta, 1] : \|[\mathcal{H}]^\alpha \hookrightarrow L^\infty(\mathcal{X}, \mu)\|_{\text{op}} < \infty \right\}.$$

The minimax optimality-type results considered the upper bound of the generalization error of an algorithm and the algorithm-independent minimax lower bound. In the renaissance of kernel regression arising from the study of neural networks, some recent work (Li et al., 2024b; Cui et al., 2021; Bordelon et al., 2020; Li et al., 2024a) further considered the learning curve of kernel regression, which aims to obtain precise formulas or exact order of the generalization error (both upper and lower bounds) under any choice of the regularization parameter and even any noise level. The learning curve-type results provide a nearly comprehensive picture of an estimator's generalization ability. Note that in order to obtain a reasonable lower bound in the learning curve scenario, we need to assume that the source condition of f^* is exactly s, i.e., $f^* \in [\mathcal{H}]^s$, and $f^* \notin [\mathcal{H}]^r, \forall r > s$. Specific descriptions of this condition were provided in these learning curve papers (see Cui et al. 2021, Eq.(8); or Li et al. 2024b, Assumption 3).

Next, we formally state the learning curve result for kernel gradient flow (from Li et al. 2024a, Theorem 3.1). Under Assumptions 1 and 2, with $\beta > 1, s > 0$, and the embedding index assumption, by choosing $t \asymp n^\theta, \theta > 0$, we have

$$\mathcal{R}(\hat{f}_t) = \begin{cases} \Theta_{\mathbb{P}}\left(n^{-s\theta} + n^{-(1-\theta/\beta)}\right), & \text{if } \theta < \beta, \\ \sigma_0^2 \cdot \tilde{\Omega}_{\mathbb{P}}(1), & \text{if } \theta \geq \beta, \end{cases} \tag{9}$$

where $a_n = \tilde{\Omega}(b_n)$ means $a_n = \Omega((\ln n)^{-p} b_n)$ for any $p > 0$, and $\tilde{\Omega}_{\mathbb{P}}$ is the probability version of $\tilde{\Omega}$. In fact, Li et al. (2024a) addressed all analytic spectral algorithms, including KRR and KGF.

The first line of (9) is the regularized regime, where the two terms correspond to the bias and variance terms, respectively. As a direct corollary, choosing the optimal regularization $t_{\text{opt}} \asymp n^{\frac{\beta}{s\beta+1}}$ leads to the optimal convergence rate of the kernel gradient flow:

$$n^{-\frac{s\beta}{s\beta+1}}, \tag{10}$$

which is also the minimax lower rate of the function space $[\mathcal{H}]^s$. The second line of (9) is the interpolating regime, where the regularization is slight, and the estimator behaves similarly to the kernel interpolation estimator (see Section 2.3.3). The result implies that the generalization ability in the interpolating regime can be arbitrarily bad. Figure 1 visualizes the convergence rates for different values of $\theta > 0 : t \asymp n^\theta$, and source condition $s > 0$. The dashed lines in Figure 1 (a) represent the minimax rates achieved by the optimal regularization t.

Cui et al. (2021) and Li et al. (2024b) also discussed the learning curve of the kernel ridge regression and the impact of the noise level. Compared with the convergence rates of KGF in (9), the difference in KRR is the saturation effect, while KGF never saturates. The saturation

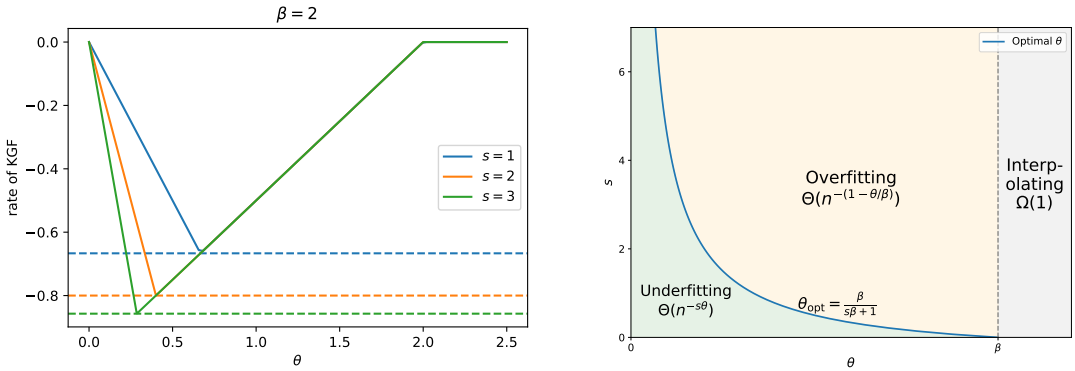

Figure 1: Asymptotic learning curve of kernel gradient flow.

effect, first conjectured by Bauer et al. (2007), says that when $s > 2$, regardless of how carefully the KRR is tuned, the convergence rate cannot be faster than $n^{-\frac{2\beta}{2\beta+1}}$, which is suboptimal compared with the minimax rate $n^{-\frac{s\beta}{s\beta+1}}$.

To end this subsection, we consider an example of the Sobolev RKHS. Assume that $\mathcal{X} \subset \mathbb{R}^d$ is a bounded domain with a smooth boundary and the marginal distribution μ on $\mathcal{X}$ has Lebesgue density $0 < c \leq p(x) \leq C$ for two constants c and C. The (fractional) Sobolev space $W^{m,2}(\mathcal{X})$ is an RKHS if $m > d/2$ (Adams and Fournier, 2003). Denoting $\mathcal{H} = W^{m,2}(\mathcal{X})$, previous results have shown that the decay rate of the eigenvalues of $\mathcal{H}$ is $\beta = 2m/d$ (Edmunds and Triebel, 1996), and $\mathcal{H}$ satisfies the embedding index assumption $\alpha_0 = 1/\beta$. Furthermore, the interpolation space of $\mathcal{H}$ is still a Sobolev space, i.e., $[\mathcal{H}]^p \cong W^{mp,2}(\mathcal{X}), \forall p > 0$. Suppose that the true function $f^* \in W^{r,2}(\mathcal{X})$ for some $r > 0$ and that we use the kernel associated with $\mathcal{H} = W^{m,2}(\mathcal{X})$ to run the kernel gradient flow. Then, the source condition of f^* with respect to $\mathcal{H}$ is $s = r/m$. Thus, we know that the optimal convergence rate of the generalization error is $s\beta/(s\beta + 1) = 2r/(2r + d)$, which is consistent with the minimax rate of $W^{r,2}(\mathcal{X})$.

2.3.3 Kernel interpolation

In practice, neural networks are usually trained to a near-zero training error and have good generalization ability (Belkin et al., 2018; Zhang et al., 2021). This benign overfitting phenomenon of neural networks makes one wonder about the generalization ability of kernel interpolation estimator $\hat{f}_{\text{inter}}$ in (7), which is also the interpolation estimator with the minimum RKHS norm:

$$\hat{f}_{\text{inter}} = \arg\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}, \text{ s.t. } f(\boldsymbol{x}_i) = y_i, \quad i = 1, \ldots, n.$$

In fixed dimensions, several works have claimed the inconsistency of $\hat{f}_{\text{inter}}$ under various settings. Rakhlin and Zhai (2019) showed that $\hat{f}_{\text{inter}}$ with the Laplace kernel is inconsistent when dimension d is odd. Buchholz (2022) proved the inconsistency when k is the kernel function associated with the Sobolev space $W^{m,2}(\mathcal{X}), d/2 < m < 3d/4$. Beaglehole et al. (2023a) showed the inconsistency for a class of shift-invariant periodic kernels under mild spectral assumptions. Li et al. (2023) showed that the generalization ability of kernel interpolation can be arbitrarily bad for those RKHSs that satisfy the embedding index assumption. Lai et al. (2023) showed that kernel interpolation with the neural tangent kernel of a fully connected two-layer ReLU

neural network on one-dimensional data is approximately a linear interpolation. All of these results suggest that kernel interpolation can not generalize well in fixed dimensions.

Beyond inconsistency, Mallinar et al. (2022) and Cheng et al. (2024) discussed the tempered regime (the generalization error remains bounded) and the catastrophic regime (the generalization error diverges to infinity) of kernel interpolation. Haas et al. (2024) found that adding spike components to kernels could lead to consistent or even rate-optimal kernel interpolation in fixed dimensions.

2.4 Kernel regression in high dimensions

Since neural networks often perform well on high-dimensional data, high-dimensional kernel regression has garnered much recent interest. In this subsection, we still use the notations in Section 2.2 and summarize the results when the sample size and dimension satisfy $n \asymp d^\gamma$ for some $\gamma > 0$. Compared with the fixed-dimensional setting, as d varies, the eigenvalues of $\mathcal{H}$ usually depend on d in an unpleasant way. Thus, the polynomial decay rate of the eigenvalues in Assumption 1 must not hold. Most existing results considered specific kernels (e.g., inner product kernel) or special input spaces (e.g., sphere, discrete hypercube), where the eigenvalues and eigenfunctions of $\mathcal{H}$ are well understood. Nevertheless, many new phenomena have emerged in high dimensions.

2.4.1 Polynomial approximation barrier

We first review the "polynomial approximation barrier" of kernel regression studied by Ghorbani et al. (2021) and several subsequent publications (Ghorbani et al., 2020; Mei et al., 2022; Ghosh et al., 2021; Xiao et al., 2022; Hu and Lu, 2022a; Misiakiewicz, 2022; Donhauser et al., 2021; Mei et al., 2021, etc.). This line of work assumed the true function f^* to be square-integrable. Specifically, Ghorbani et al. (2021) considered the inner product kernel on the sphere $\mathcal{X} = \mathbb{S}^{d-1}$ with uniform distribution, defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = h\left(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle\right), \ \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{S}^{d-1}, \tag{11}$$

where $h(z) \in \mathcal{C}^\infty([-1,1])$ is a fixed function independent of d and

$$h(z) = \sum_{j=0}^{\infty} a_j z^j, \ a_j > 0, \ \forall j = 0, 1, 2, \ldots.$$

(This definition of kernel is equivalent to assuming that Assumption 3 of Ghorbani et al. (2021) holds for all levels $\ell \in \mathbb{N}$.) The eigenfunctions of such kernels are spherical harmonic polynomials, and there exists a concise characterization of the order of eigenvalues (see, e.g., Smola et al. 2000 and Lu et al. (2023)). When $n \asymp d^\gamma, \gamma \in (0, \infty) \backslash \mathbb{N}^+$, Theorem 4 of Ghorbani et al. (2021) showed that the generalization error of the kernel ridge regression $\hat{f}_\lambda$ with $\lambda = O(n^{-1})$ satisfies (with high probability)

$$\left| \mathcal{R}(\hat{f}_\lambda) - \|\mathbf{P}_{>\ell} f^*\|_{L^2}^2 \right| \le \varepsilon \left(\|f^*\|_{L^2}^2 + \sigma_0^2 \right),$$

where $\ell = \lfloor \gamma \rfloor$, $\mathbf{P}_{\leq \ell}$ denotes the projection operator that projects to the subspace of polynomials of degree at most ℓ, $\mathbf{P}_{>\ell} = \mathbf{I} - \mathbf{P}_{\leq \ell}$, and ε is a positive real number. The results can be viewed more intuitively as

$$\mathcal{R}(\hat{f}_\lambda) = \|\mathbf{P}_{>\ell} f^*\|_{L^2}^2 + o_{\mathbb{P}}(1).$$

They also showed that $\|\mathbf{P}_{>\ell} f^*\|_{L^2}^2 + o_{\mathbb{P}}(1)$ is the best generalization error achievable by a kernel regression in the form of $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^n a_i k(\boldsymbol{x}, \boldsymbol{x_i}), a_i \in \mathbb{R}$. This polynomial approximation barrier was later used to quantify advantages of feature learning in some literature (see Section 3.3). In the case of $\gamma \in \mathbb{N}^+$, which was not covered by Ghorbani et al. (2021), subsequent works Xiao et al. (2022), Hu and Lu (2022a), and Misiakiewicz (2022) derived the precise asymptotic formulas for the bias and variance terms of the kernel ridge regression and showed that the generalization error achieved the peak when $n \approx d^\gamma/\gamma!, \gamma \in \mathbb{N}^+$. Mei et al. (2022) studied a similar approximation barrier for kernel regression for kernels whose eigenspaces have hypercontractivity and satisfy certain spectral conditions.

The key idea of this line of work is to decompose the empirical kernel matrix $\mathbb{K}(\boldsymbol{X}, \boldsymbol{X})$ into low-frequency and high-frequency parts, then to take advantage of the nice properties of the eigenfunctions (which are spherical harmonic polynomials for the inner kernel of the product on the sphere) when d is large enough.

2.4.2 Generalization behaviors under the source condition

Another line of work considered the source condition (Assumption 2) in the high-dimensional setting (Liu et al., 2021; Zhang et al., 2024b; Lu et al., 2023). Compared with the work mentioned in Section 2.4.1, which only assumes f^* to be square-integrable, $[\mathcal{H}]^s$ is a smaller function space than $L^2(\mathcal{X}, \mu)$ when $s > 0$, and $L^2(\mathcal{X}, \mu)$ can be viewed as a limiting case of $s \to 0$.

Next, we review the results in Zhang et al. (2024b), which also studied the inner product kernel on the sphere (11), provided minimax optimal rates, and found some new phenomena for the kernel ridge regression. Note that two parameters, s and γ, determine the generalization ability. Zhang et al. (2024b) showed that when $s \in (0, 1)$, $s \in [1, 2]$ and $s > 2$, the generalization error behaved differently along γ (Zhang et al. 2024b, Theorems 2 and 3). Specifically, when $0 < s < 1$, the generalization error under the best choice of the regularization parameter (denoted as R^*) has two periods:

(i) if $\gamma \in (p + ps, p + ps + s]$, $p \in \mathbb{N}$,

$$R^* = \Theta_{\mathbb{P}}\left(d^{-\gamma+p}\right) = \Theta_{\mathbb{P}}\left(n^{-1+\frac{p}{\gamma}}\right);$$

(ii) if $\gamma \in (p + ps + s, (p + 1) + (p + 1)s]$, $p \in \mathbb{N}$,

$$R^* = \Theta_{\mathbb{P}}\left(d^{-(p+1)s}\right) = \Theta_{\mathbb{P}}\left(n^{-\frac{(p+1)s}{\gamma}}\right).$$

When $1 \leq s \leq 2$, R^* has three periods:

(i) if $\gamma \in (p+ps,\ p+ps+1]$, $p \in \mathbb{N}$,

$$R^* = \Theta_{\mathbb{P}}\left(d^{-\gamma+p}\right) = \Theta_{\mathbb{P}}\left(n^{-1+\frac{p}{\gamma}}\right);$$

(ii) if $\gamma \in (p+ps+1,\ p+ps+2s-1]$, $p \in \mathbb{N}$,

$$R^* = \Theta_{\mathbb{P}}\left(d^{-\frac{\gamma-p+ps+1}{2}}\right) = \Theta_{\mathbb{P}}\left(n^{-\frac{\gamma-p+ps+1}{2\gamma}}\right).$$

(ii) if $\gamma \in (p+ps+2s-1, (p+1)+(p+1)s]$, $p \in \mathbb{N}$,

$$R^* = \Theta_{\mathbb{P}}\left(d^{-(p+1)s}\right) = \Theta_{\mathbb{P}}\left(n^{-\frac{(p+1)s}{\gamma}}\right).$$

When $s > 2$, R^* is exactly the same as $s = 2$. Furthermore, Zhang et al. (2024b), Theorem 5, provided the corresponding minimax lower rate of $[\mathcal{H}]^s, s > 0$, which is omitted here. For instance, Figure 2 shows the best convergence rates of KRR and the corresponding minimax lower rates (with respect to d) for $s = 1.5$ and $\gamma > 0$. More visualizations of these convergence rates can be found in Zhang et al. (2024b), Figures 2 and 3.

Several interesting phenomena arise from these results.

- Periodic plateau behavior: when γ varies within a certain period, the rate with respect to dimension d does not change with γ.
- Multiple descent behavior: the rates with respect to sample size n achieve peaks and isolated valleys at certain values of γ (Zhang et al. 2024b, Figure 3).
- Minimax optimality and new saturation effect: when $0 < s \leq 1$, the best convergence rate of KRR matches the minimax lower rate for all $\gamma > 0$. When $s > 1$, KRR cannot achieve the minimax lower rate for certain ranges of γ, which is called the new saturation effect of KRR.

The periodic plateau and multiple-descent behaviors were first reported in Lu et al. (2023) for kernel gradient flow with $s = 1$. They also included an example of the neural tangent kernel of the two-layer fully connected ReLU neural network.

2.4.3 Kernel interpolation

We have shown in Section 2.3.3 that kernel interpolation $\hat{f}_{\text{inter}}$ cannot generalize in fixed dimensions. However, kernel interpolation can generalize surprisingly well in high dimensions, as first demonstrated theoretically by Liang and Rakhlin (2020). Liang and Rakhlin (2020) studied the inner product kernel and the setting $n \asymp d$. Using the linear approximation of high-dimensional kernel matrices from Karoui (2010), Liang and Rakhlin (2020) first proved the upper bounds of the bias and variance terms of kernel interpolation, and showed that the generalization error will converge to zero when the data exhibit a low-dimensional structure.

Liang et al. (2020) also considered the inner product kernel and assumed that the coordinates of the d-dimensional input are independent and identically distributed. Considering the

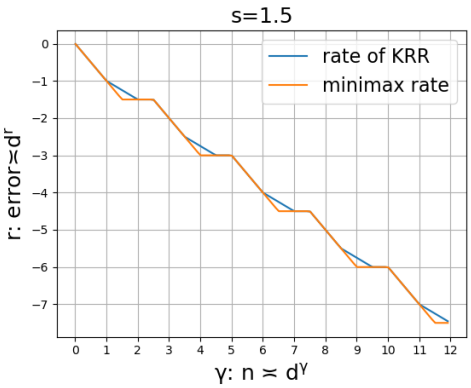

Figure 2: Best convergence rates of KRR and corresponding minimax lower rate (w.r.t. d) for $s = 1.5$ and $\gamma > 0$.

high-dimensional setting $n \asymp d^\gamma, \gamma > 0$, they proved an upper bound of the variance term. Furthermore, assuming $f^*(\boldsymbol{x}) = \langle k(\boldsymbol{x}, \cdot), \rho(\cdot)\rangle_{L^2}$, with $\|\rho\|_{L^4}^4 \leq C$ for some constant $C > 0$, they demonstrated that the bias term is infinitesimal of a higher order compared with the variance term. They obtained the following upper bound for the generalization error with a concrete convergence rate (denote $l = \lfloor \gamma \rfloor$):

$$\mathcal{R}(\hat{f}_{\text{inter}}) = O_{\mathbb{P}}\left(d^{l-\gamma} + d^{\gamma-l-1}\right).$$

This upper bound exhibits multiple-descent behavior, i.e., the convergence rate is non-monotone as γ increases. They also provided visualization and empirical evidence for the multiple-descent behavior (see Liang et al. 2020, Figures 1 and 2).

To our knowledge, Aerni et al. (2022) is the first to provide lower bounds of the bias and variance terms for kernel interpolation in high dimensions. They considered the convolutional kernel on the discrete hypercube $\{-1, 1\}^d$ and a special form of the true function $f^*(x) = x_1 x_2 \cdots x_{L^*}$, where L^* is formulated in Aerni et al. (2022), Theorem 1. Using a similar decomposition of the empirical kernel matrix $\mathbb{K}(\boldsymbol{X}, \boldsymbol{X})$ (as mentioned at the end of Section 2.4.1) as the line of work Ghorbani et al. (2021), this f^* will fall into the eigenspace corresponding to the low-frequency part. They demonstrated that kernel interpolation has generalization ability in this setting and discovered the multiple-descent behavior of kernel interpolation.

Furthermore, Zhang et al. (2024c) showed that whether kernel interpolation is a good choice (in the sense of minimax optimality) depends on the relative smoothness of the true function. Specifically, they considered the inner product kernel on the sphere, $n \asymp d^\gamma, \gamma > 0$, and assumed the source condition of f^* to be exactly $s \geq 0$. For all values of $s \geq 0$ and $\gamma > 0$, they fully characterized the exact orders of the bias and variance terms, leading to an exact order of the generalization error: (denote $l = \lfloor \gamma \rfloor$)

$$\mathcal{R}(\hat{f}_{\text{inter}}) = \Theta_{\mathbb{P}}\left(d^{l-\gamma} + d^{\gamma-l-1} + d^{-(l+1)s}\right).$$

Comparing this rate with the minimax lower rate in $[\mathcal{H}]^s$ in Zhang et al. (2024b), they further showed that for different values of (s, γ), kernel interpolation can be minimax optimal, consistent but sub-optimal, or inconsistent (see Figure 3, borrowed from Zhang et al. 2024c). Specifically,

for any fixed $\gamma \in (0, \infty) \backslash \mathbb{N}^+$, there exists a threshold $\Gamma(\gamma)$ (see Zhang et al. 2024c, Eq.(11)) such that when $s > \Gamma(\gamma)$, kernel interpolation is sub-optimal; when $0 < s \leq \Gamma(\gamma)$, kernel interpolation is minimax optimal; and when $s = 0$ or $\gamma \in \mathbb{N}^+$, kernel interpolation is inconsistent. The existence of a threshold $\Gamma(\gamma)$ provides a comprehensive answer to the question "when does the benign overfitting phenomenon occur" in kernel regression and shows how it depends on the relative smoothness of the true function and the high-dimensional scaling.

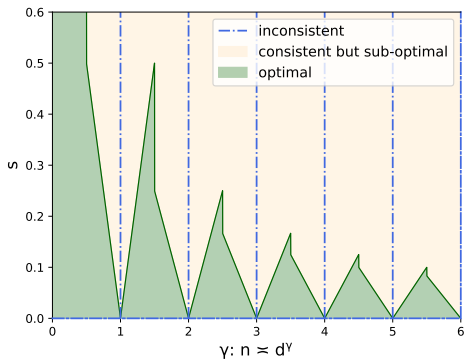

Figure 3: Phase diagram about the consistency and the optimality of kernel interpolation.

2.4.4 Other topics

Recall that most of the results discussed above focused on specific kernels and input spaces. Gavrilopoulos et al. (2024), Barzilai and Shamir (2023), and Misiakiewicz and Saeed (2024) aimed to provide a unified theory of the generalization error of kernel regression under mild assumptions. It is also worth mentioning the study of benign overfitting in high-dimensional linear regression with an arbitrary structure of the covariance of the input data (Bartlett et al., 2020; Tsigler and Bartlett, 2023; Hastie et al., 2022; Kobak et al., 2020, etc.). It can be seen that high-dimensional linear regression and kernel regression share many common characteristics.

3 From fixed kernel/feature to feature learning

3.1 Limitations of fixed kernel/feature

Despite significant successes of the neural tangent kernel (NTK) theory, an increasing number of researchers argue that it does not explain the outstanding performance of real-world neural networks. Notable works highlighting these limitations include, but are not limited to, Wei et al. (2019) and Ghorbani et al. (2019, 2020) from the theoretical perspective; Chizat et al. (2019) and Arora et al. (2019b) from the empirical perspective. The most widely discussed shortcoming of the NTK theory is its lack of feature learning capability, since it equates the training dynamics of neural networks entirely with kernel regression using the specific neural tangent kernel. In the following, we analyze the limitations of the NTK theory from multiple perspectives.

For a kernel function k, Mercer's decomposition (4) can be written as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \Psi(\boldsymbol{x})^\top \boldsymbol{\Lambda} \Psi(\boldsymbol{x}),$$

where we denote $\Psi(\boldsymbol{x}) = (\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \ldots)^\top$ and $\boldsymbol{\Lambda} = \mathrm{Diag}\{\lambda_1, \lambda_2, \ldots\}$. Define the feature map:

$$\boldsymbol{x} \to \Phi(\boldsymbol{x}) = \boldsymbol{\Lambda}^{\frac{1}{2}} \Psi(\boldsymbol{x}). \tag{12}$$

The kernel function is then the inner product of the two features at $\boldsymbol{x}$ and $\boldsymbol{x}'$, i,e.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \Phi(\boldsymbol{x})^\top \Phi(\boldsymbol{x}'). \tag{13}$$

It is well known that kernel regression is equivalent to a linear regression in the feature space. Specifically, (i) the kernel ridge regression estimator in Definition 2.2.1 is the same as the ridge regression estimator in the feature space; (ii) the kernel gradient flow estimator in Definition 2.2.2 can be expressed as $\hat{f}_t(\boldsymbol{x}) = \Phi(\boldsymbol{x})^\top \boldsymbol{b}_t$, where $\boldsymbol{b}_t$ is obtained by a gradient flow to minimize square loss $\mathcal{L}(\boldsymbol{b}) = \|\boldsymbol{Y} - \Phi(\boldsymbol{X})^\top \boldsymbol{b}\|_2^2/n$. Once the architecture of a neural network is determined, the feature corresponding to the NTK is fixed and independent of the data. Therefore, kernel regression with NTK disregards neural networks' ability to learn features from the data.

In Sections 2.3.2 and 2.4.2, the lower bounds for the generalization error indicate that kernel regression (in fixed or high dimensions) cannot outperform the minimax lower bound, even as a point estimator of a specific f^* in the hypothetical function space. The polynomial approximation barrier discussed in Section 2.4.1 demonstrates that kernel regression with inner-product kernels requires at least d^l samples to learn an l-degree polynomial when $\|\mathbf{P}_{>l} f^*\|_{L^2}$ does not vanish. Therefore, if we pre-specify a kernel function before observing the data, kernel regression (or a sufficiently wide neural network in the NTK regime) is unlikely to achieve comparable performance to neural networks in practice.

Additionally, the results in Sections 2.3.3 and 2.4.3 show that kernel interpolation cannot generalize in fixed dimensions and does not outperform regularized kernel regression (unless f^* is less smooth) in high dimensions. This also suggests that kernel regression may not be an appropriate method to interpret "the benign overfitting phenomenon ", which is widely observed for neural networks (Zhang et al., 2021).

In the NTK theory, a special symmetric initialization of the parameters is widely used (Chizat et al., 2019; Hu et al., 2019; Lai et al., 2023), ensuring that the output of a neural network at initialization is zero, e.g., $f^m_{\boldsymbol{\theta}(0)}(\boldsymbol{x}) \equiv 0$ in (3). Recall that the explicit expression of the KGF estimator (6) also relies on the zero initialization $\hat{f}_0(\boldsymbol{x}) \equiv 0$. Recent work (Chen et al., 2024) showed that, in the NTK regime, nonzero initialization of a neural network would introduce a terrible bias. Specifically, they showed that a sufficiently wide and fully connected neural network with each parameter initialized as an independent standard normal variable achieves an optimal convergence rate of generalization error $n^{-\frac{3}{d+3}}$, no matter how smooth f^* is (in the sense of the source condition $f^* \in [\mathcal{H}]^s$). This convergence rate falls into the "curse of dimensionality": As the dimensionality d increases, the sample size n required for good generalization ability increases exponentially. Thus, they suggested that there is a divergence between the NTK theory and real-world neural networks.

3.2 Implications of a fixed kernel

Although we have discussed the limitations of fixed kernel/feature theory in explaining real-world neural networks, it does offer insights into what constitutes a good kernel/feature. In this subsection, as a warm-up of the adaptive feature approach, we review two examples to illustrate a key observation:

The alignment between the true function and the chosen feature matters.

More precisely, for a given kernel feature map $\Phi(\boldsymbol{x})$ (defined in (12)) and the corresponding kernel k, we say that the feature function is more aligned with the true function f^* if " the projections $\langle f^*, \psi_j \rangle_{L^2}$ concentrate more on the eigenspaces corresponding to larger eigenvalues λ_j."

3.2.1

Consider the following example of a one-dimensional regression problem: suppose that $\mathcal{X} = [0, 1]$ with uniform distribution. Consider a true function $f^*(x) = \sin(2\pi x)$, and apply kernel regression with the following two kernels:

$$k_1(x, x') = \mathbf{1}_{\{x \geq x'\}}(1 - x)x' + \mathbf{1}_{\{x < x'\}}(1 - x')x,$$

and

$$k_2(x, x') = \min\{x, x'\}.$$

The results of Wainwright (2019) and Li et al. (2022) give rise to explicit formulas for the eigenvalues and eigenfunctions of k_1, k_2, i.e., for $j = 1, 2, \ldots$

$$\lambda_{1,j} = \frac{1}{\pi^2 j^2}, \quad \psi_{1,j} = \sqrt{2}\sin(j\pi x),$$

and

$$\lambda_{2,j} = \frac{4}{\pi^2(2j-1)^2}, \quad \psi_{2,j}(x) = \sqrt{2}\sin\left(\frac{2j-1}{2}\pi x\right).$$

Therefore, k_1, k_2 have the same decay rates of eigenvalues, $\beta_1 = \beta_2 = 2$. Recalling the definition of an interpolation space $[\mathcal{H}]^s$ in (8), since the projection $\langle f^*, \psi_{1,j} \rangle$ is nonzero only when $j = 2$, f^* has the source condition $s_1 = \infty$ with respect to k_1. Since the projections $\langle f^*, \psi_{2,j} \rangle \asymp j^{-2}$, f^* has the source condition $s_2 = 1.5$ with respect to k_2. Using the results (10) in Section 2.3.2, we obtain the optimal convergence rates as $r_1 = n^{-1}, r_2 = n^{-0.75}$, respectively, when using k_1, k_2 to estimate f^*. This implies that the feature Φ_1 (or kernel k_1), which aligns more with f^*, is a better choice than Φ_2 (or k_2).

3.2.2

The second example, from Arora et al. (2019a), considered the two-layer fully connected ReLU neural network. Denote the noiseless labels as $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top$, and the empirical NTK matrix $\mathbb{K} \in \mathbb{R}^{n \times n}$ as $\mathbb{K}_{ij} = k(\boldsymbol{x}_j, \boldsymbol{x}_j), i, j = 1, \ldots, n$, where k is the NTK of the fully connected two-layer ReLU neural network. Then under certain assumptions (the width of the hidden layer grows

sufficiently fast, and the neural network is trained for a sufficiently long time), the generalization error of the neural network estimator $\hat{f}_{\mathrm{NN}}$ has an upper bound:

$$\mathcal{R}(\hat{f}_{\mathrm{NN}}) = O_{\mathbb{P}}\left(\sqrt{\frac{\boldsymbol{Y}^\top \mathbb{K}^{-1}\boldsymbol{Y}}{n}}\right). \tag{14}$$

Let the eigenvalues and eigenvectors of the matrix $\mathbb{K}$ be $\hat{\lambda}_j, \hat{\boldsymbol{v}}_j$, i.e., $\mathbb{K} = \sum_{j=1}^n \hat{\lambda}_j \hat{\boldsymbol{v}}_j \hat{\boldsymbol{v}}_j^\top$, we have

$$\boldsymbol{Y}^\top \mathbb{K}^{-1}\boldsymbol{Y} = \sum_{j=1}^n \frac{1}{\hat{\lambda}_j}\left(\hat{\boldsymbol{v}}_j^\top \boldsymbol{Y}\right)^2. \tag{15}$$

$\hat{\lambda}_j, \hat{\boldsymbol{v}}_j$ are empirical versions of the eigenvalues and eigenfunctions of k. Also note that they considered the noiseless case, so $\boldsymbol{Y}$ is an empirical version of the true function f^*. Therefore, (14) and (15) imply that projections $\langle f^*, \psi_j\rangle_{L^2}$ corresponding to small eigenvalues λ_j must be small for a good generalization ability.

3.3 Recent advances in feature learning

The aim of the adaptive feature approach in the next section is to theoretically analyze the feature learning process of neural networks and its advantages. Before that, we summarize some of the recent advances in feature learning.

There is a line of work empirically studying the evolution of the alignment between labels and features during the training of neural networks (Oymak et al., 2019; Kopitkov and Indelman, 2020; Fort et al., 2020; Maennel et al., 2020; Shan and Bordelon, 2021; Baratin et al., 2021; Atanasov et al., 2021; Ortiz-Jiménez et al., 2021, etc.). Some of these papers considered an index called "(centered) kernel alignment" (Cortes et al., 2012; Kornblith et al., 2019) under various settings and observed that the alignment increases as training progresses.

Chizat et al. (2019), Geiger et al. (2020), Woodworth et al. (2020), and Yang and Hu (2020); Bordelon and Pehlevan (2022) demonstrated that the scale of initialization can induce feature learning even for infinite-width neural networks. In these scenarios, feature learning refers to the phenomenon that the parameters of neural networks evolve non-trivially and cannot be considered approximately unchanged, which is in contrast to the frozen feature in the neural tangent kernel (NTK) regime.

Theoretically, characterizing what feature can be learned from the data and analyzing its impact on the method's generalization ability is a challenge. A notable line of work dealt with the problem by considering the one-step gradient descent setting (Ba et al., 2022; Damian et al., 2022; Dandi et al., 2023; Moniri et al., 2023; Cui et al., 2024). These works originated from the random feature model (Rahimi and Recht, 2007; Gerace et al., 2020; Mei and Montanari, 2022; Hu and Lu, 2022b), etc., and aimed to prove that feature learning brought by one-step gradient descent can lead to advantages over random features and kernel regression estimators. Next, we briefly introduce the results in Ba et al. (2022). They considered the following two-layer fully

connected neural network

$$f_{\mathrm{NN}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(\boldsymbol{w}_r^\top \boldsymbol{x}\right) = \frac{1}{\sqrt{m}} \boldsymbol{a}^\top \sigma\left(\boldsymbol{W}^\top \boldsymbol{x}\right), \tag{16}$$

and the proportional asymptotic limits, i.e., the width m, sample size n and sample dimension d all tend to infinity and satisfy

$$n/d \to \gamma_1, m/d \to \gamma_2, \quad \gamma_1, \gamma_2 \in (0, \infty).$$

Denoting $\boldsymbol{W}_0$ as the weight matrix at initialization, they first updated $\boldsymbol{W}_0$ for one step:

$$\boldsymbol{W}_1 = \boldsymbol{W}_0 + \eta\sqrt{m} \cdot \boldsymbol{G}_0,$$

where $\boldsymbol{G}_0$ is the gradient of the square loss with respect to $\boldsymbol{W}_0$, and η is the learning rate. Then, a ridge regression was conducted in the new feature space $\boldsymbol{x} \to \sigma\left(\boldsymbol{W}_1^\top \boldsymbol{x}\right)$. They also assumed that the true function was a single index function $f^*(\boldsymbol{x}) = \sigma^*(\langle \boldsymbol{x}, \boldsymbol{\beta}_* \rangle)$, where $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $\|\boldsymbol{\beta}_*\|_2 = 1$. Denote $\mathcal{R}_0(\lambda)$ and $\mathcal{R}_1(\lambda)$ as the generalization errors of the ridge regression estimators with the regularization parameter $\lambda > 0$ using the features $\sigma\left(\boldsymbol{W}_0^\top \boldsymbol{x}\right)$ and $\sigma\left(\boldsymbol{W}_1^\top \boldsymbol{x}\right)$, respectively. Under some detailed assumptions, by choosing the learning rate $\eta = \Theta(1)$, they proved that (Ba et al. 2022, Theorem 5):

$$\mathcal{R}_0(\lambda) - \mathcal{R}_1(\lambda) \xrightarrow{\mathbb{P}} \delta \geq 0,$$

where δ is constant with explicit expression in their paper. For this constant learning rate $\eta = \Theta(1)$, one-step gradient descent has already shown an improvement over the initial random feature. Furthermore, by choosing a larger learning rate $\eta = \Theta(\sqrt{m})$, they proved an upper bound of $\mathcal{R}_1(\lambda)$ (Ba et al. 2022, Theorem 7), which outperforms the kernel lower bound $\|\mathbf{P}_{>1} f^*\|_{L^2}^2$ (Hu and Lu 2022b; Montanari and Zhong 2022, the same as the polynomial approximation barrier in Section 2.4.1) for some examples.

Something more interesting was also studied about the first step gradient $\boldsymbol{G}_0$ and the new feature $\boldsymbol{W}_1$. They showed that $\boldsymbol{G}_0$ is close to a rank-1 matrix $\boldsymbol{X}^\top \boldsymbol{Y} \boldsymbol{a}^\top / n\sqrt{m}$ (omitting the constant), where $\boldsymbol{a}$ is the weights of the randomly initialized output layer, and $\boldsymbol{X}^\top \boldsymbol{Y}$ is roughly $\boldsymbol{\beta}_*$ in their true function. Furthermore, they showed that the first singular vector $\boldsymbol{u}_1$ of $\boldsymbol{W}_1$ (corresponding to the leading singular value) satisfies the following:

$$|\langle \boldsymbol{u}_1, \boldsymbol{\beta}_* \rangle|^2 \xrightarrow{\mathbb{P}} C_0,$$

where the expression $C_0 \in (0, 1)$ is provided in their paper. It can be seen from their expression that $C_0 \to 1$ when $\gamma_1 \to \infty$, and C_0 increases as the learning rate increases. Despite many specific assumptions, they theoretically proved that the alignment (similar to the alignment discussed at the beginning of Section 3.2) between the feature and labels emerges from the training of neural networks, and showed the advantage of feature learning.

There is a substantial body of literature studying feature learning, which is too extensive to list comprehensively here. For instance, Hanin and Nica (2019), Dyer and Gur-Ari (2019),

Huang and Yau (2020), Yaida (2020), Naveh and Ringel (2021), and Bordelon and Pehlevan (2024) studied the finite-width corrections which enhanced feature evolving. Toward analyzing feature learning with fewer assumptions on the true function, Radhakrishnan et al. (2022), Beaglehole et al. (2023b), Beaglehole et al. (2024), and Radhakrishnan et al. (2024) proposed a general structure of the weights of neural networks during training, which was called the neural feature ansatz.

4 Adaptive feature model and over-parameterized Gaussian sequence

4.1 Neural networks as adaptive feature model

We have demonstrated in Sections 3.1 and 3.2 that a kernel regression is equivalent to a linear regression in the corresponding feature space, and the alignment between the feature and the true function affects the generalization ability. Conventional neural networks can be naturally viewed as linear regressions in the feature space, where the feature can be learned during training. We still assume that the output is one-dimensional. Denote the weights of the last layer as $\boldsymbol{\beta} \in \mathbb{R}^m$, and the feature before the last layer as $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x}) \in \mathbb{R}^m$. Then, the output of neural networks can be written as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \Phi_{\boldsymbol{\eta}}(\boldsymbol{x})^\top \boldsymbol{\beta}, \tag{17}$$

where $\boldsymbol{\theta} = \mathrm{vec}(\{\boldsymbol{\beta}, \boldsymbol{\eta}\})$ represents the learnable parameters of the neural network. For example, in (16), $\boldsymbol{\beta} = \boldsymbol{a}$, and $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x}) = \sigma(\boldsymbol{W}^\top \boldsymbol{x})/\sqrt{m}$ with $\boldsymbol{\eta} = \boldsymbol{W}$.

Although the parameterization of $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$ varies with the specific architecture of the neural network, which can be extremely complicated, we can abstractly summarize the estimator (17) as a more general *adaptive feature model*.

Definition 4.1.1 (Adaptive feature model)**.** *Consider the nonparametric regression problem (1). For a learnable feature $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x}) \in \mathbb{R}^m$ with parameters $\boldsymbol{\eta}$, the adaptive feature estimator is defined as*

$$f_t(\boldsymbol{x}) = \Phi_{\boldsymbol{\eta}_t}(\boldsymbol{x})^\top \boldsymbol{\beta}_t,$$

and $\boldsymbol{\eta}_t, \boldsymbol{\beta}_t$ are obtained by the gradient flow (at time t) minimizing the loss:

$$\mathcal{L}(\boldsymbol{\eta}, \boldsymbol{\beta}) = \left\| \boldsymbol{Y} - \Phi_{\boldsymbol{\eta}}(\boldsymbol{X})^\top \boldsymbol{\beta} \right\|_2^2. \tag{18}$$

The feature dimension can be finite or infinite. By analogy to (13), we can define the corresponding time-varying kernel function as

$$k_{\boldsymbol{\eta}}(\boldsymbol{x}, \boldsymbol{x}') := \Phi_{\boldsymbol{\eta}}(\boldsymbol{x})^\top \Phi_{\boldsymbol{\eta}}(\boldsymbol{x}').$$

The adaptive feature model in Definition 4.1.1 is an intuitive illustration of feature learning of neural networks, but the training dynamic in (18) is still difficult to study. The difficulties include but are not limited to:

(i) The parameterization of $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$ with respect to $\boldsymbol{\eta}$ is usually complicated;

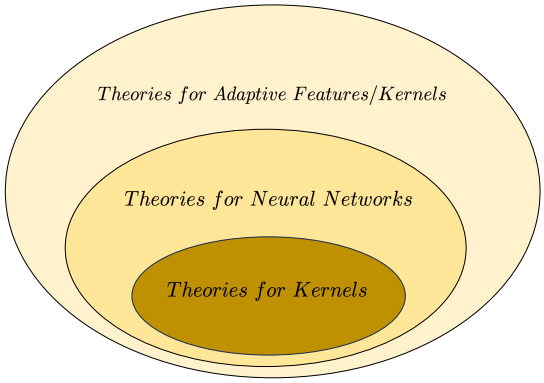

Figure 4: The inclusion relationships of the adaptive feature theories, neural networks theories and kernel regression theories.

(ii) The evolution of $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$ and $\boldsymbol{\beta}$ during training is coupled together.

The neural tangent kernel (NTK) theory addresses the first difficulty by assuming that the width is sufficiently large, ensuring that the feature remains approximately unchanged during training, i.e., $\Phi_{\boldsymbol{\eta}_t}(\boldsymbol{x}) \approx \Phi_{\boldsymbol{\eta}_0}(\boldsymbol{x})$. The line of work Ba et al. (2022), Damian et al. (2022), Dandi et al. (2023), Moniri et al. (2023), and Cui et al. (2024) (discussed in Section 3.3) simplifies the second difficulty by studying the one-step gradient descent. As described in the following subsections, our objective is to propose an alternative approach to study the training dynamic in (18), while preserving its feature learning characteristics of the neural network as much as possible. Figure 4 illustrates the inclusion relationships among the adaptive feature theories, neural network theories, and kernel regression (as a sufficiently wide neural network) theories.

At the end of this subsection, we present experiments to demonstrate that the feature learned by a neural network ($\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$ in (17)) indeed aligns with the data gradually along with training. Recall that in Section 3.2, we claim that a better feature vector $\Phi(\boldsymbol{x}) = \boldsymbol{\Lambda}^{\frac{1}{2}}\Psi(\boldsymbol{x})$ should satisfy that: "the projections $\langle f^*, \psi_j\rangle_{L^2}$ concentrate more on the eigenspaces corresponding to larger eigenvalues λ_j." Denote $\hat{\boldsymbol{v}}_j \in \mathbb{R}^n$ as the j-th singular vector (singular values arranged in descending order) of the neural network's feature $\Phi_{\boldsymbol{\eta}}(\boldsymbol{X})$, thus $\sqrt{n}\hat{\boldsymbol{v}}_j$ serves as an approximation of the j-th eigenfunction of the feature $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$. Denote $\boldsymbol{Y} \in \mathbb{R}^n$ as the labels of the n samples. Then, in our experiments, we use $f_j := \frac{1}{\sqrt{n}}\hat{\boldsymbol{v}}_j^\top \boldsymbol{Y}$ as an approximation to $\langle f^*, \psi_j\rangle_{L^2}$. Figure 5 shows the percentage of the first p projections $\sum_{j=1}^p f_j^2 / \sum_{j=1}^m f_j^2$ (p = 1, 100 or 300) after each iteration t. Here, m represents the dimension of $\Phi_{\boldsymbol{\eta}}(\boldsymbol{x})$ and we choose $m = 500$ in the experiments. We observe that the percentage increases as the training proceeds, indicating that the alignment between the neural network's feature and the data improves. In our experiments, we used a fully connected two-layer neural network (FCN) on the MNIST dataset and a three-layer convolutional neural network (CNN) on the CIFAR-10 dataset. For more details on the experiments, see Appendix.

4.2 Relation between kernel regression and the Gaussian sequence model

We have introduced the adaptive feature model and the difficulties in studying its training dynamic. In Section 4.3, we will describe the *over-parameterized Gaussian sequence model* as a feasible prototype for the adaptive feature model. We view the adaptive feature model as an

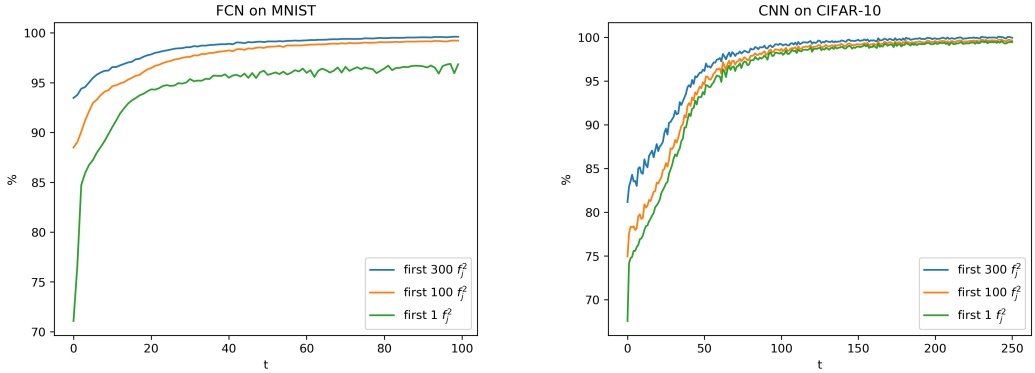

Figure 5: x-axis: number of training iterations; y-axis: the percentage of the first p projections: $\sum_{j=1}^{p} f_j^2 / \sum_{j=1}^{m} f_j^2$, p = 1, 100, 300, m = 500. The projections f_j concentrate on top eigenspaces as the training proceeds.

adaptive version of the kernel regression (fixed feature) model, and view the "over-parameterized Gaussian sequence" model as an adaptive version of the *Gaussian sequence model*.

Let the kernel function be $k(\boldsymbol{x}, \boldsymbol{x}') = \Psi(\boldsymbol{x})^\top \boldsymbol{\Lambda} \Psi(\boldsymbol{x}')$, where $\Psi(\boldsymbol{x}) = (\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \ldots)^\top$ is the vector of eigenfunctions that forms an orthonormal basis of $L^2(\mathcal{X}, \mu)$, and $\boldsymbol{\Lambda} = \mathrm{Diag}\{\lambda_1, \lambda_2, \ldots\}$ is the diagonal matrix with the corresponding eigenvalues (in decreasing order). We denote the true function as $f^*(\boldsymbol{x}) = \sum_{j=1}^{\infty} \theta_j^* \psi_j(\boldsymbol{x})$. We further rescale the parameters taking into account the eigenvalues: let $\beta_j^* = \lambda_j^{-\frac{1}{2}} \theta_j^*$. Thus, $f^*(\boldsymbol{x}) = \sum_{j=1}^{\infty} \lambda_j^{\frac{1}{2}} \beta_j^* \psi_j(\boldsymbol{x})$. In the following, let $\beta_{j,t}$ be an estimate of β_j^*, and let $\theta_{j,t} = \lambda_j^{\frac{1}{2}} \beta_{j,t}$ be the corresponding estimate of θ_j^*, for $j = 1, 2, \ldots$.

Kernel gradient flow with kernel k (also known as a linear regression in the feature space) applies the gradient flow of $\boldsymbol{\beta}_t = (\beta_{1,t}, \beta_{2,t}, \ldots)^\top$ to minimize the square loss:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\beta}) &= \frac{1}{2n} \left\| \boldsymbol{Y} - \Psi(\boldsymbol{X})^\top \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{\beta} \right\|_2^2, \\ \text{i.e.,} \quad \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}_t &= \nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}_t), \end{aligned} \tag{19}$$

where $\Psi(\boldsymbol{X}) \in \mathbb{R}^{\infty \times n}$, and use $\hat{f}_t = \Psi(\boldsymbol{X})^\top \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{\beta}_t$ as an estimate of f^*.

In the corresponding Gaussian sequence model, we observe a noisy sequence $\{z_j\}_{j=1}^{\infty}$ generated from the following model:

$$z_j = \theta_j^* + \xi_j, \ j \in \mathbb{N}^+; \ \xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, n^{-1}), \tag{20}$$

where $\{\theta_j^*\}_{j=1}^{\infty}$ are the true parameters, given by the projections of f^* onto the eigenfunctions $\langle f^*, \psi_j \rangle_{L^2}$. We consider the gradient flow of $\boldsymbol{\beta}_t = (\beta_{1,t}, \beta_{2,t}, \ldots)^\top$ to minimize the square loss:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\beta}) &= \frac{1}{2} \sum_{j=1}^{\infty} \left(z_j - \lambda_j^{\frac{1}{2}} \beta_j \right)^2, \\ \text{i.e.,} \quad \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}_t &= \nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}_t), \end{aligned} \tag{21}$$

and use $\theta_{j,t} = \lambda_j^{\frac{1}{2}} \beta_{j,t}$ as the estimate of θ_j^*.

We hypothesize that there is a strong equivalence between the kernel gradient flow (19) (for estimating f^*) and the corresponding gradient flow for the Gaussian sequence model (21) (for estimating $\{\theta_j^*\}_{j=1}^\infty$). We here provide supporting evidences in two aspects (Sections 4.2.1 and 4.2.2).

4.2.1 Similar training dynamics

When applying gradient flow, the kernel regression model and the Gaussian sequence model exhibit similar training dynamics.

First, in kernel gradient flow, simple calculations of (19) show that

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\beta}_t = -\nabla_{\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta}_t) = -\frac{1}{n}\boldsymbol{\Lambda}^{\frac{1}{2}}\Psi(\boldsymbol{X})\Psi(\boldsymbol{X})^\top\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\beta}_t + \frac{1}{n}\boldsymbol{\Lambda}^{\frac{1}{2}}\Psi(\boldsymbol{X})\boldsymbol{Y}.$$

As the sample size n increases, it roughly holds that

- $\frac{1}{n}\Psi(\boldsymbol{X})\Psi(\boldsymbol{X})^\top \approx \mathbf{I}$;
- $\left[\frac{1}{n}\Psi(\boldsymbol{X})\boldsymbol{Y}\right]_j \approx \lambda_j^{\frac{1}{2}}\beta_j^* + \tilde{\epsilon}_j, \quad \tilde{\epsilon}_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, n^{-1})$.

Thus, the gradient flow of $\boldsymbol{\beta}_t$ can be approximated by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\beta_{j,t} = -\lambda_j\beta_{j,t} + \lambda_j^{\frac{1}{2}}\left(\lambda_j^{\frac{1}{2}}\beta_j^* + \tilde{\epsilon}_j\right), \ \forall j \in \mathbb{N}^+. \tag{22}$$

Second, in the Gaussian sequence model, it is easy to show that, for any $j = 1, 2, \ldots$, $\beta_{j,t}$ in (21) satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}\beta_{j,t} = -\lambda_j\beta_{j,t} + \lambda_j^{\frac{1}{2}}\left(\lambda_j^{\frac{1}{2}}\beta_j^* + \xi_j\right). \tag{23}$$

Since we assume that $\xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, n^{-1})$ in (20), the dynamics (22) and (23) are the same. Approximately, the two gradient flows (19) and (21) are also the same.

4.2.2 Similar convergence rates

In Section 2.3.2, we show the convergence rate results of the kernel gradient flow under the eigenvalue decay rate and the source condition assumptions. In this subsection, we demonstrate that under equivalent assumptions in the Gaussian sequence model, the gradient flow in the Gaussian sequence model (23) (or (21)) has the same generalization error convergence rates.

Solving equation (23) with zero initialization $\beta_{j,0} = 0, j = 1, 2, \ldots$ yields

$$\beta_{j,t} = \lambda_j^{-\frac{1}{2}}(1 - e^{-\lambda_j t})z_j.$$

Using $\theta_{j,t} := \lambda_j^{\frac{1}{2}}\beta_{j,t} = (1 - e^{-\lambda_j t})z_j$ as the estimate of θ_j^*, we obtain the square loss in the Gaussian sequence model (20) as (denote $\boldsymbol{\theta}_t, \boldsymbol{\theta}^*$ as the vectors of $\theta_{j,t}, \theta_j^*$)

$$\mathbb{E}\,\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 = \sum_{j=1}^\infty (e^{-\lambda_j t}\theta_j^*)^2 + \frac{1}{n}\sum_{j=1}^\infty (1 - e^{-\lambda_j t})^2.$$

We adopt a similar assumption for the eigenvalue decay rate (EDR): $\lambda_j \asymp j^{-\beta}$, and the assumption of the source condition: $\sum_{j=1}^{\infty} \lambda_j^{-s}(\theta_j^*)^2 < \infty$ and $\sum_{j=1}^{\infty} \lambda_j^{-r}(\theta_j^*)^2 = \infty$ for any $r > s$. Then, simple calculations show that, when $t = n^{\alpha}, \alpha > 0$, we have (here β denotes the EDR)

$$\mathbb{E}\left\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\right\|_2^2 = \begin{cases} \Theta\left(n^{-s\alpha} + n^{-(1-\alpha/\beta)}\right), & \text{if } \alpha < \beta, \\ \Omega\left(1\right), & \text{if } \alpha \geq \beta. \end{cases}$$

These convergence rates are identical to the learning curve results of the kernel gradient flow in (9).

4.3 Over-parameterized Gaussian sequence model

4.3.1 Applying the previous equivalence to different features/ kernels

In Section 4.2, we discuss the hypothesized equivalence between the kernel regression model and the Gaussian sequence model for a fixed kernel k, or equivalently, fixed eigenfunctions $\Psi(\boldsymbol{x})$, eigenvalues $\boldsymbol{\Lambda}$, and feature $\Phi(\boldsymbol{x}) = \boldsymbol{\Lambda}^{\frac{1}{2}}\Psi(\boldsymbol{x})$.

What if we use a different kernel, denoted as $\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') = \tilde{\Psi}(\boldsymbol{x})^\top \tilde{\boldsymbol{\Lambda}} \tilde{\Psi}(\boldsymbol{x}') = \tilde{\Phi}(\boldsymbol{x})^\top \tilde{\Phi}(\boldsymbol{x}')$, with a new feature $\tilde{\Phi}(\boldsymbol{x})$? We denote the true function $f^*(\boldsymbol{x}) = \sum_{j=1}^{\infty} \lambda_j^{\frac{1}{2}} \beta_j^* \psi_j(\boldsymbol{x}) = \sum_{j=1}^{\infty} \theta_j^* \psi_j(\boldsymbol{x})$ as in Section 4.2, and denote $\boldsymbol{X}, \boldsymbol{Y}$ as the data from the nonparametric regression model (1). Denote the expansion of f^* with respect to the new eigenfunctions and eigenvalues as

$$f^*(\boldsymbol{x}) = \sum_{j=1} \tilde{\lambda}_j^{\frac{1}{2}} \tilde{\beta}_j^* \tilde{\psi}_j(\boldsymbol{x}) = \sum_{j=1} \tilde{\theta}_j^* \tilde{\psi}_j(\boldsymbol{x}).$$

Then, the new kernel gradient flow estimator is obtained by the gradient flow of $\tilde{\boldsymbol{\beta}}_t$ minimizing the square loss:

$$\mathcal{L}(\tilde{\boldsymbol{\beta}}) = \frac{1}{2n} \left\|\boldsymbol{Y} - \tilde{\Psi}(\boldsymbol{X})^\top \tilde{\boldsymbol{\Lambda}}^{\frac{1}{2}} \tilde{\boldsymbol{\beta}}\right\|_2^2. \tag{24}$$

Let $\tilde{z}_j, j = 1, 2, \ldots$ be sampled from the model:

$$\tilde{z}_j = \tilde{\lambda}_j^{\frac{1}{2}} \tilde{\beta}_j^* + \tilde{\xi}_j, \quad \tilde{\xi}_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, n^{-1}). \tag{25}$$

The discussion in Section 4.2.1 shows that the dynamic of $\tilde{\beta}_t$ minimizing (24) are identical to the dynamic of $\tilde{\beta}_t$ minimizing the following loss:

$$\mathcal{L}(\tilde{\boldsymbol{\beta}}) = \frac{1}{2} \sum_{j=1}^{\infty} \left(\tilde{z}_j - \tilde{\lambda}_j^{\frac{1}{2}} \tilde{\beta}_j\right)^2. \tag{26}$$

Furthermore, denote the orthogonal matrix $\boldsymbol{A} \in \mathbb{R}^{\infty \times \infty}$ such that $\tilde{\Psi}(\boldsymbol{x}) = \boldsymbol{A}^\top \Psi(\boldsymbol{x})$, the diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{\infty \times \infty}$ such that $\tilde{\boldsymbol{\Lambda}} = \boldsymbol{D}\boldsymbol{\Lambda}$, $\boldsymbol{Z} = (z_1, z_2, \ldots)^\top$ and $\tilde{\boldsymbol{Z}} = (\tilde{z}_1, \tilde{z}_2, \ldots)^\top$. By the definition of $\boldsymbol{Z}$ in (20) and $\tilde{\boldsymbol{Z}}$ in (25) (both derived from the same function f^* but with different orthonormal basis $\Psi(\boldsymbol{x}), \tilde{\Psi}(\boldsymbol{x})$), roughly, we have

$$\tilde{\boldsymbol{Z}} = \boldsymbol{A}^\top \boldsymbol{Z}.$$

Therefore, the loss (26) can be rewritten as:

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{\beta}}) &= \frac{1}{2}\left\|\tilde{\boldsymbol{Z}} - \tilde{\boldsymbol{\Lambda}}^{\frac{1}{2}}\tilde{\boldsymbol{\beta}}\right\|_2^2 = \frac{1}{2}\left\|\boldsymbol{A}^\top \boldsymbol{Z} - \boldsymbol{D}\boldsymbol{\Lambda}^{\frac{1}{2}}\tilde{\boldsymbol{\beta}}\right\|_2^2 \\ &= \frac{1}{2}\left\|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{D}\boldsymbol{\Lambda}^{\frac{1}{2}}\tilde{\boldsymbol{\beta}}\right\|_2^2. \end{aligned} \tag{27}$$

In other words, (27) leads to a key observation:

> *Replacing the kernel/feature with a new kernel/feature in kernel gradient flow is supposed to be equivalent to multiplying a corresponding matrix $\boldsymbol{AD}$ in the corresponding gradient flow of the Gaussian sequence model.*

4.3.2 Over-parameterized Gaussian sequence model

Building on the observation in Section 4.3.1, we are now ready to propose the transformation from the adaptive feature model in Definition 4.1.1 to the following *over-parameterized Gaussian sequence model.*

Definition 4.3.1 (Over-parameterized Gaussian sequence model)**.** *Consider the model:*

$$z_j = \theta_j^* + \xi_j, \quad \xi_j \sim \mathcal{N}(0, n^{-1}), \quad j = 1, \dots N,$$

where $\boldsymbol{\theta}^* := (\theta_1^*, \dots, \theta_N^*)^\top \in \ell^2$ are the true parameters, $\boldsymbol{Z} = (z_1, \dots, z_N)^\top$ are the observed samples, and N can be either finite or infinite. Then, the over-parameterized Gaussian sequence model estimates $\boldsymbol{\theta}^*$ by applying the gradient flow of $\boldsymbol{A}_t \in \mathbb{R}^{N\times N}, \boldsymbol{D}_t \in \mathbb{R}^{N\times N}$, and $\boldsymbol{\alpha}_t \in \mathbb{R}^N$ to minimize the loss:

$$\begin{aligned} &\mathcal{L}(\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{\alpha}) = \|\boldsymbol{Z} - \boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha}\|_2^2, \\ \text{i.e.,}\quad &\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{A}_t = \nabla_{\boldsymbol{A}_t}\mathcal{L};\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{D}_t = \nabla_{\boldsymbol{D}_t}\mathcal{L};\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\alpha}_t = \nabla_{\boldsymbol{\alpha}_t}\mathcal{L}, \end{aligned} \tag{28}$$

with $\boldsymbol{A}_0$ initialized as a certain orthogonal matrix, $\boldsymbol{D}_0$ initialized as a certain diagonal matrix with non-negative elements, and $\boldsymbol{\alpha}_0 = \boldsymbol{0}$. We keep $\boldsymbol{A}_t$ as an orthogonal matrix and $\boldsymbol{D}_t$ as a diagonal matrix during gradient flow.

We propose the over-parameterized Gaussian sequence model as a prototype for the adaptive feature model in Definition 4.1.1. Intuitively, the adaptive feature model allows the kernel, or equivalently, the eigenfunctions and eigenvalues to change during training. Based on the observation in Section 4.3.1, learnable eigenfunctions and eigenvalues correspond to the learnable matrices $\boldsymbol{A}_t$ (orthogonal), $\boldsymbol{D}_t$ (diagonal) in the over-parameterized Gaussian sequence model. Note that we incorporate the fixed eigenvalues $\boldsymbol{\Lambda}$ in Section 4.3.1 into $\boldsymbol{D}_t$ in Definition 4.3.1. If we set $\boldsymbol{A}_t \equiv \boldsymbol{A}$ and $\boldsymbol{D}_t \equiv \boldsymbol{D}\boldsymbol{\Lambda}^{\frac{1}{2}}$, the gradient flow of the over-parameterized Gaussian sequence model becomes the gradient flow to minimize the loss (27).

Compared with the vanilla gradient flow in (21), "over-parameterization" refers to the extra learnable parameters $\boldsymbol{A}_t$ and $\boldsymbol{D}_t$ in the gradient flow of (28). Theoretically, we only need the learnable parameters $\boldsymbol{\beta}_t$ in (21) (or $\boldsymbol{\alpha}_t$ in (28)) to generate a consistent estimator of $\boldsymbol{\theta}^*$ as $n \to \infty$. We expect that additional learnable parameters can bring more flexibility to the gradient flow,

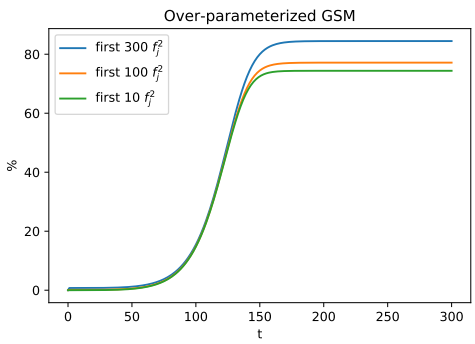

Figure 6: x-axis: number of training iterations; y-axis: the percentage of the first p projections: $\sum_{j=1}^{p} f_{j,t}^2 / \sum_{j=1}^{N} f_{j,t}^2$, $p = 10, 100, 300, N = 500$. The projections f_j concentrate on top eigenspaces as the training proceeds.

allowing it to adapt to latent structures in $\boldsymbol{\theta}^*$ and achieve a better generalization ability than the vanilla gradient flow.

Next, we provide some intuitions about roles of $\boldsymbol{A}_t$ and $\boldsymbol{D}_t$. As in Section 3.2, we hope that projections of the true parameters $\boldsymbol{u}_j^\top \boldsymbol{\theta}^*$ (where $\boldsymbol{u}_j$ is the j-th column vector of $\boldsymbol{A}_t$) concentrate more on the directions corresponding to larger λ_j (where λ_j is the j-th diagonal element of $\boldsymbol{D}_t$). Furthermore, if $\boldsymbol{u}_j^\top \boldsymbol{\theta}^*$ is negligible or even zero for some j, the corresponding λ_j should be close to zero as much as possible to reduce variance. Therefore, $\boldsymbol{A}_t$ and $\boldsymbol{D}_t$ can be viewed as adjustments of eigenvectors and eigenvalues, respectively. Ideal $\boldsymbol{A}_t$ and $\boldsymbol{D}_t$ should ensure that: the projections of $\boldsymbol{\theta}^*$ concentrate mainly on a small sub-column space of $\boldsymbol{A}_t$; the diagonal elements in $\boldsymbol{D}_t$ corresponding to this subspace are as large as possible; and other elements in $\boldsymbol{D}_t$ as close to zero as possible.

4.3.3 Simulation

We provide a simulation of the over-parameterized Gaussian sequence model in Definition 4.3.1. We initialize $\boldsymbol{D}_0 = \mathrm{Diag}\{\lambda_1^{\frac{1}{2}}, \ldots, \lambda_N^{\frac{1}{2}}\}$, $\boldsymbol{A}_0 = \mathbf{I}_N$ and use discrete-time gradient descent to approximate the gradient flow in (28). Specifically, we set $N = 500, n = 4000$ and choose a learning rate of 0.5. We set $\theta_j^* = 1/(N - j + 2)$ and $\lambda_j = 1/(j+5)^2$ for $j = 1, \ldots, N$ in our simulation. Note that $\{\lambda_j\}_{j=1}^N$ are decreasing while $\{\theta_j^*\}_{j=1}^N$ are increasing.

For $j = 1, 2, \ldots, N$, denote j' such that $\nu_{j',t}$ is the j-th largest diagonal element of $\boldsymbol{D}_t$ and denote $\boldsymbol{u}_{j',t}$ as the j'-th column vector of $\boldsymbol{A}_t$. Let $f_{j,t} := \boldsymbol{u}_{j',t}^\top \boldsymbol{\theta}^*$. With these notations, $f_{j,t}$ is the projection on the eigenvector that corresponds to the j-th largest diagonal element in $\boldsymbol{D}_t$.

Since $\boldsymbol{A}_0 = \mathbf{I}_N$, $\boldsymbol{D}_0 = \mathrm{Diag}\{\lambda_1, \ldots, \lambda_N\}$, and $f_{j,0} = \theta_j^*$ at initialization, the data are misaligned with the feature at initialization (i.e., θ_j^* are smaller for those larger λ_j). In the over-parameterized Gaussian sequence model, we expect that as the gradient descent proceeds, the projections $f_{j,t}$ will concentrate on the eigenspaces corresponding to larger diagonal elements of $\boldsymbol{D}_t$. This is observed in Figure 6, where the percentage of the first p projections $\sum_{j=1}^{p} f_{j,t}^2 / \sum_{j=1}^{N} f_{j,t}^2$ ($p = 10, 100$ or 300) increases as the number of training iterations increases.

4.4 Over-parameterized models in related literature

Li and Lin (2024) considered a data generation model similar to that in Definition 4.3.1:

$$z_j = \theta_j^* + \xi_j, \quad j = 1, 2, \ldots,$$

where the noise $\{\xi_j\}_{j=1}^{\infty}$ are ϵ^2-sub-Gaussian. They used $\theta_{j,t} := a_{j,t}\beta_{j,t}$ as the estimates of θ_j^*, and applied the gradient flow of $\boldsymbol{a}_t = (a_{1,t}, a_{2,t}, \ldots)^\top, \boldsymbol{\beta}_t = (\beta_{1,t}, \beta_{2,t}, \ldots)^\top$ to minimize the loss:

$$\mathcal{L}(\boldsymbol{a}, \boldsymbol{\beta}) = \frac{1}{2}\sum_{j=1}^{\infty}(z_j - a_j\beta_j)^2. \tag{29}$$

They initialized $a_{j,0} = \lambda_j^{\frac{1}{2}}$ for a sequence $\lambda_j \asymp j^{-\beta}, \beta > 1$, and let $\beta_{j,0} = 0, j = 1, 2, \ldots$. Denote $\boldsymbol{\theta}_t, \boldsymbol{\theta}^*$ as the vectors of $\theta_{j,t}, \theta_j^*$, respectively.

Furthermore, they compared the generalization error of the gradient flow (29) with the vanilla gradient flow (21), where $\boldsymbol{a}_t$ does not evolve and remains at $a_{j,t} \equiv \lambda_j^{\frac{1}{2}}, j = 1, 2, \ldots$. Corollary 3.3 of Li and Lin (2024) provides a concrete example where the gradient flow (29) has a faster generalization error convergence rate than the vanilla gradient flow (21).

Recalling Definition 4.3.1, if we let $N = \infty$, $\boldsymbol{A}_t \equiv \mathbf{I}$, $\boldsymbol{D}_t = \mathrm{Diag}\{a_{1,t}, a_{2,t}, \ldots\}$, then the gradient flow of (29) in Li and Lin (2024) is a special case of our over-parameterized Gaussian sequence model. Specifically, the gradient flow of (29) only allows the eigenvalues ($\boldsymbol{D}_t$) to be adjustable, whereas the gradient flow in Definition 4.3.1 allows both the eigenvectors and eigenvalues to be adjustable. It is reasonable to expect our over-parameterized Gaussian sequence model to achieve greater improvements in generalization ability.

Another line of research studies the advantages of overparameterization in high-dimensional linear regression problems with sparse signals (Vaskevicius et al., 2019; Li et al., 2021; Zhao et al., 2022). For example, Zhao et al. (2022) considered applying the gradient descent of $\boldsymbol{g}_t, \boldsymbol{l}_t \in \mathbb{R}^d$ to minimize the square loss:

$$\mathcal{L}(\boldsymbol{g}, \boldsymbol{l}) = \frac{1}{2n}\|\boldsymbol{Y} - \boldsymbol{X}^\top(\boldsymbol{g} \circ \boldsymbol{l})\|_2^2,$$

where "$\circ$" is the Hadamard product operator (element-wise product). They showed that the over-parameterized version of gradient descent could lead to a near-minimax optimal rate and an even faster dimension-free convergence rate in the strong signal case.

5 Discussion

Theoretically analyzing the feature learning characteristics of neural network models is a challenging task. The primary difficulty lies in the complex training dynamics of these models. Although existing results have made valuable progress (Section 3.3), fully understanding feature learning and its benefits for generalization ability still requires considerable effort. This paper introduces the over-parameterized Gaussian sequence model (Definition 4.3.1) as a viable paradigm to address this problem. This paradigm borrows the concept of a good feature from fixed kernel regression theory and retains feature learning characteristics of neural networks as

adaptive feature learners.

We hope that the over-parameterized Gaussian sequence model can offer some insights into how feature learning occurs and its associated benefits, thus facilitating the study of feature learning of real-world neural networks. To this end, we need to investigate (but not limited to) the following questions in the future:

- Theoretical understanding of the over-parameterized Gaussian sequence model in Definition 4.3.1. Specifically, the first step is to characterize the alignment between $\boldsymbol{A}_t, \boldsymbol{D}_t$ and the true parameters as the gradient flow progresses. The next step is to analyze how the alignment enhances the generalization ability (compared with vanilla gradient flow, or models without over-parameterization). We can begin with a simple family of feature maps and gradually increase their complexity. For example, Li and Lin (2024) considered the simplest case where the eigenvalues are adjustable. In addition, one can consider the case where $\boldsymbol{A}_t$ varies among a family of orthogonal matrices, making the eigenvectors adjustable.

- Rigorous equivalence between the (adaptive feature) kernel regression model and the (over-parameterized) Gaussian sequence model, especially in the high dimensions. We have shown some evidence of the equivalence in Section 4.2.2, but the rigorous proof and the high-dimensional case remain open problems. This equivalence will be a crucial step in extending the results of the over-parameterized Gaussian sequence model to the adaptive feature model.

- The specific architecture of neural networks (e.g., the number of layers, fully connected or convolutional neural networks, etc.) will inevitably affect the manner and efficiency of feature learning. Characterizing the differences between various neural network architectures using our paradigm will be a more advanced problem.

References

Robert A Adams and John JF Fournier. *Sobolev Spaces*. Elsevier, 2003.

Michael Aerni, Marco Milanta, Konstantin Donhauser, and Fanny Yang. Strong inductive biases provably prevent harmless interpolation. In *The Eleventh International Conference on Learning Representations*, 2022.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019b.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2021.

Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems*, 35:37932–37946, 2022.

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, pages 2269–2277. PMLR, 2021.

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.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

Daniel Barzilai and Ohad Shamir. Generalization in kernel regression under realistic assumptions. *arXiv preprint arXiv:2312.15995*, 2023.

Benedikt Bauer and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4):2261–2285, 2019.

F. Bauer, S. Pereverzyev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of complexity*, 23(1):52–72, 2007.

Daniel Beaglehole, Mikhail Belkin, and Parthe Pandit. On the inconsistency of kernel ridgeless regression in fixed dimensions. *SIAM Journal on Mathematics of Data Science*, 5(4):854–872, 2023a.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in convolutional neural networks. *arXiv preprint arXiv:2309.00570*, 2023b.

Daniel Beaglehole, Ioannis Mitliagkas, and Atish Agarwala. Gradient descent induces alignment between weights and the empirical ntk for deep non-linear networks. *arXiv preprint arXiv:2402.05271*, 2024.

Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:3617641.

G. Blanchard and Nicole Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18:971–1013, 2018.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *ICML*, 2020.

Simon Buchholz. Kernel interpolation in sobolev spaces is not consistent in low dimensions. In *Conference on Learning Theory*, pages 3410–3440. PMLR, 2022.

Andrea Caponnetto. Optimal rates for regularization operators in learning theory. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE COMPUTER SCIENCE AND ARTIFICIAL . . . , 2006.

Andrea Caponnetto and Ernesto de Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

Alain Celisse and Martin Wahl. Analyzing the discrepancy principle for kernelized spectral filter learning algorithms. *J. Mach. Learn. Res.*, 22:76:1–76:59, 2020.

Guhan Chen, Yicheng Li, and Qian Lin. On the impacts of the random initialization in the neural tangent kernel theory. *arXiv preprint arXiv:2410.05626*, 2024. URL https://arxiv.org/pdf/2410.05626.

Tin Sum Cheng, Aurelien Lucchi, Anastasis Kratsios, and David Belius. Characterizing overfitting in kernel ridgeless regression through the eigenspectrum. *arXiv preprint arXiv:2402.01297*, 2024.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Felipe Cucker and Stephen Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2001. URL https://api.semanticscholar.org/CorpusID:8188805.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. *Advances in Neural Information Processing Systems*, 34:10131–10143, 2021.

Hugo Cui, Luca Pesce, Yatin Dandi, Florent Krzakala, Yue M Lu, Lenka Zdeborová, and Bruno Loureiro. Asymptotics of feature learning in two-layer networks after one gradient-step. *arXiv preprint arXiv:2402.04980*, 2024.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pages 5413–5452. PMLR, 2022.

Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*, 2023.

Aymeric Dieuleveut and Francis Bach. Nonparametric stochastic approximation with large step-sizes1. *THE ANNALS*, 44(4):1363–1399, 2016.

Konstantin Donhauser, Mingqi Wu, and Fanny Yang. How rotational invariance of common kernels prevents generalization in high dimensions. In *International Conference on Machine Learning*, pages 2804–2814. PMLR, 2021.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. In *International Conference on Learning Representations*, 2019.

D. E. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge Tracts in Mathematics. Cambridge University Press, 1996. doi: 10.1017/CBO9780511662201.

Jianqing Fan, Cong Ma, and Yiqiao Zhong. A selective overview of deep learning. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 36(2):264, 2021.

Simon-Raphael Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *Journal of Machine Learning Research*, 21:205:1–205:38, 2020.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

Georgios Gavrilopoulos, Guillaume Lecué, and Zong Shang. A geometrical analysis of kernel ridge regression and its applications. *arXiv preprint arXiv:2404.07709*, 2024.

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Generalisation error in learning with random features and the hidden manifold model. In *International Conference on Machine Learning*, pages 3452–3462. PMLR, 2020.

L. Lo Gerfo, Lorenzo Rosasco, Francesca Odone, E. De Vito, and Alessandro Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. *Advances in Neural Information Processing Systems*, 32, 2019.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems*, 33:14820–14830, 2020.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029 – 1054, 2021. doi: 10.1214/20-AOS1990. URL https://doi.org/10.1214/20-AOS1990.

Nikhil Ghosh, Song Mei, and Bin Yu. The three stages of learning dynamics in high-dimensional kernel methods. In *International Conference on Learning Representations*, 2021.

Moritz Haas, David Holzmüller, Ulrike Luxburg, and Ingo Steinwart. Mind the spikes: Benign overfitting of kernels and neural networks in fixed dimension. *Advances in Neural Information Processing Systems*, 36, 2024.

Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949 – 986, 2022. doi: 10.1214/21-AOS2133. URL https://doi.org/10.1214/21-AOS2133.

Hong Hu and Yue M Lu. Sharp asymptotics of kernel ridge regression beyond the linear regime. *arXiv preprint arXiv:2205.06798*, 2022a.

Hong Hu and Yue M Lu. Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory*, 69(3):1932–1964, 2022b.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv preprint arXiv:1905.11368*, 2019.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *International conference on machine learning*, pages 4542–4551. PMLR, 2020.

Kaixuan Huang, Yuqing Wang, Molei Tao, and Tuo Zhao. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in neural information processing systems*, 33:2698–2709, 2020.

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

Noureddine El Karoui. The spectrum of kernel random matrices. *The Annals of Statistics*, 38 (1):1 – 50, 2010. doi: 10.1214/08-AOS648. URL https://doi.org/10.1214/08-AOS648.

Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez. The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *Journal of Machine Learning Research*, 21(169):1–16, 2020.

Michael Kohler and Adam Krzyżak. Nonparametric regression estimation using penalized least squares. *IEEE Trans. Inf. Theory*, 47:3054–3059, 2001.

Dmitry Kopitkov and Vadim Indelman. Neural spectrum alignment: Empirical study. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part II 29*, pages 168–179. Springer, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

Jianfa Lai, Manyun Xu, Rui Chen, and Qian Lin. Generalization ability of wide neural networks on $\mathbb{R}$. *arXiv preprint arXiv:2302.05933*, 2023.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Jiangyuan Li, Thanh Nguyen, Chinmay Hegde, and Ka Wai Wong. Implicit sparse regularization: The impact of depth and early stopping. *Advances in Neural Information Processing Systems*, 34:28298–28309, 2021.

Yicheng Li and Qian Lin. Improving adaptivity via over-parameterization in sequence models. *arXiv preprint arXiv:2409.00894*, 2024.

Yicheng Li, Haobo Zhang, and Qian Lin. On the saturation effect of kernel ridge regression. In *The Eleventh International Conference on Learning Representations*, 2022.

Yicheng Li, Haobo Zhang, and Qian Lin. Kernel interpolation generalizes poorly. *Biometrika*, page asad048, 08 2023. ISSN 1464-3510. doi: 10.1093/biomet/asad048. URL https://doi.org/10.1093/biomet/asad048.

Yicheng Li, Weiye Gan, Zuoqiang Shi, and Qian Lin. Generalization error curves for analytic spectral algorithms under power-law decay. *arXiv preprint arXiv:2401.01599*, 2024a.

Yicheng Li, Qian Lin, et al. On the asymptotic learning curves of kernel ridge regression under power-law decay. *Advances in Neural Information Processing Systems*, 36, 2024b.

Yicheng Li, Zixiong Yu, Guhan Chen, and Qian Lin. On the eigenvalue decay rates of a class of neural-network related kernel functions defined on general domains. *Journal of Machine Learning Research*, 25(82):1–47, 2024c.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "Ridgeless" regression can generalize. *The Annals of Statistics*, 48(3):1329 – 1347, 2020. doi: 10.1214/19-AOS1849. URL https://doi.org/10.1214/19-AOS1849.

Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory*, pages 2683–2711. PMLR, 2020.

Junhong Lin and Volkan Cevher. Optimal convergence for distributed learning with stochastic gradient methods and spectral algorithms. *Journal of Machine Learning Research*, 21:147–1, 2020.

Junhong Lin, Alessandro Rudi, L. Rosasco, and V. Cevher. Optimal rates for spectral algorithms with least-squares regression over Hilbert spaces. *Applied and Computational Harmonic Analysis*, 48:868–890, 2018.

Fanghui Liu, Zhenyu Liao, and Johan Suykens. Kernel regression in high dimensions: Refined analysis beyond double descent. In *International Conference on Artificial Intelligence and Statistics*, pages 649–657. PMLR, 2021.

Weihao Lu, Haobo Zhang, Yicheng Li, Manyun Xu, and Qian Lin. Optimal rate of kernel regression in large dimensions. *arXiv preprint arXiv:2309.04268*, 2023.

Hartmut Maennel, Ibrahim M Alabdulmohsin, Ilya O Tolstikhin, Robert Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. What do neural networks learn when trained with random labels? *Advances in Neural Information Processing Systems*, 33:19693–19704, 2020.

Neil Mallinar, James B Simon, Amirhesam Abedsoltan, Parthe Pandit, Mikhail Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: A taxonomy of overfitting. *arXiv preprint arXiv:2207.06569*, 2022.

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.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pages 3351–3418. PMLR, 2021.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2021.12.003. URL https://www.sciencedirect.com/science/article/pii/S1063520321001044.

Theodor Misiakiewicz. Spectrum of inner-product kernel matrices in the polynomial regime and multiple descent phenomenon in kernel ridge regression. *arXiv preprint arXiv:2204.10425*, 2022.

Theodor Misiakiewicz and Basil Saeed. A non-asymptotic theory of kernel ridge regression: deterministic equivalents, test error, and gcv estimator. *arXiv preprint arXiv:2403.08938*, 2024.

Behrad Moniri, Donghwan Lee, Hamed Hassani, and Edgar Dobriban. A theory of nonlinear feature learning with one gradient step in two-layer neural networks. *arXiv preprint arXiv:2310.07891*, 2023.

Andrea Montanari and Yiqiao Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. *The Annals of Statistics*, 50(5):2816–2847, 2022.

Gadi Naveh and Zohar Ringel. A self consistent theory of gaussian processes captures feature learning effects in finite cnns. *Advances in Neural Information Processing Systems*, 34:21352–21364, 2021.

Atsushi Nitanda and Taiji Suzuki. Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. In *International Conference on Learning Representations*, 2020.

Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. What can linearized neural networks actually say about generalization? *Advances in Neural Information Processing Systems*, 34:8998–9010, 2021.

Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. *Advances in Neural Information Processing Systems*, 31, 2018.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. *arXiv preprint arXiv:2212.13881*, 2022.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Alexander Rakhlin and Xiyu Zhai. Consistency of interpolation with laplace kernels is a high-dimensional phenomenon. In *Conference on Learning Theory*, pages 2595–2623. PMLR, 2019.

Lorenzo Rosasco, Ernesto De Vito, and Alessandro Verri. Spectral methods for regularization in learning theory. *DISI, Universita degli Studi di Genova, Italy, Technical Report DISI-TR-05-18*, 2005.

Yoshihiro Sawano. *Theory of Besov spaces*, volume 56. Springer, 2018.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *ArXiv*, abs/1708.06633, 2017.

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

Alex Smola, Zoltán Ovári, and Robert C. Williamson. Regularization with dot-product kernels. *Advances in neural information processing systems*, 13, 2000.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

Ingo Steinwart and C. Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35(3):363–417, 2012.

Ingo Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *COLT*, pages 79–93, 2009.

Namjoon Suh and Guang Cheng. A survey on statistical theory of deep learning: Approximation, training dynamics, and generative models. *arXiv preprint arXiv:2401.07187*, 2024.

Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2018.

Luc Tartar. *An introduction to Sobolev spaces and interpolation spaces*, volume 3. Springer Science & Business Media, 2007.

Tom Tirer, Joan Bruna, and Raja Giryes. Kernel-based smoothness analysis of residual networks. In *Mathematical and Scientific Machine Learning*, pages 921–954. PMLR, 2022.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. *Advances in Neural Information Processing Systems*, 32, 2019.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

Wenjia Wang and Bing-Yi Jing. Gaussian process regression: Optimality, robustness, and relationship with kernel ridge regression. *Journal of Machine Learning Research*, 23(193): 1–67, 2022. URL http://jmlr.org/papers/v23/21-0570.html.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, pages 3635–3673. PMLR, 2020.

L Xiao, H Hu, T Misiakiewicz, Y Lu, and J Pennington. Precise learning curves and higher-order scaling limits for dot product kernel regression. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Sho Yaida. Non-gaussian processes and neural networks at finite widths. In *Mathematical and Scientific Machine Learning*, pages 165–192. PMLR, 2020.

Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

Haobo Zhang, Yicheng Li, Weihao Lu, and Qian Lin. On the optimality of misspecified kernel ridge regression. In *International Conference on Machine Learning*, pages 41331–41353. PMLR, 2023.

Haobo Zhang, Yicheng Li, and Qian Lin. On the optimality of misspecified spectral algorithms. *Journal of Machine Learning Research*, 25(188):1–50, 2024a.

Haobo Zhang, Yicheng Li, Weihao Lu, and Qian Lin. Optimal rates of kernel ridge regression under source condition in large dimensions. *arXiv preprint arXiv:2401.01270*, 2024b.

Haobo Zhang, Weihao Lu, and Qian Lin. The phase diagram of kernel interpolation in large dimensions. *Biometrika*, page asae057, 11 2024c. ISSN 1464-3510. doi: 10.1093/biomet/asae057. URL https://doi.org/10.1093/biomet/asae057.

Peng Zhao, Yun Yang, and Qiao-Chu He. High-dimensional linear regression via implicit regularization. *Biometrika*, 109(4):1033–1046, 2022.

Experiments details

We provide some details of the neural networks experiments in Section 4.1.

- The fully connected neural network (for MNIST). Input dimension = 784; first hidden layer dimension = 500; second hidden layer dimension = 500; readout layer dimension = 1.
- The convolutional neural network (for CIFAR-10). Input shape = $3 * 32 * 32$; first convolutional layer: 64 channels, $5 * 5$ kernel, dropout = 0.5, max pooling; second convolutional layer: 256 channels, $5 * 5$ kernel, dropout = 0.5, average pooling; third convolutional layer: 32 channels, $5*5$ kernel, dropout = 0.5, no pooling; a fully connected layer: dimension=500; readout layer: dimension = 1.

We use ReLU activation, MSE loss, batch size of 32 and a learning rate of 0.001. We train neural networks in all 60000 samples of MNIST or CIFAR-10 and use 10000 of these samples to calculate $f_j := \frac{1}{\sqrt{n}} \hat{\boldsymbol{v}}_j^\top \boldsymbol{Y}$ (i.e., $n = 10000$) in Section 4.1.