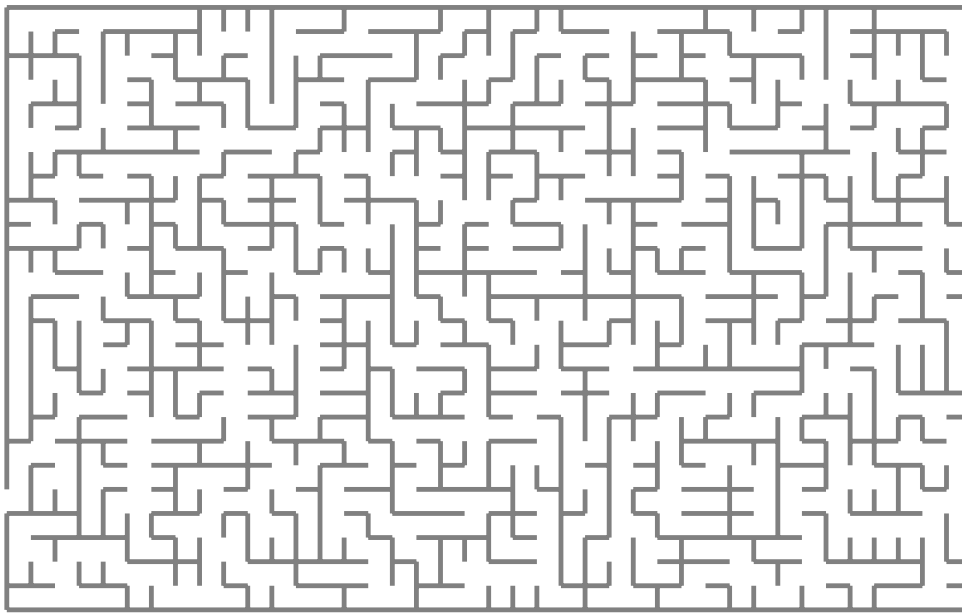
Training and Generalization in Overparameterized Neural Networks

Version of January 4, 2026



Shreyas Kalvankar

Training and Generalization in Overparameterized Neural Networks

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Shreyas Kalvankar
born in Nashik, India

Pattern Recognition & Bioinformatics Group
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

© 2026 Shreyas Kalvankar. *Note that this notice is for demonstration purposes and that the LaTeX style and document source are free to use as basis for your MSc thesis.*

Cover picture: A “random” maze generated in postscript.

Training and Generalization in Overparameterized Neural Networks

Author: Shreyas Kalvankar
Student id: 6255191
Email: skalvankar@tudelft.nl

Abstract

This document describes the standard thesis style for the Software Engineering department at Delft University of Technology. The document and it's source are an example of the use of the standard LaTeX style file. In addition the final appendix to this document contains a number of requirements and guidelines for writing a Software Engineering MSc thesis.

Your thesis should either employ this style or follow it closely.

Thesis Committee:

Advisor:	Dr. D. M. J. Tax, Faculty EEMCS, TU Delft
Supervisor:	Dr. F. Bartolucci, Faculty EEMCS, TU Delft
Daily Supervisor:	Dr. A. Heinlein, Faculty EEMCS, TU Delft
Committee Member:	Dr. S.T.A.F.F. Member, Faculty EEMCS, TU Delft

Preface

This is where you thank people for helping you etc.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus massa pede, feugiat sit amet, mollis in, sodales at, augue. Ut sit amet nisi egestas risus consequat adipiscing. Nulla non diam. Proin volutpat, lacus quis volutpat scelerisque, leo urna rhoncus arcu, vel ultrices dui lacus id lorem. Nam pulvinar adipiscing odio. Etiam tellus lorem, malesuada in, scelerisque sit amet, consequat a, tellus. Curabitur non urna. Mauris facilisis tempor nulla. Nam euismod semper massa. Nullam id nulla. Duis mattis nunc ut ipsum. Proin libero purus, posuere ut, tincidunt sit amet, accumsan sit amet, nisl. Integer commodo. Pellentesque suscipit, diam vel bibendum interdum, magna mauris venenatis lorem, vitae tristique nibh lacus convallis velit. Sed tellus. Mauris placerat lectus ut tellus rutrum blandit. Aliquam erat volutpat.

Suspendisse potenti. Proin sodales eros non lacus. Nam magna sapien, tristique ut, hendrerit ultricies, pretium ut, ante. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla facilisi. In libero risus, pellentesque vitae, interdum id, tincidunt ut, sapien. Mauris nec massa sit amet leo dictum pretium. Curabitur iaculis euismod mauris. Donec diam sem, pulvinar at, luctus id, blandit nec, pede. Nam scelerisque sollicitudin nunc. Nam malesuada mauris id ligula. Donec suscipit posuere justo. Mauris sed libero in mi nonummy tincidunt.

Shreyas Kalvankar
Delft, the Netherlands
January 4, 2026

Contents

List of Figures

Chapter 1

Introduction

Deep neural networks have achieved remarkable empirical success across a wide range of application domains, most notably in computer vision and natural language processing. In vision, deep convolutional and residual architectures have led to dramatic improvements on large-scale benchmarks, establishing deep learning as the dominant paradigm for visual recognition tasks (Krizhevsky et al., 2012; He et al., 2016). Similarly, in natural language processing, attention-based architectures introduces by Vaswani et al. (2023) such as Transformers have enabled substantial advances in sequence modeling and representation learning, and now form the foundation of modern large-scale language models. Over the past decade, this empirical progress has been accompanied by a steady increase in model size, depth, and computational scale, often resulting in highly overparameterized models that generalize well despite their capacity to fit random labels (Zhang et al., 2017; Belkin et al., 2019). Much of this progress has been driven by empirical experimentation and architectural intuition rather than by a complete theoretical understanding of the mechanisms underlying training and generalization in deep neural networks.

Despite their practical success, the mechanisms governing the training dynamics and convergence behavior of neural networks remain only partially understood, even in highly simplified settings. From an optimization perspective, neural network training involves high-dimensional, non-convex objectives whose geometry depends intricately on architectural choices, initialization, and optimization dynamics. Early analyses of neural network loss landscapes have highlighted the prevalence of saddle points and complex critical structures, underscoring the difficulty of directly characterizing training dynamics in parameter space Choromanska et al. (2015). While more recent theoretical work has made progress in analyzing overparameterized models under restrictive assumptions such as two-layer networks or specific scaling regimes these results do not yet provide a unified explanation of neural network training in general settings Arora et al. (2019). More broadly, this gap between empirical performance and theoretical understanding has motivated the study of neural networks through simplified models and asymptotic limits, where training dynamics can be analyzed more precisely.

1.1 Problem setup

Empirical risk minimization. A standard way to formalize supervised learning is through the principle of empirical risk minimization (ERM). Given a training dataset $\{(x_i, y_i)\}_{i=1}^n$, a parametric model f_θ, and a loss function $\ell(\cdot,\cdot)$, ERM seeks parameters

$$\theta^\star \in \arg\min_\theta \frac{1}{n}\sum_{i=1}^n \ell(f_\theta(x_i), y_i)\,. \tag{1.1}$$

This formulation isolates the key ingredients that determine the behavior of learning algorithms: the function class induced by the parameterization $\theta \mapsto f_\theta$, the geometry of the loss landscape, and the optimization procedure used to minimize the empirical risk. In modern neural network training, the empirical risk is typically minimized using gradient-based methods, which induce a dynamical system in parameter space whose properties depend intricately on both the model architecture and the chosen loss function.

Linear models and gradient descent. To build intuition, consider the least-squares empirical risk minimization problem with a one layer linear model $f_w(x) = x^\top w$, $w \in \mathbb{R}^d$. Given a dataset $\{(x_i, y_i)\}_{i=1}^n$, let $X \in \mathbb{R}^{d\times n}$ denote the data matrix whose columns are x_i, and let $y \in \mathbb{R}^n$ denote the vector of labels. The empirical risk is

$$\mathcal{L}(w) = \frac{1}{2}\,\|X^\top w - y\|_2^2. \tag{1.2}$$

This objective is convex and differentiable, with gradient

$$\nabla_w \mathcal{L}(w) = X(X^\top w - y) = XX^\top w - Xy. \tag{1.3}$$

At a stationary point $w^\star$, the gradient vanishes, yielding the equations

$$XX^\top w^\star = Xy, \tag{1.4}$$

which characterize the set of global minimizers of $\mathcal{L}$.
The structure of the minimizers depends on the rank of the empirical covariance operator $A := XX^\top$. If A is positive definite (equivalently, if X has full row rank), then the minimizer is unique and given by $w^\star = A^{-1}Xy$. In contrast, in the overparameterized regime where $\operatorname{rank}(X) < d$, the matrix A is singular and the least-squares objective admits infinitely many global minimizers. In this case, the Moore–Penrose pseudoinverse can be used to approximate a solution. Since A is symmetric, its pseudoinverse A^+ acts as the inverse on $\operatorname{range}(A)$ and annihilates $\ker(A)$, yielding the minimum-norm least-squares solution $w^\star = A^+Xy$.
Gradient descent with step size $\eta > 0$ takes the explicit form

$$w_{t+1} = w_t - \eta\nabla_w \mathcal{L}(w_t) = \left(I - \eta XX^\top\right) w_t + \eta Xy. \tag{1.5}$$

Assuming $w_0 = 0$, we can write a closed form solution

$$w_t = \eta\sum_{s=0}^{t-1}\left(I - \eta XX^\top\right)^s Xy. \tag{1.6}$$

Let $\lambda_{\max}(A)$ denote the largest eigenvalue of A. If the step size satisfies $\eta \in (0, 2/\lambda_{\max}(A))$, then for any least-squares minimizer $w^\star$ the error $e_t := w_t - w^\star$ evolves as

$$e_{t+1} = (I - \eta A)e_t. \tag{1.7}$$

Since A is symmetric, it admits an eigendecomposition

$$A = Q\Lambda Q^\top,$$

where $Q \in \mathbb{R}^{d\times d}$ is orthogonal ($Q^\top Q = I$) and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ contains the (real) eigenvalues of A. Moreover, A is positive semidefinite because $v^\top A v = \|X^\top v\|_2^2 \geq 0$ for all v, hence $\lambda_i \geq 0$ and in particular $\lambda_i \in [0, \lambda_{\max}(A)]$.

Starting from the error recursion

$$e_{t+1} = (I - \eta A)e_t,$$

we express the error in the eigenbasis of A by defining $\tilde{e}_t := Q^\top e_t$. Left-multiplying by $Q^\top$ and using $e_t = Q\tilde{e}_t$ gives

$$\tilde{e}_{t+1} = Q^\top (I - \eta A) Q \tilde{e}_t = \big(I - \eta Q^\top A Q\big)\tilde{e}_t = \big(I - \eta\Lambda\big)\tilde{e}_t.$$

Since Λ is diagonal, this yields the component-wise dynamics

$$\tilde{e}_{t+1,i} = (1 - \eta\lambda_i)\tilde{e}_{t,i}, \qquad i = 1, \ldots, d. \tag{1.8}$$

If $\eta \in (0, 2/\lambda_{\max}(A))$, then for any $\lambda_i > 0$ we have $0 < \eta\lambda_i \leq \eta\lambda_{\max}(A) < 2$, hence $-1 < 1 - \eta\lambda_i < 1$, i.e. $|1 - \eta\lambda_i| < 1$. Under the stated condition on η, we have $|1 - \eta\lambda_i| < 1$ for all $\lambda_i > 0$, implying geometric decay of all components of e_t in $\mathrm{range}(A)$. In particular, if A is positive definite, then $w_t \to w^\star$ as $t \to \infty$. More generally, if $w_0 \in \mathrm{range}(A)$ (for example, $w_0 = 0$), then w_t converges to the minimum-norm least-squares solution $w^\star = A^+ X y$.

The eigen-decomposition above shows that gradient descent acts as a linear dynamical system whose behavior is governed by the spectrum of the empirical covariance operator $XX^\top$. Directions corresponding to larger eigenvalues converge more rapidly, while components in the null space of $XX^\top$ remain unchanged. In the overparameterized regime, where multiple *interpolating solutions* exist, i.e., solutions satisfying $X^\top w = y$, gradient descent from standard initializations converges to a particular interpolating solution, namely the minimum-norm solution. This illustrates how the optimization algorithm induces an implicit bias even in this simplest linear setting.

Deep linear network. Even in the absence of nonlinear activation functions, introducing an additional layer already leads to substantially more intricate training dynamics. Consider a two-layer linear network with scalar output,

$$f_{v,W}(x) := v^\top W x, \tag{1.9}$$

where $W \in \mathbb{R}^{m\times d}$, $v \in \mathbb{R}^m$, and m denotes the width of the hidden layer. The least-squares empirical risk is

$$\mathcal{L}(v, W) := \frac{1}{2}\left\|X^\top W^\top v - y\right\|_2^2. \tag{1.10}$$

Although the resulting function is linear in the input, the loss is no longer convex in the parameters (v, W) due to their multiplicative coupling.

Let

$$r := X^\top W^\top v - y \in \mathbb{R}^n \tag{1.11}$$

denote the residual vector. The gradients of $\mathcal{L}$ are given by

$$\nabla_v \mathcal{L} = WXr, \qquad \nabla_W \mathcal{L} = v(Xr)^\top. \tag{1.12}$$

Under continuous-time gradient flow, the parameters therefore evolve according to

$$\dot{v}_t = -\nabla_v \mathcal{L}(v_t, W_t) = W_t X \big(y - X^\top W_t^\top v_t\big), \qquad \dot{W}_t = -\nabla_W \mathcal{L}(v_t, W_t) = v_t \big(y - X^\top W_t^\top v_t\big)^\top X^\top. \tag{1.13}$$

This coupled system consists of a vector-valued and a matrix-valued ordinary differential equation and is nonlinear in the parameters, despite the underlying predictor being linear in the input. As a result, the training dynamics no longer admit a simple spectral characterization as in the one-layer case.

Nevertheless, recent work has shown that deep linear networks admit a more structured description in suitable asymptotic regimes. In particular, Chizat et al. (2024) study gradient flow for deep linear networks in an infinite-width limit, where the dynamics are described at the level of a measure-valued evolution. While no closed-form solution of the finite-dimensional ODE system is obtained, this framework establishes global well-posedness, convergence to global minimizers, and the emergence of implicit regularization effects induced by depth.

Nonlinear networks. The analysis of training dynamics becomes more involved once nonlinear activation functions are introduced. Consider again a two-layer network of the form

$$f_{v,W}(x) = v^\top \phi(Wx),$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable activation function applied elementwise. For the least-squares objective, the corresponding gradient flow equations involve both the activations $\phi(Wx)$ and their derivatives $\phi'(Wx)$. In particular, the evolution of the parameters mixes standard matrix products with elementwise nonlinear operations. As a consequence, the training dynamics of nonlinear networks do not admit a simple closed-form or spectral description in general. This motivates the study of simplified regimes in which the dynamics become tractable, either by linearizing the network around its initialization or by considering suitable large-width limits. Two prominent examples of such approaches are the neural tangent kernel and mean-field formulations, which provide complementary perspectives on the behavior of wide neural networks during training.

Chapter 2

Related Work

2.1 Neural Tangent Kernel

As discussed in Chapter 1.1, for nonlinear neural networks the gradient flow dynamics are parameter-dependent and do not admit a simple closed-form description. One approach to recovering a tractable analytical framework is to consider regimes in which the network behaves approximately linearly around its random initialization. The *Neural Tangent Kernel* (NTK), introduced by Jacot et al. (2020), formalizes this idea by studying training dynamics through a first-order linearization in parameter space, leading to a kernel-based description of learning in wide neural networks.

2.1.1 Linearization around initialization

Let $f(x;\theta)$ denote a neural network with parameters $\theta \in \mathbb{R}^p$, initialized at θ_0. A first-order Taylor expansion around θ_0 yields

$$f(x;\theta) \;\approx\; f(x;\theta_0) + \nabla_\theta f(x;\theta_0)^\top (\theta - \theta_0), \tag{2.1}$$

where $f(x;\theta_0)$ is the network output at initialization and

$$\phi(x) := \nabla_\theta f(x;\theta_0)$$

defines the tangent feature map. Locally around initialization, the network thus behaves as a linear model in parameter space,

$$f(x;\theta) \approx f(x;\theta_0) + \phi(x)^\top (\theta - \theta_0). \tag{2.2}$$

Definition 1 (Neural Tangent Kernel (Jacot et al., 2020)) *Given an initialization* θ_0*, the neural tangent kernel is defined as*

$$\Theta_0(x,x') := \nabla_\theta f(x;\theta_0)^\top \nabla_\theta f(x';\theta_0).$$

The NTK measures how infinitesimal parameter updates couple the network outputs at different inputs and plays the role of an empirical covariance operator in the tangent feature space.

2.1.2 Training dynamics induced by the NTK

We consider training with the squared loss

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{n}\big(f(x_i;\theta) - y_i\big)^2$$

using gradient descent with step size $\eta > 0$,

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta L(\theta_t), \qquad t = 0,1,2,\dots$$

Let $f_t(x_i) := f(x_i;\theta_t)$ the gradient of the loss is

$$\nabla_\theta L(\theta_t) = \sum_{j=1}^{n}\big(f_t(x_j) - y_j\big)\,\nabla_\theta f(x_j;\theta_t).$$

Applying a first-order Taylor expansion of $f(\cdot;\theta)$ around θ_t gives

$$f_{t+1}(x_i) \approx f_t(x_i) + \nabla_\theta f(x_i;\theta_t)^\top(\theta_{t+1} - \theta_t).$$

Substituting the gradient descent update $\theta_{t+1} - \theta_t = -\eta\nabla_\theta L(\theta_t)$ yields

$$f_{t+1}(x_i) = f_t(x_i) - \eta\sum_{j=1}^{n}\nabla_\theta f(x_i;\theta_t)^\top\nabla_\theta f(x_j;\theta_t)\big(f_t(x_j) - y_j\big),$$

which can be written compactly as

$$f_{t+1}(x_i) = f_t(x_i) - \eta\sum_{j=1}^{n}\Theta_t(x_i,x_j)\big(f_t(x_j) - y_j\big), \tag{2.3}$$

where the iteration-dependent NTK is

$$\Theta_t(x_i,x_j) := \nabla_\theta f(x_i;\theta_t)^\top\nabla_\theta f(x_j;\theta_t).$$

If we stack predictions into $f_t := (f_t(x_1),\dots,f_t(x_n))^T \in \mathbb{R}^n$, in vector form, the dynamics read

$$f_{t+1} = f_t - \eta\,\Theta_t(f_t - y). \tag{2.4}$$

where $\bar{\Theta} = \big(\bar{\Theta}(x_i,x_j)\big)_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ is the kernel gram matrix. In general, Θ_t evolves during training, reflecting changes in the network's tangent features.

2.1.3 One-hidden-layer NTK and infinite-width limit

To make the NTK explicit, consider a two-layer network of width m,

$$f(x) = \frac{1}{\sqrt{m}}\sum_{\alpha=1}^{m} v_\alpha\,\varphi(w_\alpha^\top x), \qquad v_\alpha \in \mathbb{R}, \quad w_\alpha, x \in \mathbb{R}^d. \tag{2.5}$$

Each neuron is parameterized by $\theta_\alpha = (v_\alpha, w_\alpha)$ and initialized as

$$v_\alpha \sim \mathcal{N}(0,\sigma_v^2), \qquad w_\alpha \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{d} I_d\right),$$

independently across $\alpha = 1,\dots,m$. $\varphi : \mathbb{R} \to \mathbb{R}$ is a non-linear activation function.

Finite-width NTK at initialization. By direct computation,

$$\nabla_{v_\alpha} f(x) = \frac{1}{\sqrt{m}} \varphi(w_\alpha^\top x), \tag{2.6}$$

$$\nabla_{w_\alpha} f(x) = \frac{1}{\sqrt{m}} v_\alpha \varphi'(w_\alpha^\top x) x. \tag{2.7}$$

Substituting into the definition of the NTK yields the empirical kernel

$$\Theta_0^{(m)}(x,x') = \frac{1}{m} \sum_{\alpha=1}^{m} \varphi(w_\alpha^\top x)\, \varphi(w_\alpha^\top x') + \frac{1}{m} \sum_{\alpha=1}^{m} v_\alpha^2 \varphi'(w_\alpha^\top x)\, \varphi'(w_\alpha^\top x')\, x^\top x'. \tag{2.8}$$

Infinite-width limit. Each sum in (2.8) is an empirical average of i.i.d. terms. By the strong law of large numbers,

$$\Theta_0^{(m)}(x,x') \xrightarrow{\text{a.s.}} \bar{\Theta}(x,x') \qquad \text{as } m \to \infty,$$

where the limiting NTK is deterministic and given by

$$\bar{\Theta}(x,x') = \mathbb{E}_w\big[\varphi(w^\top x)\, \varphi(w^\top x')\big] + \sigma_v^2 x^\top x' \, \mathbb{E}_w\big[\varphi'(w^\top x)\, \varphi'(w^\top x')\big]. \tag{2.9}$$

This argument extends to deep fully connected networks, where both the forward covariance kernel and the NTK satisfy recursive layerwise equations in the infinite-width limit (Jacot et al., 2020; Lee et al., 2020).

2.1.4 Constant-kernel regime and spectral dynamics

In the infinite-width limit, the NTK remains constant throughout training, $\Theta_t \equiv \bar{\Theta}$. Equation (2.4) then reduces to the linear iteration

$$f_{t+1} = f_t - \eta\, \bar{\Theta}(f_t - y). \tag{2.10}$$

Letting $r_t := f_t - y$, we obtain

$$r_{t+1} = (I - \eta \bar{\Theta})\, r_t.$$

Assuming $\eta < 2/\lambda_{\max}(\bar{\Theta})$, the residual admits the closed form

$$r_t = (I - \eta\bar{\Theta})^t r_0, \qquad f_t = y + (I - \eta\bar{\Theta})^t (f_0 - y). \tag{2.11}$$

As in the linear setting of Chapter 1.1, convergence occurs independently along the eigenvectors of $\bar{\Theta}$, with geometric rates determined by the corresponding eigenvalues.

2.1.5 NTK dynamics, solution structure, and implicit bias

In the constant-kernel (infinite-width) regime, the NTK prediction dynamics reduce to a linear iteration in prediction space. Eq 2.10, admits fixed points $f^\star$ satisfying

$$\bar{\Theta}\,(f^\star - y) = 0. \tag{2.12}$$

Equation (2.12) characterizes the set of global minimizers of the squared loss in the NTK regime. As in linear least squares, the structure of this solution set depends on the rank of the operator $\bar{\Theta}$.

If $\bar{\Theta}$ is strictly positive definite on the training set, the solution is unique and satisfies $f^\star = y$. When $\bar{\Theta}$ is singular, the loss admits infinitely many interpolating solutions in prediction space, differing by elements of $\ker(\bar{\Theta})$. In this case, gradient descent converges to a particular fixed point determined by the initialization.

Tangent features and operator factorization. To formulate the NTK in a way that remains meaningful in the infinite-width limit, it is convenient to view parameter perturbations as elements of a Hilbert space $\mathcal{H}_\theta$ equipped with an inner product $\langle \cdot, \cdot \rangle$. In the finite-dimensional case, $\mathcal{H}_\theta = \mathbb{R}^p$ with the Euclidean inner product; in the infinite-width limit, $\mathcal{H}_\theta$ denotes the corresponding limit space of parameter perturbations, often referred to as the tangent parameter space (Jacot et al., 2020; Lee et al., 2020).

In this setting, the limiting NTK Gram matrix on the training set can be expressed as

$$\bar{\Theta} = J_0 J_0^*, \tag{2.13}$$

where $J_0 : \mathcal{H}_\theta \to \mathbb{R}^n$ denotes the Jacobian of the network outputs with respect to parameters at initialization, viewed as a linear operator,

$$(J_0 h)_i = \langle \nabla_\theta f(x_i; \theta_0), h \rangle,$$

and J_0^* is its adjoint. This representation makes explicit that $\bar{\Theta}$ is symmetric and positive semidefinite, and that it acts as an empirical covariance operator for the tangent features induced by the network at initialization (Jacot et al., 2020).

Implicit bias and representer viewpoint. When $\bar{\Theta}$ is singular, the Moore–Penrose pseudoinverse $\bar{\Theta}^+$ acts as the inverse on $\mathrm{range}(\bar{\Theta})$ and annihilates $\ker(\bar{\Theta})$. Consequently, constant-kernel NTK training eliminates only the residual component lying in $\mathrm{range}(\bar{\Theta})$ while preserving the component in $\ker(\bar{\Theta})$. This behavior reflects an implicit regularization effect analogous to linear least squares and can be interpreted as convergence to a minimum-norm solution in the reproducing kernel space associated with $\bar{\Theta}$, as formalized by representer-theorem results (Bietti and Mairal, 2019; Bartolucci et al., 2021).

Limitations of stable kernels and feature learning. Although the NTK framework provides a clean and tractable description of training dynamics, the stability of the kernel also introduces inherent limitations. When the kernel does not change during training, learning is effectively confined to a fixed feature space. Indeed, any positive semidefinite kernel admits a representation $K(x,x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$, so that optimization in the NTK regime corresponds to fitting a linear model in the associated reproducing kernel Hilbert space $\mathcal{H}$.

In contrast, for finite-width networks the NTK typically evolves during training, implying that the tangent features $\Phi_t(x) = \nabla_\theta f(x; \theta_t)$ also change over time. This evolution allows the network to adapt its representation to the data, a phenomenon commonly referred

to as *feature learning*. Such effects are absent in a strictly stable-kernel regime, where only the coefficients of fixed features are adjusted.

The importance of feature learning is already visible in simplified settings. Even purely linear networks exhibit nontrivial training dynamics due to the coupling of parameters across layers, leading to implicit biases that cannot be captured by a fixed kernel model (Saxe et al., 2014; Tu et al., 2024). From this viewpoint, the NTK regime can be seen as a useful but restrictive approximation, which motivates studying regimes where kernel evolution and feature learning play an explicit role.

2.2 Mean-field representation of a two-layer network

While the neural tangent kernel provides a linearized description of training dynamics around initialization, it does not capture regimes in which features evolve significantly during optimization. An alternative and complementary approach is provided by the *mean-field* perspective, which studies wide neural networks by viewing their parameters as interacting particles and describing training as the evolution of a probability distribution over parameter space.

As the network width tends to infinity under appropriate scalings, the discrete training dynamics induced by stochastic gradient descent converge to a deterministic evolution equation for the parameter distribution. This viewpoint was first formalized for two-layer networks by Mei et al. (2018), who derived a nonlinear partial differential equation governing the evolution of the parameter measure. Related interacting particle system formulations and convergence results were developed by Rotskoff and Vanden-Eijnden (2022), while extensions to multilayer networks were proposed by Nguyen (2019). From an optimization perspective, Chizat and Bach (2018) showed that, under suitable initialization and in the many-particle limit, training dynamics can be interpreted as a Wasserstein gradient flow on the space of probability measures, and that this flow converges to global minimizers despite the non-convexity of the finite-dimensional parameterization.

In the following, we adopt this mean-field viewpoint and reformulate a two-layer network as a linear functional of an empirical measure on parameter space, which serves as the starting point for distributional descriptions of training dynamics.

Consider a two-layer neural network of the form

$$f_\theta(x) = \frac{1}{m}\sum_{\alpha=1}^{m} v_\alpha\, \varphi(w_\alpha^\top x), \qquad v_\alpha \in \mathbb{R},\ w_\alpha, x \in \mathbb{R}^d, \tag{2.14}$$

where each neuron is parameterized by $\theta_\alpha = (v_\alpha, w_\alpha)$, $\alpha \in \{1,\dots,m\}$

2.2.1 Parameter space and empirical measure.

Let $\Omega := \mathbb{R}\times\mathbb{R}^d$ denote the parameter space, and let $\mathcal{F}$ be its Borel σ-algebra. For each neuron α, define the Dirac measure δ_{θ_α} on $(\Omega,\mathcal{F})$, i.e. if $A \in \mathcal{F}$ then,

$$\delta_{\theta_\alpha}(A) = \begin{cases} 1, & \theta_\alpha \in A, \\ 0, & \text{otherwise.} \end{cases}$$

Define the empirical neuronal measure

$$\mu_m := \frac{1}{m} \sum_{\alpha=1}^{m} \delta_{\theta_\alpha}. \tag{2.15}$$

By construction, $\mu_m \geq 0$ and $\mu_m(\Omega) = 1$, hence $\mu_m \in \mathcal{P}(\Omega)$ where $\mathcal{P}(\Omega)$ is the space of Borel probability measures on Ω.

Integration against Dirac measures. Let χ_A denote the indicator function of a measurable set $A \subset \Omega$. Then, for any $\theta_\alpha \in \Omega$,

$$\int_\Omega \chi_A(\theta)\, d\delta_{\theta_\alpha}(\theta) = \delta_{\theta_\alpha}(A) = \chi_A(\theta_\alpha).$$

More generally, if $s(\theta) = \sum_{k=1}^{r} c_k \chi_{A_k}(\theta)$ is a simple function, then

$$\int_\Omega s(\theta)\, d\delta_{\theta_\alpha}(\theta) = \sum_{k=1}^{r} c_k \chi_{A_k}(\theta_\alpha) = s(\theta_\alpha).$$

Any nonnegative measurable function $g : \Omega \to \mathbb{R}$ can be approximated by a sequence of simple functions,$\{s_n\}_{n\geq 0}$. If $s_n \uparrow g$ and each $s_n \geq 0$ then by Monotone Convergence Theorem,

$$\int_\Omega g\, d\delta_{\theta_\alpha} = \int_\Omega \lim_{n\to\infty} s_n\, d\delta_{\theta_\alpha} = \lim_{n\to\infty} \int_\Omega s_n\, d\delta_{\theta_\alpha} = \lim_{n\to\infty} s_n(\theta_\alpha) = g(\theta_\alpha). \tag{2.16}$$

Integral representation of the network. Consider now the integral of g with respect to the empirical measure μ_m:

$$\int_\Omega g\, d\mu_m = \int_\Omega g\, d\left(\frac{1}{m} \sum_{\alpha=1}^{m} \delta_{\theta_\alpha}\right) = \frac{1}{m} \sum_{\alpha=1}^{m} g(\theta_\alpha).$$

Define

$$g(\theta_\alpha) = g(v_\alpha, w_\alpha) := v_\alpha\, \varphi(w_\alpha^\top x).$$

Then

$$\int_\Omega g\, d\mu_m = \frac{1}{m} \sum_{\alpha=1}^{m} v_\alpha\, \varphi(w_\alpha^\top x) = f_\theta(x), \tag{2.17}$$

recovering the finite-width network (2.14).

Mean-field viewpoint. Equation (2.17) shows that the network output can be written as a linear functional of the empirical measure μ_m. This representation is only possible because we scale the output by $1/m$ and because the hidden neurons are permutation invariant. This implies that the model depends on the parameters $\{\theta_\alpha\}_{\alpha=1}^m$ only through their empirical distribution. Such observations motivate the mean-field viewpoint, in which the empirical measure μ_m is regarded as the fundamental state variable describing the network (Mei et al., 2018; Chizat and Bach, 2018).

Initialization and weak convergence. Assume that the parameters $\{\theta_\alpha(0)\}_{\alpha=1}^m$ are initialized i.i.d. according to a probability measure μ_0 on Ω. Define the empirical measure at initialization

$$\mu_{m,0} := \frac{1}{m}\sum_{\alpha=1}^{m} \delta_{\theta_\alpha(0)}.$$

Then $\mu_{m,0}$ converges weakly to μ_0 almost surely as $m \to \infty$. Indeed, for any bounded continuous test function $\psi : \Omega \to \mathbb{R}$,

$$\int_\Omega \psi \, d\mu_{m,0} = \frac{1}{m}\sum_{\alpha=1}^{m} \psi(\theta_\alpha(0)) \xrightarrow[m\to\infty]{\text{a.s.}} \mathbb{E}_{\theta\sim\mu_0}[\psi(\theta)] = \int_\Omega \psi \, d\mu_0,$$

where the convergence follows from the strong law of large numbers applied to the i.i.d. random variables $\psi(\theta_\alpha(0))$. Since this holds for all bounded continuous ψ, we conclude that $\mu_{m,0} \to \mu_0$ (Varadarajan, 1958). As a consequence, the network output at initialization converges pointwise to the deterministic limit

$$f_{\mu_0}(x) := \int_\Omega v\,\varphi(w^\top x)\, d\mu_0(v,w).$$

For instance, under i.i.d. Gaussian initialization of the parameters, i.e. $v_0 \sim \mathcal{N}(0, I_m)$ $W_0 \sim \mathcal{N}(0, I_m \otimes I_d)$, the limiting initial measure $\mu_0 = \mathcal{N}(0, I_{d+1})$ is a Gaussian measure on a $d+1$-dimensional space.

2.2.2 Training dynamics and evolution of the empirical measure

Let $X = (x_1, \ldots, x_n) \in \mathbb{R}^{d\times n}$ and $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ denote the training set, and consider the squared loss

$$L(v, W) = \frac{1}{2}\sum_{i=1}^{n} \big(f_{v,W}(x_i) - y_i\big)^2 = \frac{1}{2}\|f_{v,W}(X) - y\|_2^2,$$

where $f_{v,W}(X) := (f_{v,W}(x_i))_{i=1}^n$.

We study continuous-time gradient flow for a finite-width network with m neurons and learning rate $\eta > 0$,

$$\dot{v}_{t,\alpha} = -\eta\, \nabla_{v_\alpha} L(v_t, W_t), \qquad \dot{w}_{t,\alpha} = -\eta\, \nabla_{w_\alpha} L(v_t, W_t), \qquad \alpha \in \{1, \ldots, m\}.$$

A direct computation yields, for each α,

$$\dot{v}_{t,\alpha} = \frac{\eta}{m}\, \varphi(X^\top w_{t,\alpha})^\top \big(y - f_{v_t,W_t}(X)\big), \tag{2.18}$$

$$\dot{w}_{t,\alpha} = \frac{\eta}{m}\, X\Big(\varphi'(X^\top w_{t,\alpha}) \odot \big(y - f_{v_t,W_t}(X)\big)\Big) v_{t,\alpha}, \tag{2.19}$$

where $\odot$ denotes elementwise (Hadamard) multiplication.

As seen in (2.18)–(2.19), evolution of each neuron depends on the parameters only through the current empirical measure $\mu_{m,t}$ via $f_{v_t,W_t}(X)$. Thus, the particle system (2.18)–(2.19) induces an evolution of the empirical measure in parameter space (Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2022).

Hence, the particle system (2.18)–(2.19) can be viewed as an interacting particle system driven by a measure-dependent velocity field. For any $\theta = (v,w) \in \Omega$ and any $\mu \in \mathcal{P}(\Omega)$, define

$$b((v,w);\mu) := \begin{pmatrix} \varphi(X^\top w)^\top \big(y - f_\mu(X)\big) \\ X\Big(\varphi'(X^\top w) \odot \big(y - f_\mu(X)\big)\Big) v \end{pmatrix}$$

Then the gradient-flow dynamics can be written compactly as

$$\dot{\theta}_{t,\alpha} = \frac{\eta}{m}\, b(\theta_{t,\alpha}; \mu_{m,t}), \qquad \alpha = 1, \dots, m.$$

In particular, each particle interacts with the rest of the system only through the current empirical measure $\mu_{m,t}$ (via $f_{\mu_{m,t}}(X)$).

Equivalently, $\mu_{m,t}$ solves the continuity equation (Rotskoff and Vanden-Eijnden, 2022; Chizat and Bach, 2018).

$$\partial_t \mu_{m,t} + \nabla_\theta \cdot \Big(\mu_{m,t}\, \frac{\eta}{m}\, b(\cdot; \mu_{m,t})\Big) = 0, \tag{2.20}$$

in the sense of distributions, i.e. for every test function $\psi \in C_c^\infty(\Omega)$,

$$\frac{d}{dt} \int_\Omega \psi(\theta)\, d\mu_{m,t}(\theta) = \frac{\eta}{m} \int_\Omega \nabla_\theta \psi(\theta) \cdot b(\theta; \mu_{m,t})\, d\mu_{m,t}(\theta).$$

Choosing the mean-field scaling $\eta = m$ yields a nontrivial $O(1)$ evolution in time and, as $m \to \infty$, one expects $\mu_{m,t} \to \mu_t$, where the limit μ_t satisfies (Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2022; Golikov, 2025; Chizat and Bach, 2018).

$$\partial_t \mu_t + \nabla_\theta \cdot \big(\mu_t\, b(\cdot; \mu_t)\big) = 0, \qquad \mu_0 = \mathcal{N}(0, I_{d+1}), \tag{2.21}$$

Push-forward formulation. Recall that if $g : \mathcal{X} \to \mathcal{Z}$ is measurable and $\mu \in \mathcal{P}(\mathcal{X})$, then the *push-forward* of μ by g is the measure $g_*\mu \in \mathcal{P}(\mathcal{Z})$ defined by

$$(g_*\mu)(A) := \mu\big(g^{-1}(A)\big), \qquad A \subset \mathcal{Z} \text{ measurable.}$$

Let $\Phi_t : \Omega \to \Omega$ denote the flow map associated with the velocity field $\frac{\eta}{m}\, b(\cdot; \mu_{m,t})$. Then the empirical measure is transported by this flow:

$$\mu_{m,t} = (\Phi_t)_* \mu_{m,0}.$$

In particular, for any measurable observable $h : \Omega \to \mathbb{R}$,

$$\int_\Omega h(\theta)\, d\mu_{m,t}(\theta) = \int_\Omega h(\Phi_t(\theta))\, d\mu_{m,0}(\theta),$$

so evaluating the network at time t is obtained by taking $h_x(\theta) := v\,\varphi(w^\top x)$ and writing

$$f_{\mu_{m,t}}(x) = \int_\Omega v\,\varphi(w^\top x)\, d\mu_{m,t}(v,w) = \int_\Omega v\,\varphi(w^\top x)\, d\big((\Phi_t)_*\mu_{m,0}\big)(v,w).$$

Advantages over the NTK limit. Recall the finite-width NTK expression (2.8). In the mean-field parameterization $f(x) = \frac{1}{m}\sum_{\alpha=1}^{m} v_\alpha \varphi(w_\alpha^\top x)$, the empirical tangent kernel is

$$\Theta_t^{(m)}(x,x') = \frac{1}{m^2}\sum_{\alpha=1}^{m}\Big(\varphi(w_{t,\alpha}^\top x)\,\varphi(w_{t,\alpha}^\top x') + v_{t,\alpha}^2\,\varphi'(w_{t,\alpha}^\top x)\,\varphi'(w_{t,\alpha}^\top x')\,x^\top x'\Big). \tag{2.22}$$

Assuming the weights remain $O(1)$ as $m\to\infty$, each coordinate of $\nabla_\theta f_t(x)$ is $O(1/m)$, so $\Theta_t^{(m)}(x,x') = \langle \nabla_\theta f_t(x), \nabla_\theta f_t(x')\rangle$ is a sum of m terms of size $O(m^{-2})$, hence $\Theta_t^{(m)}(x,x') = O(m^{-1})$. Moreover, for the squared loss, differentiating the predictions along gradient flow yields

$$\partial_t f_t(X) = -\eta\,\Theta_t^{(m)}(X,X)\big(f_t(X)-y\big),$$

so the effective operator driving learning is the *scaled* kernel $\eta\,\Theta_t^{(m)}$. The mean-field time scaling $\eta = m$ therefore produces an $O(1)$ evolution and yields a finite kernel limit:

$$\lim_{m\to\infty}\big(\eta\,\Theta_t^{(m)}(x,x')\big) = \int_\Omega\Big(\varphi(w^\top x)\,\varphi(w^\top x') + v^2\,\varphi'(w^\top x)\,\varphi'(w^\top x')\,x^\top x'\Big)\,d\mu_t(v,w).$$

As μ_t evolves according to the mean-field transport equation, this limiting kernel also evolves in time, capturing feature learning beyond the frozen-kernel NTK regime (Golikov, 2025).

Disadvantages. The mean-field scaling $\eta = m$ yields a deterministic measure-valued evolution $(\mu_t)_{t\ge 0}$ as $m\to\infty$, governed by the nonlinear continuity equation (2.21). However, its long-time behavior is generally difficult to characterize: it is not clear in general whether μ_t converges as $t\to\infty$, nor which minimizer (if any) is selected at the level of measures. Nevertheless, when the dynamics is interpreted as a Wasserstein gradient flow, there exist results proving global convergence of the loss L under suitable assumptions (Mei et al., 2018; Chizat and Bach, 2018). Moreover, unlike the NTK limit, extending the above measure-evolution construction beyond two-layer networks is not straightforward. The main difficulty is that for deep networks the hidden units are no longer permutation-invariant in a way that yields a simple empirical measure on a fixed parameter space. Several multilayer mean-field-type constructions have been proposed to address this, but they are typically more involved than the two-layer case (Nguyen, 2019; Araújo et al., 2019; Sirignano and Spiliopoulos, 2019; Nguyen and Pham, 2023).

2.3 Spectral bias

A striking and widely reported phenomenon in neural network training is that gradient-based optimization does not fit all components of a target function at the same rate. Instead, networks tend to learn "simple" structure first and refine fine-scale or highly oscillatory structure later. This preference is commonly referred to as *spectral bias* (or the *frequency principle*). In its classical formulation, spectral bias asserts that, when a target function is decomposed into Fourier modes, lower-frequency components are learned earlier and at a faster rate than higher-frequency components (Rahaman et al., 2019).

In this section we briefly review empirical evidence for spectral bias and summarize theoretical viewpoints that relate it to the spectrum of the operator governing training dynamics in the kernel (NTK) regime. We also highlight extensions that study how the phenomenon depends on the input distribution and how it manifests outside the training set.

2.3.1 Empirical evidence and experimental protocols

A standard way to empirically probe spectral bias is to choose a target with a controlled frequency decomposition and to track the frequency content of the network's prediction f_t during training.

Synthetic Fourier regression. A canonical experiment is 1D regression on a target constructed as a sum of sinusoids,

$$y(z) = \sum_{i=1}^{r} A_i \sin(2\pi k_i z + \phi_i), \qquad z \in [0,1],$$

and monitoring the discrete Fourier spectrum of the learned predictor as a function of training time. Rahaman et al. (2019) report that, for deep ReLU networks trained by (full-batch) gradient descent, Fourier coefficients at smaller frequencies k_i grow substantially earlier than those at larger k_i, even when the amplitudes are matched or when high-frequency components have larger amplitude (Rahaman et al., 2019). This "frequency-dependent learning speed" is one of the most direct empirical signatures of spectral bias.

Robustness and perturbation tests. A complementary protocol is to train to near-zero training error and then apply random perturbations in parameter space, comparing how the Fourier spectrum of the realized function changes. Rahaman et al. (2019) observe that lower-frequency components of the learned function are substantially more robust to such perturbations than higher-frequency components, suggesting that the network parameterization itself represents low frequencies in a more stable manner (Rahaman et al., 2019).

Distributional effects and "local" frequency learning. Spectral bias is often presented under uniformly sampled inputs, but empirical studies also examine how it interacts with the input distribution. For example, Basri et al. (2020) compare learning under uniform and non-uniform sampling densities and show that the ordering "low frequencies before high frequencies" can be modulated by where samples concentrate: dense regions of the input space may exhibit faster learning of higher-frequency structure than sparse regions (Basri et al., 2020). This line of experimentation motivates viewing spectral bias as a property of an operator defined jointly by the model and the data distribution, rather than as a purely architectural effect.

2.3.2 Operator spectrum viewpoint and the NTK regime

A common theoretical explanation of spectral bias proceeds by identifying an operator whose eigendecomposition controls learning rates. In the kernel/NTK regime, training dynamics are approximately linear in function space and decompose along eigendirections

of the corresponding kernel operator. In this setting, spectral bias can be interpreted more generally as a bias toward learning the leading eigenfunctions of the kernel (rather than specifically Fourier modes) (Bowman and Montufar, 2022).

Decomposition along NTK eigenfunctions. Cao et al. (2020) give a rigorous account of this mechanism in the NTK regime: the training process can be decomposed along eigenfunctions of the NTK operator, with each component converging at a rate determined by the associated eigenvalue. As a consequence, components aligned with larger eigenvalues are learned faster, yielding a principled notion of "spectral preference" (Cao et al., 2020).

In general, the eigenfunctions of the NTK operator depend jointly on the input distribution and the kernel, so they need not coincide with the standard Fourier basis. A simplification occurs in the *infinite-width* (kernel) limit, where the empirical NTK concentrates around a deterministic limit $\bar{\Theta}$ given by an expectation over the random initialization.

To illustrate how Fourier modes arise, consider inputs on the circle S^1, parameterized by $\theta \in [0, 2\pi)$ and embedded as $x(\theta) = (\cos\theta, \sin\theta) \in \mathbb{R}^2$. Assume isotropic initialization (e.g. Gaussian), so that $R^\top w$ has the same distribution as w for every orthogonal matrix R. Writing $\bar{\Theta}$ as an expectation (cf. Eq. (2.9)) and using this rotational invariance in distribution implies

$$\bar{\Theta}(Rx, Rx') = \bar{\Theta}(x, x') \qquad \text{for all orthogonal } R.$$

Since $x(\theta+\alpha) = R_\alpha x(\theta)$ and $x(\theta)^\top x(\theta') = \cos(\theta-\theta')$, it follows that $\bar{\Theta}(\theta, \theta')$ depends only on $\theta - \theta'$; equivalently, there exists κ such that

$$\bar{\Theta}(\theta, \theta') = \kappa(\theta - \theta').$$

Convolution form on S^1. Let ρ be the uniform probability measure on S^1, $d\rho(\theta') = d\theta'/(2\pi)$, and define the integral operator

$$(T_{\bar{\Theta}} g)(\theta) := \int_0^{2\pi} \bar{\Theta}(\theta, \theta')\, g(\theta')\, \frac{d\theta'}{2\pi}.$$

Using $\bar{\Theta}(\theta, \theta') = \kappa(\theta - \theta')$ and the change of variables $u = \theta - \theta'$, we obtain

$$(T_{\bar{\Theta}} g)(\theta) = \int_0^{2\pi} \kappa(u)\, g(\theta - u)\, \frac{du}{2\pi} =: (\kappa * g)(\theta),$$

which is the (circular) convolution operator on S^1.

Fourier eigenfunctions and frequency ordering. Because $T_{\bar{\Theta}}$ is a convolution, the Fourier modes $\phi_q(\theta) = e^{iq\theta}$, $q \in \mathbb{Z}$, are the eigenfunctions:

$$(T_{\bar{\Theta}} \phi_q)(\theta) = e^{iq\theta} \int_0^{2\pi} \kappa(u) e^{-iqu}\, \frac{du}{2\pi} = \lambda_q\, \phi_q(\theta), \qquad \lambda_q = \int_0^{2\pi} \kappa(u) e^{-iqu}\, \frac{du}{2\pi}.$$

If κ is real and even (as in $\kappa(u) = \psi(\cos u)$), then $\lambda_q \in \mathbb{R}$ and $\lambda_q = \lambda_{-q}$, and one may equivalently use the real basis $\{\cos(q\theta), \sin(q\theta)\}$. In kernel gradient dynamics, each mode

contracts independently at a rate set by λ_q (e.g. $(1-\eta\lambda_q)^t$ in discrete time), so modes with larger eigenvalues are learned faster. In symmetric NTK settings on the circle/sphere, the eigenvalues decrease with frequency, leading to the characteristic "low frequencies first" behavior (Basri et al., 2020; Rahaman et al., 2019).

2.3.3 Uniform vs. non-uniform sampling

The Fourier-mode picture above relies not only on a shift-invariant kernel but also on the *uniform* input distribution on S^1. Under uniform sampling, $d\rho(\theta') = d\theta'/(2\pi)$, shift invariance $\bar{\Theta}(\theta,\theta') = \kappa(\theta-\theta')$ implies that the integral operator is a circular convolution and therefore diagonalizes in the Fourier basis. In this setting, spectral bias can be stated directly in terms of frequency: learning rates are controlled by the eigenvalues λ_q associated with Fourier modes.

Under *non-uniform* sampling, this simplification breaks. If inputs are distributed according to a density $p(\theta)$ on S^1, then the relevant population operator becomes

$$(T_p g)(\theta) := \int_0^{2\pi} \kappa(\theta-\theta')\, g(\theta')\, p(\theta')\, \frac{d\theta'}{2\pi}.$$

Even when κ depends only on $\theta-\theta'$, the extra factor $p(\theta')$ destroys shift invariance, so T_p is no longer a pure convolution operator and its eigenfunctions need not be the global Fourier modes. As a result, a Fourier harmonic target can project onto multiple eigendirections of T_p, and "frequency" becomes distribution-dependent: the modes learned early are those aligned with the *top eigenfunctions of the density-weighted operator* rather than the lowest Fourier frequencies.

This dependence on p is analyzed explicitly by Basri et al. (2020), who study the kernel regime on the circle/sphere and show that non-uniform densities can produce eigenfunctions with localized oscillatory structure and learning behavior that varies across the input space, with denser regions exhibiting effectively faster learning of finer-scale structure (Basri et al., 2020).

2.3.4 Spectral bias beyond the training set

Most early demonstrations of spectral bias are reported on the training set via Fourier spectra or projections onto empirical kernel eigenvectors. A natural question is whether a comparable phenomenon holds in function space more globally. Bowman and Montufar (2022) address this in the kernel regime by proving bounds that compare finite-width training trajectories to idealized kernel dynamics and conclude that networks inherit the bias of the NTK integral operator over the input space, not merely at the training samples. In particular, they argue that eigenfunctions of the NTK integral operator are learned at rates corresponding to their eigenvalues, providing a mechanism for spectral bias that persists outside the training set (Bowman and Montufar, 2022).

2.3.5 Discussion and connection to feature learning

The operator-spectrum viewpoint provides a clean account of spectral bias in kernelized regimes, where learning rates are dictated by the spectrum of a fixed (or nearly fixed) operator. Outside the NTK regime, however, features can evolve substantially during optimization, and the relevant operator (e.g. the tangent kernel) may drift. From this perspective, spectral bias can be studied both as a *property of the initial linearized dynamics* and as a phenomenon that may interact with feature learning through time-dependent spectral structure. This interaction motivates empirical analyses that track mode-wise error decay alongside kernel evolution and provides a natural link between kernel-based and mean-field viewpoints.

Chapter 3

Preliminary Experiments

Purpose. This chapter summarizes exploratory experiments conducted during the early phase of this thesis. They are *not* the final experimental results, but they served two roles: (i) to sanity-check the NTK baseline (infinite-width predictor and kernel-profile behavior), and (ii) to identify diagnostics and regimes where finite-width effects (kernel drift, mode-wise decay, and spectral bias) become visible and measurable. Unless stated otherwise, models are fully-connected ReLU MLPs trained with full-batch gradient descent over multiple random seeds.

3.1 Experimental setting and diagnostics

Probe manifold and Fourier regression targets. Several experiments use the unit circle S^1 parameterized by $\gamma \in [0, 2\pi)$ and embedded as $x(\gamma) = (\cos\gamma, \sin\gamma)$. This choice is convenient because (i) it yields an interpretable Fourier basis for the target and residual, and (ii) in symmetric kernel regimes the relevant integral operator diagonalizes in Fourier modes (Section 2.3).

Function-space comparison to the NTK predictor. When an analytic (infinite-width) NTK predictor is available for the same architecture/initialization, we compare the finite-width network prediction $f_{\text{net}}(\gamma, t)$ to the NTK prediction $f_{\text{NTK}}(\gamma, t)$ on a dense evaluation grid. We report overlays and a relative error metric (as implemented in the experimental codebase).

Empirical NTK snapshots, drift, and spectra. Let X_{train} denote the training inputs and define the empirical NTK Gram matrix on the training set

$$K(t) \;:=\; \Theta_t(X_{\text{train}}, X_{\text{train}}) \in \mathbb{R}^{M \times M}.$$

We track the normalized kernel drift

$$\Delta_K(t) \;=\; \frac{\|K(t) - K(0)\|_F}{\|K(0)\|_F},$$

and eigenvalue trajectories of $K(t)$, focusing on the leading eigenvalues which govern the fastest directions in kernelized dynamics (Section 2.1).

Kernel-regression baseline at a frozen kernel. Given a snapshot time $t^\star$, we form the (ridge) kernel regression solution with kernel $K_{\text{train,train}}(t^\star)$ and cross-kernel $K_{\text{eval,train}}(t^\star)$:

$$\alpha(t^\star) = \left(K_{\text{train,train}}(t^\star) + \lambda I\right)^{-1} y_{\text{train}}, \qquad f_{\text{KR}}(\cdot; t^\star) = K_{\text{eval,train}}(t^\star)\,\alpha(t^\star),$$

where $\lambda \geq 0$ is a numerical stabilizer (set small in practice).

Mode-wise residual projections. To probe spectral bias and its interaction with kernel evolution, we study residual projections onto (i) the top eigenvectors of $K(t)$, and (ii) the known Fourier components of the target when the target is a Fourier mixture on S^1.

3.2 EXP001: Finite-width networks and convergence to the NTK predictor

This block of experiments tests whether increasing width drives finite-width MLP predictions toward the infinite-width NTK predictor, and isolates when discrepancies are due to (a) representational limitations at small width versus (b) slow convergence of small-eigenvalue modes (spectral bias / finite effective time).

3.2.1 Kernel-profile reproduction on a probe manifold

We first reproduce a standard qualitative diagnostic from the NTK literature: the kernel profile $\gamma \mapsto \Theta_{\theta_t}(x_0, x(\gamma))$ on the unit circle, with anchor $x_0 = (1,0)$ and probe points $x(\gamma) = (\cos\gamma, \sin\gamma)$. Training itself is performed on Gaussian inputs with target $f^\star(x) = x_1 x_2$; the circle is used only for probing. We use a ReLU MLP with depth $L = 4$ in the NTK parameterization, widths $m \in \{100, 500, 2000\}$, trained for 200 steps with $\eta = 1.0$ across 10 seeds.

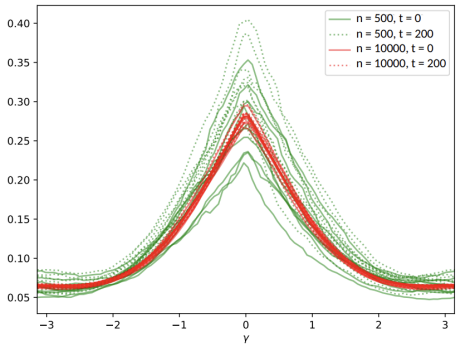
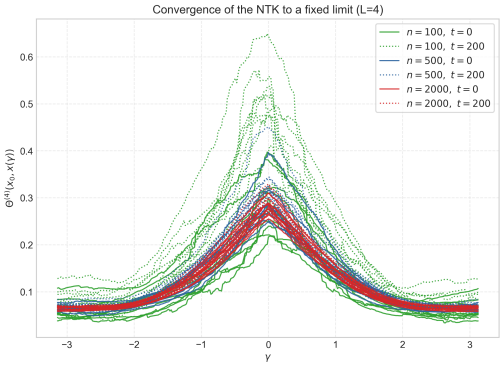
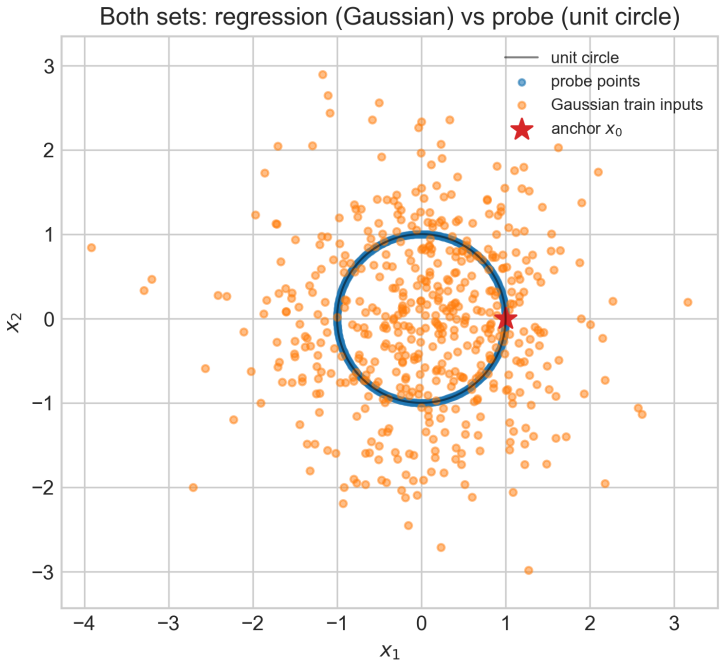


Figure 1: Convergence of the NTK to a fixed limit for two widths n and two times t.

(a) Reference profile (NTK paper).



(b) Empirical profile ($t = 0$ solid, $t = 200$ dotted).



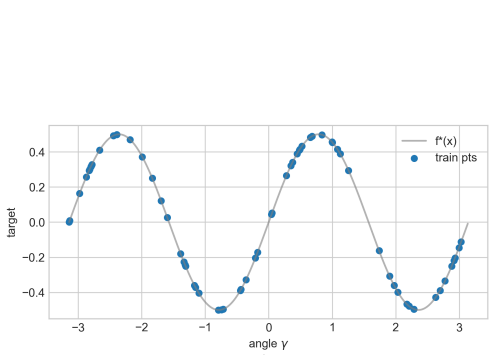
(c) Gaussian training inputs vs circle probe manifold.

Figure 3.1: Kernel-profile diagnostic used in EXP001.

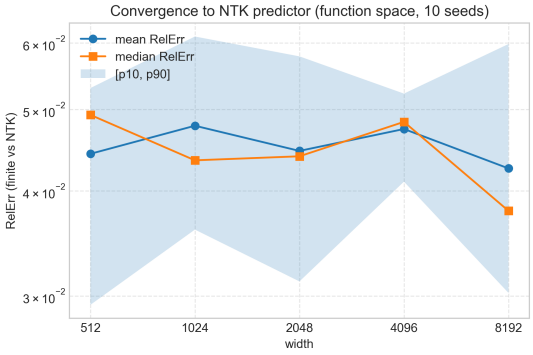
Observation. The empirical kernel profile concentrates as width increases, with markedly reduced variance across seeds. The peak near $\gamma = 0$ reflects self-similarity ($x_0^\top x(\gamma) \approx 1$). This reproduces the expected "concentration toward a deterministic NTK" picture at initialization and suggests that, in this setting, kernel evolution during short training is small at large width.

3.2.2 Function-space convergence on S^1 for a simple target

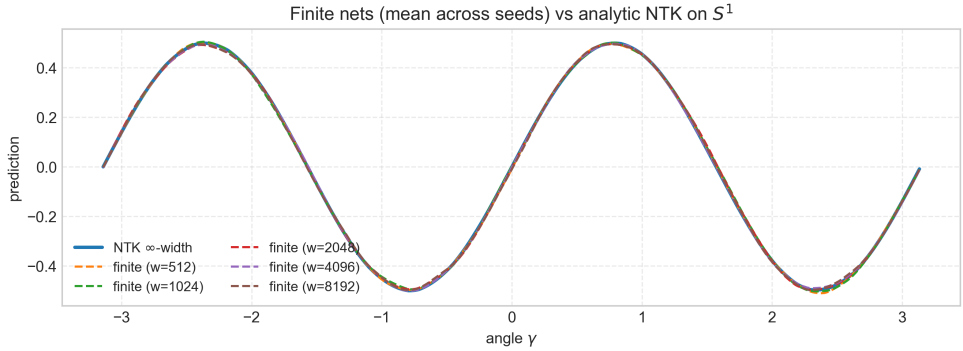
We next test function-space convergence directly on the circle with the low-frequency target $f^\star(x) = x_1 x_2 = \frac{1}{2}\sin(2\gamma)$. We use depth $L = 1$ and sweep widths $m \in \{64, 128, 256, 512, 1024, 2048, 4096, 8192\}$, training for 30,000 steps with $\eta = 10^{-2}$. We compare to an analytic NTK predictor built using the same activation and initialization.



(a) Target: $\frac{1}{2}\sin(2\gamma)$.



(b) RelErr (finite vs NTK).



(c) Prediction overlays (finite vs analytic NTK).

Figure 3.2: Function-space convergence on S^1 for a simple low-frequency target (EXP001).

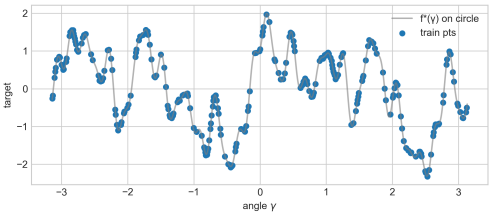
Result. Finite-width networks match the analytic NTK predictor almost exactly on this task; the relative error is flat across widths. This is consistent with the target aligning strongly with a single low-frequency eigendirection, which is learned rapidly under kernel dynamics.

3.2.3 Fourier-mixture target: apparent low-frequency trapping

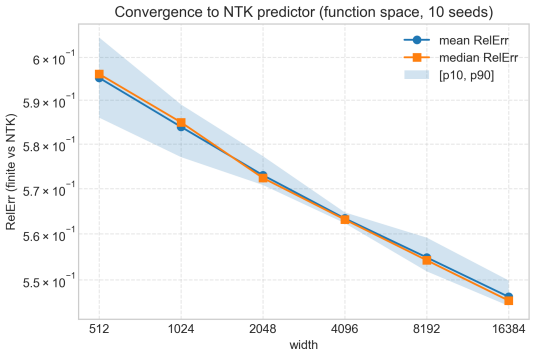
We then move to a Fourier mixture target

$$y(\gamma) = \sum_{k \in \{2,4,7,11,16,23,32\}} a_k \sin(k\gamma + \phi_k),$$

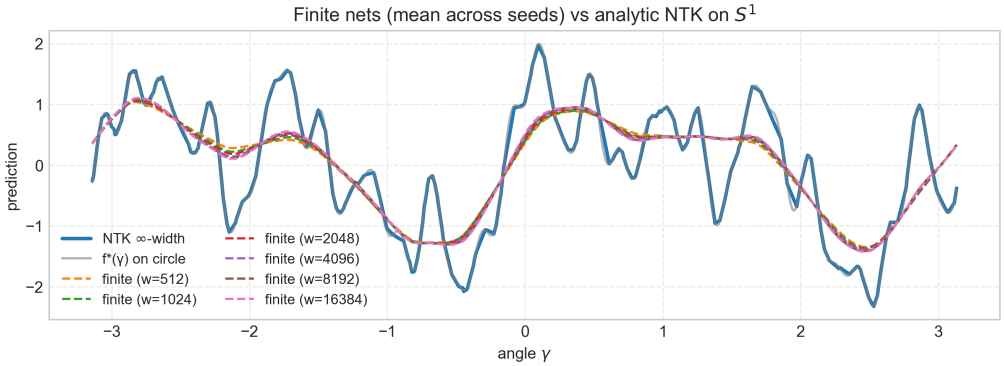
trained for 30,000 steps with $\eta = 0.01$.



(a) Fourier-mixture target.



(b) RelErr vs width.



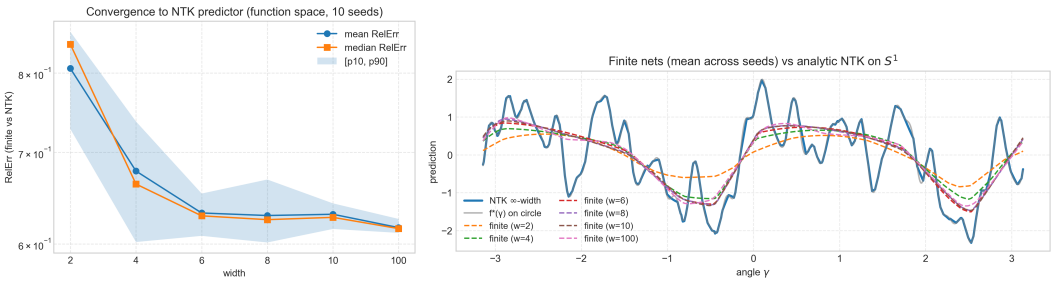
(c) Prediction overlays (finite vs analytic NTK).

Figure 3.3: Fourier-mixture task highlights slow convergence of high-frequency structure under a fixed horizon (EXP001).

Observation. Across the moderate-to-large widths shown, the learned functions appear qualitatively similar and dominated by low harmonics. Relative error decreases only slightly with width under this fixed training horizon, suggesting that width alone is not the main bottleneck at these scales.

3.2.4 Very small widths reveal progressive convergence

To make convergence differences visible, we sweep down to extremely small widths. At widths as small as $m = 2$, representing seven harmonics is difficult, and function-space differences become clear.



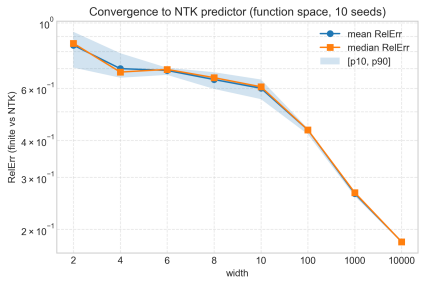
(a) RelErr at very small widths. (b) Function-space progression.

Figure 3.4: Very small widths reveal visible progression toward the NTK predictor (EXP001).

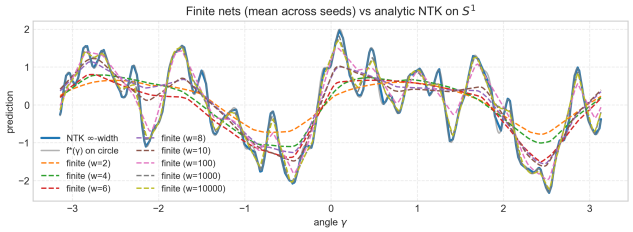
Observation. Narrow networks capture only coarse structure (low-order harmonics). Increasing width improves representation and moves predictions closer to the NTK predictor, consistent with the expectation that finite-width predictors approach the NTK limit.

3.2.5 Longer effective training reveals slow high-frequency convergence

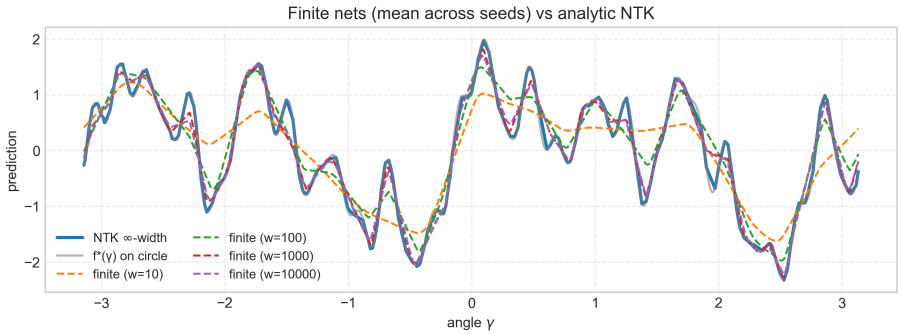
Motivated by spectral-bias considerations (Section 2.3), we test whether larger effective training (ηt) enables higher-frequency components to emerge. We train for 100,000 steps with $\eta = 1.0$ for widths $m \in \{2, 4, 6, 8, 10, 100, 1000, 10000\}$.



(a) RelErr vs width (longer training, higher LR).



(b) Overlays across widths (longer effective time).



(c) High-width overlays approach the NTK predictor.

Figure 3.5: Increasing effective training ηt reveals late-learning of high-frequency structure (EXP001).

Interpretation. These results suggest that the earlier "low-frequency trapping" was largely a horizon/step-size effect: high-frequency components correspond to smaller eigenvalues of the relevant kernel operator and therefore converge much more slowly. Wider networks can approximate these components, but doing so may require substantially larger effective training time ηt.

3.2.6 Discrete-time kernel dynamics as an explanation

The preceding observations are consistent with the discrete-time kernel dynamics reviewed in Section 2.1. Under the approximation that the empirical kernel is effectively constant ($K_t \approx K$), the residual $r_t = f_t - y$ approximately follows

$$r_{t+1} \approx (I - \eta K)\, r_t, \qquad \text{hence} \qquad r_t \approx (I - \eta K)^t r_0.$$

Decomposing in the eigenbasis of K shows that each mode contracts as $(1 - \eta\lambda_j)^t$. Modes with small λ_j therefore require much larger effective time ηt to decay, providing a direct mechanism for the delayed emergence of higher-frequency structure (spectral bias) observed in Figures 3.3 and 3.5.

3.3 EXP002: Empirical NTK drift and the onset of a lazy regime

This experiment suite investigates when (and whether) finite-width networks enter a regime where the empirical NTK is effectively frozen, and whether that freezing coincides with (i) a plateau in the loss and (ii) kernel-regression behavior using the frozen NTK.

3.3.1 Kernel drift and a slope-based freeze-time criterion

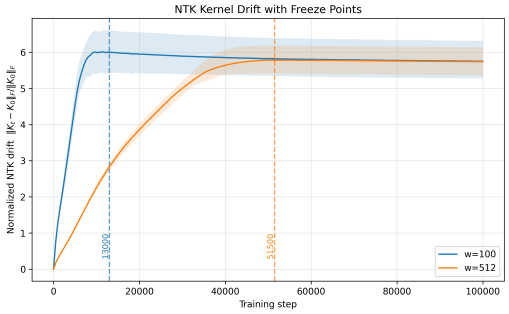
We compute the training-set NTK $K(t) = K_{\text{train,train}}(t)$ at snapshot steps and track

$$\Delta_K(t) = \frac{\|K(t) - K(0)\|_F}{\|K(0)\|_F}.$$

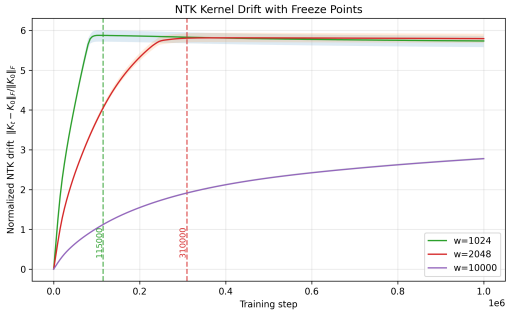
To detect when the kernel has effectively stopped evolving, we monitor the finite-difference slope

$$s(t) \;=\; \frac{\Delta_K(t) - \Delta_K(t-\Delta)}{\text{step}(t) - \text{step}(t-\Delta)},$$

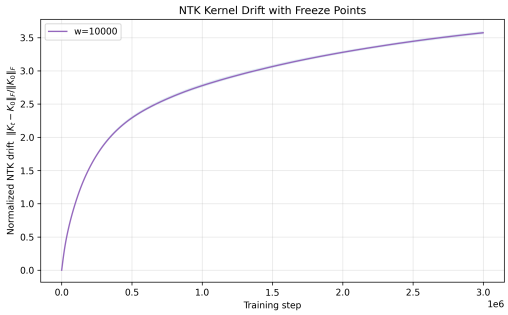
and define a freeze time t_{freeze} as the earliest t such that $|s(t)| < \varepsilon |s(0)|$ for k consecutive snapshot intervals (with fixed ε, k in the implementation).



(a) Low widths.



(b) Mid widths.



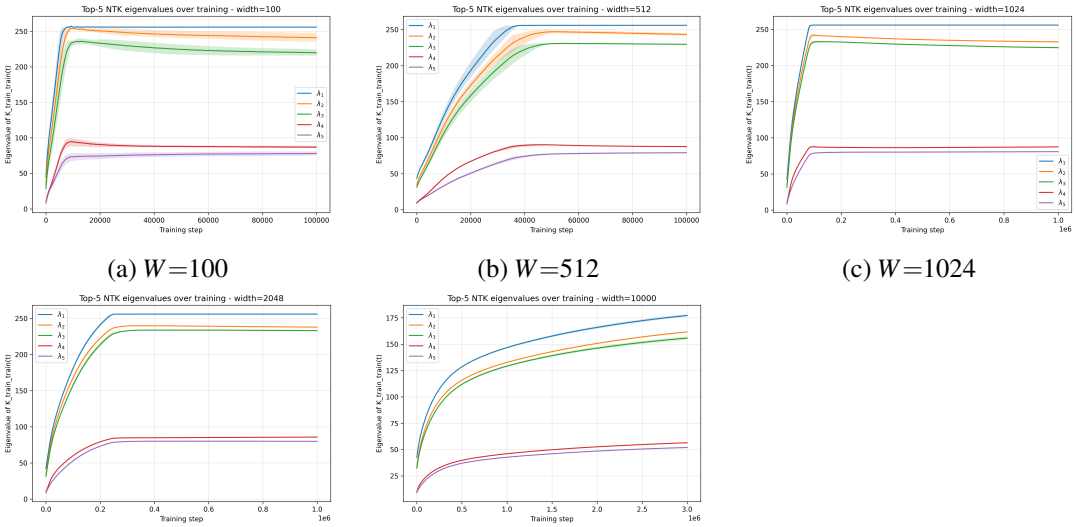
(c) High width ($W = 10000$).

Figure 3.6: Normalized NTK drift curves across widths with a slope-based freeze-time criterion (EXP002).

Observation. Wider networks freeze later under this criterion: narrow networks reach a negligible drift slope quickly, while the widest model does not satisfy the freeze criterion within the available training horizon. This is compatible with the empirical observation that kernel evolution per optimization step becomes smaller as width increases (in NTK-like scalings), hence longer training is needed to observe comparable drift.

3.3.2 Eigenvalue drift and width-dependent stabilization timescales

Let $K(t) = U(t)\Lambda(t)U(t)^\top$ be the eigendecomposition at snapshot time t. We track the top eigenvalues $\lambda_1(t), \ldots, \lambda_5(t)$.



(a) $W=100$ (b) $W=512$ (c) $W=1024$

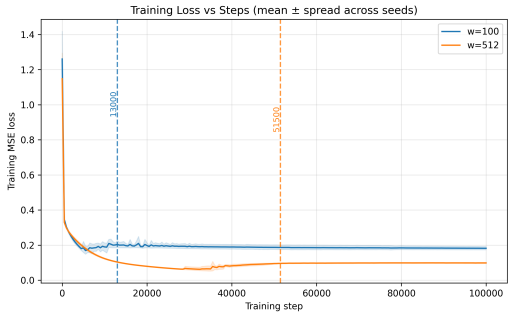
(d) $W=2048$ (e) $W=10000$

Figure 3.7: Top eigenvalue trajectories of the empirical NTK across widths (EXP002).

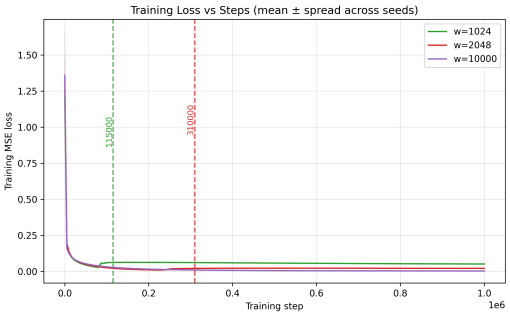
Observation. Across widths, the leading eigenvalues appear to saturate toward similar plateau values, but the timescale of this stabilization is strongly width-dependent: narrow networks stabilize quickly, while wide networks evolve more slowly. This suggests that width changes the *rate* at which the spectrum evolves, even when the eventual top-eigenvalue scale is comparable.

3.3.3 Loss evolution versus kernel drift

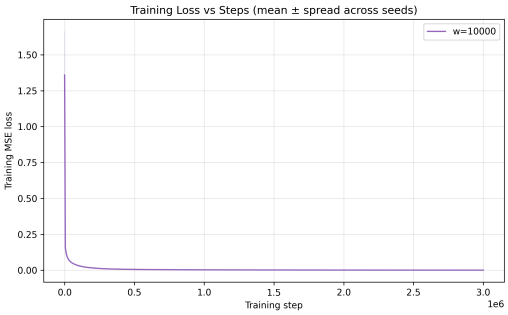
We record the training loss and compare it to the drift and drift-slope signals.



(a) Low widths.



(b) Mid widths.



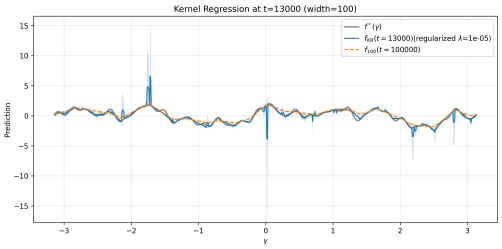
(c) High width.

Figure 3.8: Training loss evolution across widths (EXP002).

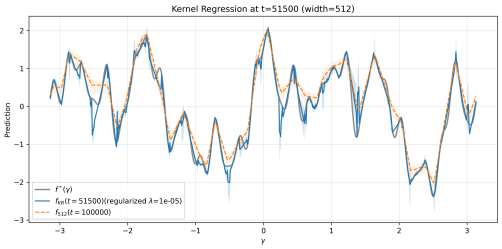
Observation. In these runs, the loss can plateau earlier than the time at which the drift-slope criterion declares the kernel frozen, especially for large widths. This motivates treating "kernel freezing" and "optimization slowing" as related but distinct signals.

3.3.4 Kernel regression at freeze time and comparison to network predictions

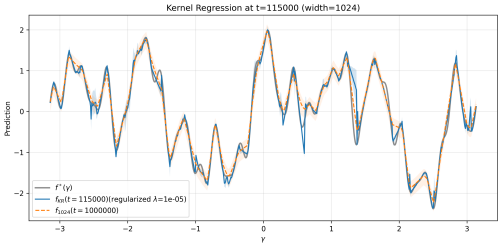
After estimating a freeze time $t^\star$, we compute the kernel-regression predictor $f_{\mathrm{KR}}(\cdot;t^\star)$ using the frozen NTK at $t^\star$ and compare it to (i) the network prediction at $t^\star$ and (ii) the final network prediction.



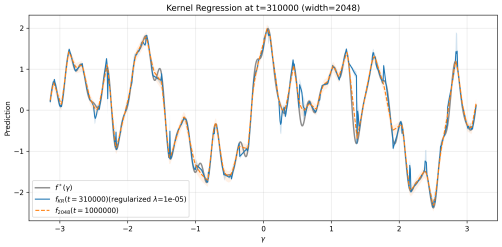
(a) $W=100$



(b) $W=512$

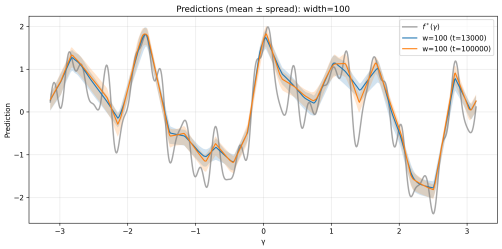


(c) $W=1024$

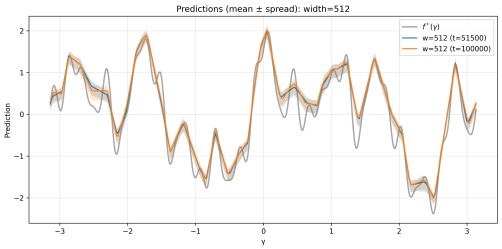


(d) $W=2048$

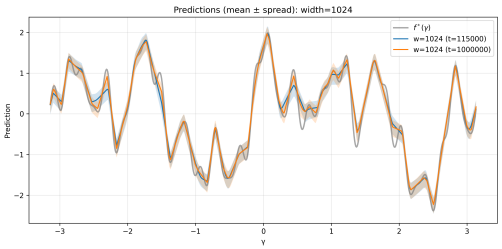
Figure 3.9: Kernel-regression predictions using the frozen NTK at the detected freeze time (EXP002).



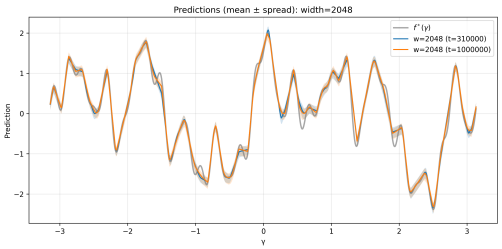
(a) $W=100$



(b) $W=512$



(c) $W=1024$



(d) $W=2048$

Figure 3.10: Network predictions at freeze time vs final time across widths (EXP002).

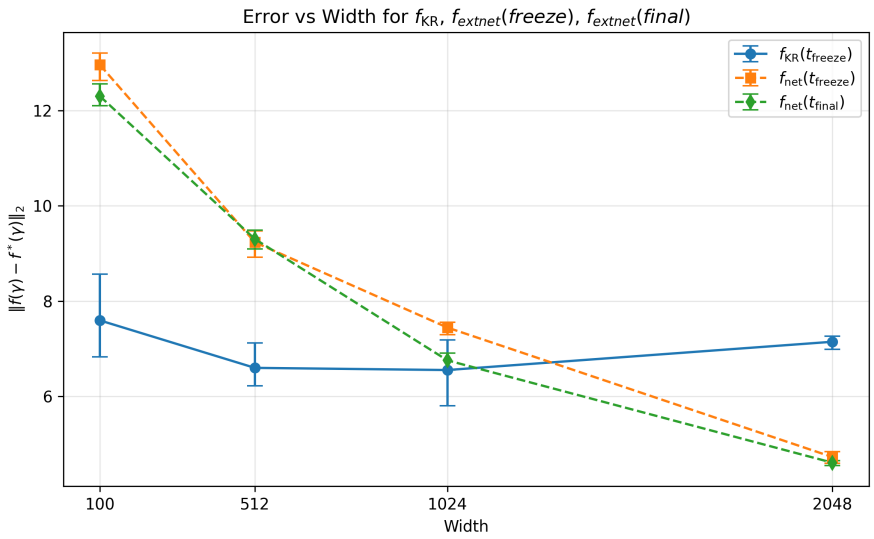


Figure 3.11: Kernel-regression error and final network error versus width. Narrow networks show lower kernel-regression error than final-network error, while for wide networks the final network is closer to the target than kernel regression at the detected freeze time.

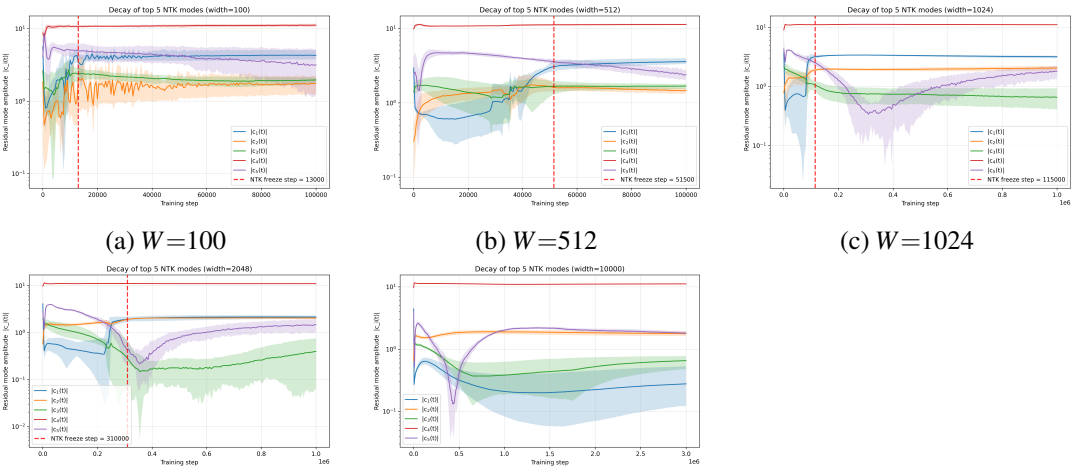
Observation. The relationship between "frozen-kernel regression" and the trained network depends on width: for narrow networks, the kernel-regression predictor at $t^\star$ can be closer to the target than the final network, whereas for wide networks the final trained network can outperform the kernel-regression predictor computed at $t^\star$. At a minimum, this indicates that (i) a drift-based freeze criterion may be too conservative or too late/early depending on width, and (ii) small kernel drift in Frobenius norm does not necessarily guarantee that subsequent function-space evolution is negligible.

3.4 EXP003: Mode-wise decay of residuals (NTK eigenmodes and Fourier components)

This note introduces two complementary mode-wise diagnostics: projections onto (i) leading empirical NTK eigenmodes and (ii) the known Fourier components of the target mixture.

3.4.1 Residual projections onto NTK eigenmodes

At each snapshot time t we compute $K(t) = U(t)\Lambda(t)U(t)^\top$ and project the training residual $r_t = f_t - y$ onto the top eigenvectors. A dashed line indicates the freeze time estimated from EXP002.



(a) $W=100$ (b) $W=512$ (c) $W=1024$

(d) $W=2048$ (e) $W=10000$

Figure 3.12: Residual projections onto the top empirical NTK eigenmodes across widths (EXP003).

Observation. Across widths, the leading mode amplitudes can exhibit non-monotone behavior (dips, rebounds, and long drifts), and do not "collapse" immediately after the freeze time. This suggests that a small drift in $\|K(t)-K(0)\|_F$ may coexist with meaningful changes in how the residual aligns with the instantaneous eigenspaces of $K(t)$ (e.g. through slow rotations of eigenvectors or subtle spectrum changes).

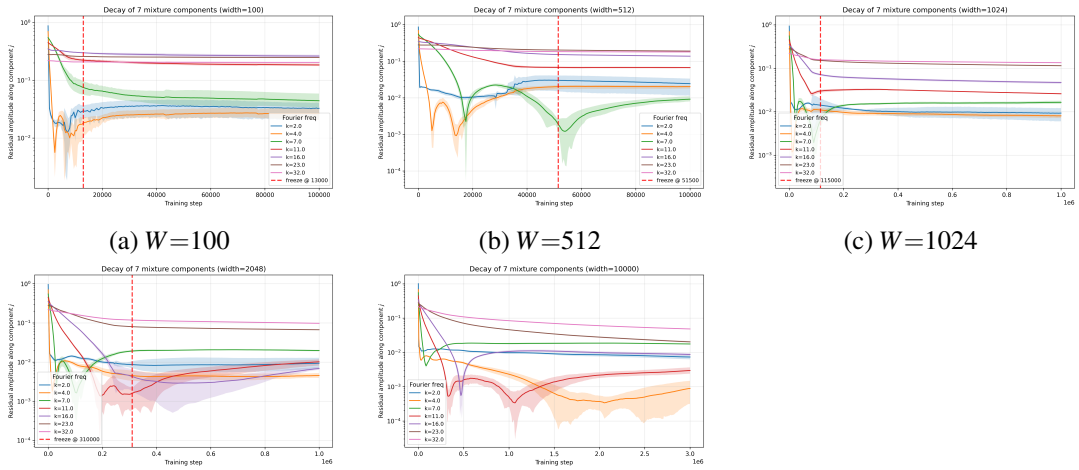
3.4.2 Residual projections onto Fourier-mixture components

Let the Fourier-mixture components be

$$b_j(\gamma) = \sin(K_j\gamma+\phi_j), \qquad K_j \in \{2,4,7,11,16,23,32\}.$$

We track the residual projection coefficients

$$c_j(t) = \frac{\langle r_t, b_j\rangle}{\langle b_j, b_j\rangle}.$$



(a) $W=100$ (b) $W=512$ (c) $W=1024$

(d) $W=2048$ (e) $W=10000$

Figure 3.13: Residual projections onto Fourier-mixture components across widths (EXP003).

Observation. Low-frequency components ($k = 2, 4$) decay early, whereas high-frequency components (e.g. $k \geq 16$) decay much more slowly. This provides a direct empirical signature of spectral bias and supports the "small-eigenvalue modes converge late" mechanism predicted by kernel dynamics (Section 2.3).

3.5 Takeaways for the main experimental study

- **Width versus effective training:** on multi-frequency targets, increasing width alone may not visibly improve agreement with the NTK predictor under a fixed horizon; larger effective training ηt is critical for reducing slow (small-eigenvalue) modes.
- **Kernel drift is informative but not definitive:** a small drift in $\|K(t) - K(0)\|_F$ does not necessarily imply that function-space evolution has essentially stopped. Mode-wise diagnostics can reveal continued dynamics even when drift appears to plateau.
- **Mode-wise views are essential:** tracking residual projections onto NTK eigenmodes and/or known Fourier components cleanly exposes spectral bias and helps separate "capacity" effects (very small width) from "slow convergence" effects (small eigenvalues / limited ηt).

Bibliography

Araújo, D., Oliveira, R. I., and Yukimura, D. (2019). A mean-field limit for certain deep neural networks.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks.

Bartolucci, F., Vito, E. D., Rosasco, L., and Vigogna, S. (2021). Understanding neural networks with reproducing kernel banach spaces.

Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., and Kritchman, S. (2020). Frequency bias in neural networks for input of non-uniform density.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Bietti, A. and Mairal, J. (2019). On the inductive bias of neural tangent kernels.

Bowman, B. and Montufar, G. (2022). Spectral bias outside the training set for deep networks in the kernel regime.

Cao, Y., Fang, Z., Wu, Y., Zhou, D.-X., and Gu, Q. (2020). Towards understanding the spectral bias of deep learning.

Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport.

Chizat, L., Colombo, M., Fernández-Real, X., and Figalli, A. (2024). Infinite-width limit of deep linear neural networks. *Communications on Pure and Applied Mathematics*, 77(10):3958–4007.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks.

Golikov, E. (2025). *Deep Neural Networks: Large-Width Behavior and Generalization Bounds*. PhD thesis, EPFL, Lausanne.

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

Jacot, A., Gabriel, F., and Hongler, C. (2020). Neural tangent kernel: Convergence and generalization in neural networks.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2020). Wide neural networks of any depth evolve as linear models under gradient descent $*$. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124002.

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

Nguyen, P.-M. (2019). Mean field limit of the learning dynamics of multilayer neural networks.

Nguyen, P.-M. and Pham, H. T. (2023). A rigorous framework for the mean field limit of multilayer neural networks.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks.

Rotskoff, G. and Vanden-Eijnden, E. (2022). Trainability and accuracy of artificial neural networks: An interacting particle system approach. *Communications on Pure and Applied Mathematics*, 75(9):1889–1935.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks.

Sirignano, J. and Spiliopoulos, K. (2019). Mean field analysis of neural networks: A law of large numbers.

Tu, Z., Aranguri, S., and Jacot, A. (2024). Mixed dynamics in linear networks: Unifying the lazy and active regimes.

Varadarajan, V. S. (1958). On the convergence of sample probability distributions. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 19(1/2):23–26.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization.

Appendix A

Glossary

In this appendix we give an overview of frequently used terms and abbreviations.

foo: ...

bar: ...