Understanding the dynamics of the frequency bias in neural networks

Juan Molina

Mircea Petrache

Francisco Sahli Costabal

Matías Courdurier

Pontificia Universidad Católica de Chile

{jjmolina1,mpetrache,fsahli1,mcourdurier}@uc.cl

Abstract

Recent works have shown that traditional Neural Network (NN) architectures display a marked frequency bias in the learning process. Namely, the NN first learns the low-frequency features before learning the high-frequency ones. In this study, we rigorously develop a partial differential equation (PDE) that unravels the frequency dynamics of the error for a 2-layer NN in the Neural Tangent Kernel regime. Furthermore, using this insight, we explicitly demonstrate how an appropriate choice of distributions for the initialization weights can eliminate or control the frequency bias. We focus our study on the Fourier Features model, an NN where the first layer has sine and cosine activation functions, with frequencies sampled from a prescribed distribution. In this setup, we experimentally validate our theoretical results and compare the NN dynamics to the solution of the PDE using the finite element method. Finally, we empirically show that the same principle extends to multi-layer NNs.

1 Introduction

An interesting open question for Deep Learning (DL) models is to find an informative way to correctly describe their learning dynamics. In the absence of well documented ad-hoc training strategies, practitioners are obliged to use architecture search heuristics and empirical hyperparameter tuning, to improve the predictive accuracy of Neural Networks (NNs), especially for finer (high-frequency) details in the data. In view of both cost-efficiency and environmental impact of DL training, there is an increasing need for a theoretical understanding of learning dynamics, which may lead to concise practices to improve the accuracy and efficiency in training of NNs. So far, multiple phenomena regarding the learning process have been studied:

Simplicity Bias. In DL models, it has been observed that the "simplest" features tend to be learned at earlier stages than more complicated features. This is the so-called *simplicity bias* of DL models. The phenomenon has been first identified in basic architectures to frame different blunders for efficient learning [3, 13, 26, 34, 48], and beyond DL, being related to Occam's razor principle, it has also been proposed as a plausible structural element of generic learning processes [9, 10, 33, 34]. In DL, this phenomenon has been connected to the regularization aspect of stochastic gradient descent [16], and to the subdivision in two phases of memorization/fine-tuning in the so-called Information Bottleneck Principle [8, 38].

Spectral Bias / Frequency Principle. A simple yet flexible way to quantify the simplicity bias is the so-called *Frequency Principle* [19, 42], also known as *Spectral (or Frequency) Bias* [27], which refers to the experimentally observed tendency of NNs to learn lower frequency features earlier than higher

ones. This phenomenon has been empirically proven in studies such as [44, 45]. Furthermore, the existence of frequency bias has been investigated using various techniques. For instance, shallow NNs with tanh activation functions were analyzed in [42], while [7, 32] explored frequency bias through the lens of eigenvalues theory and certain data hypotheses. Additionally, [31] provides a qualitative approach to avoiding frequency bias by assuming the existence of optimal parameters. Similarly, [20, 21, 49] gives clues on how to mitigate frequency bias under the infinite-width condition of hidden layers. The theory behind the frequency bias has been applied to enhance physics-informed NNs (PINNs) [40], to study overfitting [43, 49], and to adjust gradient descent methods [11, 12].

Fourier Features. Implicit Neural Representations, a.k.a. Neural Fields, are an emerging research area where multiple techniques to mitigate the frequency bias have been developed. The goal of these architectures is to accurately approximate continuous signals $f: \mathbb{R}^d \to \mathbb{R}^l$, such as images, videos, complex 3D shapes, sensor measurements, etc., with fully connected NNs for tasks such as reconstruction, superresolution, editing, etc. [25, 41]. Given that many of the signals of interest in this field present high-frequency components that should be learned by the NNs, several empirical solutions have been proposed to alleviate the frequency bias. For instance, using sinusoidal or Gaussian activation functions, allowed increased accuracy in multiple applications [30, 36]. Fourier features (FF), an ad-hoc type of positional encoding, is one of the most successful approaches to control the frequency bias in Neural Fields [37]. Here, an input coordinate is preprocessed in the first layer by projecting it onto a high-dimensional vector consisting of cosine and sine of random frequencies. This idea was previously introduced and utilized in the field of kernel machines [28, 35], transformers [39], and more general architectures [4, 29, 50].

Neural Tankent Kernel regime. Despite its success, there is limited theoretical understanding on how the hyperparameters of FF affect the frequency bias and the training process. In this work, we study the FF architecture to give a rigorous description of frequency bias dynamics, by working in the Neural Tangent Kernel regime (NTK), in which the learning dynamics are studied in the infinite-width limit of NNs layers [2, 15, 17]. The NTK is one of the most prominent tools to study learning process in NNs, as it allows to obtain a deterministic kernel representation of the training error, leading to a natural tool to analyze frequency bias [37].

Related Work. The NTK for FF architectures has been used as an initial formalization of the model [37]. Successively, NTK has been used to study the frequency principle in recent works [21, 47], including approaches using replica symmetry [5]. The latter thesis also includes a study of frequency bias via eigenvalues of the NTK kernel, including some aspects of the dynamics. Heuristics about the influence of activation functions over frequency bias (without using NTK) follow from computations in [14], and the influence towards spectral bias of initialization choices under stochastic gradient descent is discussed in [23]. However, none of these previous studies provides a detailed enough result on the dynamics of the frequency for a FF architecture, in order to directly compute the influence of initialization distribution on the dynamics.

Main contributions

- Based on the NTK, we give a rigorous derivation of the Partial Differential Equation (PDE) describing the frequency dynamics of errors of a FF neural network with one hidden layer.
- Based on the above equation we provide a novel understanding about the precise influence on Frequency Bias of the initialization distributions of the weights.
- Experimentally, we verify that a Finite Element simulation of our model's PDEs qualitatively reproduces the spectral bias of a NN. We also show that our results still hold for multilayer NNs.

2 Problem setup

In this section, we introduce our NN model and the dynamics that follows during training in order to approximate a target function.

Let $\widetilde{f}:\mathbb{R}^d\to\mathbb{R}$ be our target function such that $\widetilde{f}\in C\left(\mathbb{R}^d\right)\cap L^2\left(\mathbb{R}^d\right)$. Our neural network is defined via a parameterized approximate of \widetilde{f} , denoted $f(\boldsymbol{x},\Theta(t)):\mathbb{R}^d\times\mathbb{R}^{(d+1)\times m}\to\mathbb{R}$ depending on parameter $\Theta:[0,T]\to\mathbb{R}^{(d+1)\times m}$. We take this parameterization to be of the following specific

form, depending on a nonlinear point-wise activation function $g: \mathbb{R} \to \mathbb{R}$:

$$f(\boldsymbol{x}, \Theta(t)) = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} a_k(t) g(\boldsymbol{w}_k(t) \cdot \boldsymbol{x}), \qquad (1)$$

where $a(t) = (a_1(t), ..., a_m(t)) \in \mathbb{R}^m$, $\mathbf{W}(t) = (\mathbf{w}_1(t), ..., \mathbf{w}_m(t)) \in \mathbb{R}^{d \times m}$ and $\Theta(t) = \{a(t), W(t)\}$.

To train the NN, we consider square error loss $\mathcal{L}(\boldsymbol{x},f) := \frac{1}{2}(f(\boldsymbol{x},\Theta(t)) - \widetilde{f}(\boldsymbol{x}))^2$, and to model stochastic gradient descent we average it over random i.i.d. batches $S := \{\boldsymbol{x}_i\}_{i=1}^N$ from the training dataset, i.e. we minimize the empirical risk

$$\mathcal{R}_S(\Theta) := rac{1}{N} \sum_{i=1}^N \mathcal{L}(oldsymbol{x}_i, f) = rac{1}{2N} \sum_{i=1}^N \left(f(oldsymbol{x}_i, \Theta) - \widetilde{f}(oldsymbol{x}_i)
ight)^2.$$

The dynamics of the parameter Θ is approximated by continuous gradient descent of the empirical risk \mathcal{R}_S , that is,

$$\frac{d}{dt}\Theta(t) = -\nabla_{\Theta} \mathcal{R}_{S}(\Theta(t)) = -\frac{1}{N} \sum_{i=1}^{N} \nabla_{\Theta} f(\boldsymbol{x}_{i}, \Theta(t)) \left(f(\boldsymbol{x}_{i}, \Theta(t)) - \widetilde{f}(\boldsymbol{x}_{i}) \right),$$

and the evolution of the NN is modeled by the solution of the ordinary differential equation:

$$\frac{d}{dt}f(\boldsymbol{x},\Theta(t)) = \nabla_{\Theta}f(\boldsymbol{x},\Theta(t)) \cdot \frac{d}{dt}\Theta(t)
= -\frac{1}{N} \sum_{i=1}^{N} \nabla_{\Theta}f(\boldsymbol{x},\Theta(t)) \cdot \nabla_{\Theta}f(\boldsymbol{x}_{i},\Theta(t)) \left(f(\boldsymbol{x}_{i},\Theta(t)) - \widetilde{f}(\boldsymbol{x}_{i})\right)
= -\frac{1}{N} \sum_{i=1}^{N} K_{m}(\boldsymbol{x},\boldsymbol{x}_{i};t) \left(f(\boldsymbol{x}_{i},\Theta(t)) - \widetilde{f}(\boldsymbol{x}_{i})\right),$$
(2)

where we introduced a time-dependent kernel $K_m : \mathbb{R}^d \times \mathbb{R}^d \times [0,T] \to \mathbb{R}$, defined as follows (with notation as in (1))

$$K_{m}(\boldsymbol{x}, \boldsymbol{x'}; t) := \nabla_{\Theta} f(\boldsymbol{x}, \Theta) \cdot \nabla_{\Theta} f(\boldsymbol{x'}, \Theta)$$

$$= \sum_{k=1}^{m} \frac{\partial f}{\partial a_{k}}(\boldsymbol{x}, \Theta) \cdot \frac{\partial f}{\partial a_{k}}(\boldsymbol{x'}, \Theta) + \sum_{k=1}^{m} \frac{\partial f}{\partial \boldsymbol{w}_{k}}(\boldsymbol{x}, \Theta) \cdot \frac{\partial f}{\partial \boldsymbol{w}_{k}}(\boldsymbol{x'}, \Theta)$$
(3)

2.1 Assumption: Infinite width regime

In this work, we study the training dynamics under the NTK regime [2, 15, 17]. This means with have two assumptions:

- (i) The initialization weights $\{\boldsymbol{w}_k(0)\}_{k=1}^m$ are i.i.d. random variables drawn from $\rho_{\boldsymbol{w}}$ with zero mean and covariance matrix $\operatorname{Var}[\boldsymbol{w}(0)] = \sigma_{\boldsymbol{w}}^2 I_{d \times d}$, and the parameters $\{a_k(0)\}_{k=1}^m$ are taken to be i.i.d. random variables drawn from the distribution ρ_a with mean zero and finite variance σ_a^2 . Note that the normalization is nonstandard from a probabilistic point of view.
- (ii) The hidden layer dimension m tends to infinity.

Then the kernel $K_m(\boldsymbol{x}, \boldsymbol{x'}; t)$ from (3) can be approximated with high probability by the continuum kernel K defined as:

$$K(\boldsymbol{x}, \boldsymbol{x'}) := \mathbb{E}_{\theta \sim \rho_a \times \rho_w} \left[\nabla_{\theta} f^0(\boldsymbol{x}, \theta) \cdot \nabla_{\theta} f^0(\boldsymbol{x'}, \theta) \right]$$
(4)

where $f^0(\boldsymbol{x}, \theta) := a \ g(\boldsymbol{w} \cdot \boldsymbol{x})$ for $\theta = (a, \boldsymbol{w})$. In other words, NTK theory says that

$$K_m(\boldsymbol{x}, \boldsymbol{x'}; t) \stackrel{(m \to \infty)}{\simeq} K(\boldsymbol{x}, \boldsymbol{x'}).$$
 (5)

Note that the whole NTK evolution is determined by $\rho_a \times \rho_w$, the distribution from which $\theta_k = (a_k, w_k), k = 1, \dots, m$ are i.i.d.-sampled at initialization.

From now on, we work with the limit "tangent" kernel K from (4) instead of K_m . In particular, note that K only depends on the initialization distributions ρ_a, ρ_w and the activation function g.

2.2 Dynamics of the error

Summarizing, under these assumptions we can approximate the dynamics of the NN by:

$$\frac{d}{dt}\left(f(\boldsymbol{x},\Theta(t)) - \widetilde{f}(\boldsymbol{x})\right) = -\frac{1}{N}\sum_{i=1}^{N}K(\boldsymbol{x},\boldsymbol{x}_i)\left(f(\boldsymbol{x}_i,\Theta(t)) - \widetilde{f}(\boldsymbol{x}_i)\right). \tag{6}$$

that can be conveniently approximated using the Central Limit Theorem by:

$$\frac{du}{dt}(\boldsymbol{x}, \Theta(t)) = -\int_{\mathbb{R}^d} K(\boldsymbol{x}, \boldsymbol{x'}) u(\boldsymbol{x'}, \Theta(t)) d\rho_{\text{data}}(\boldsymbol{x'}), \tag{7}$$

where $u(x, \Theta(t)) = f(x, \Theta(t)) - \widetilde{f}(x)$ and $\rho_{\text{data}}(x)$ is the density of the data distribution. This representation is called linearized residual dynamics, introduced in [24].

3 Derivation of the frequency bias equation

Now, using equation (7) we will deduce the frequency dynamics. These Fourier computations are formulated in the framework of tempered distributions. Let $\mathcal{S} := \mathcal{S}(\mathbb{R}^d)$ be the Schwartz space on \mathbb{R}^d and $\mathcal{S}' := \mathcal{S}'(\mathbb{R}^d)$ be the space of tempered distributions. The distributional Fourier transform is defined as

$$\langle \mathcal{F}[h], \phi \rangle = \langle h, \mathcal{F}[\phi] \rangle \tag{8}$$

where $h \in \mathcal{S}', \phi \in \mathcal{S}$ and \mathcal{F} is the usual Fourier transform defined as

$$\widehat{h}(\boldsymbol{\xi}) = \mathcal{F}[h](\boldsymbol{\xi}) := \int_{\mathbb{R}^d} h(\boldsymbol{x}) e^{-2\pi i \boldsymbol{x} \cdot \boldsymbol{\xi}} d\boldsymbol{x}$$
(9)

Theorem 3.1 (proved as Thm. A.4). Under the assumptions 2.1, and assuming that $g \in H^1$, the dynamics (7) can be expressed in frequency space (in the sense of distributions, i.e. in duality with an arbitrary Schwarz test function $\psi \in \mathcal{S}(\mathbb{R}^d)$) as:

$$\left\langle \frac{d}{dt}\widehat{u}, \psi \right\rangle = -\frac{\sigma_a^2}{4\pi^2} \int_{\mathbb{R}^d} \overline{\nabla \psi(\boldsymbol{\xi})} \cdot \int_{\mathbb{R}^2} \nabla \widehat{u}_{\rho_{data}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\widehat{g'}(y') \widehat{g'}(y)}{|y|^d} dy' dy d\boldsymbol{\xi}$$

$$- \int_{\mathbb{R}^d} \overline{\psi(\boldsymbol{\xi})} \int_{\mathbb{R}^2} \widehat{u}_{\rho_{data}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\overline{\widehat{g}(y')} \widehat{g}(y)}{|y|^d} dy' dy d\boldsymbol{\xi}.$$

$$(10)$$

where $\hat{u}_{\rho_{data}} = \mathcal{F}[u\rho_{data}].$

This initial result (10) sheds light on how the dynamics of the error frequency \widehat{u} depend on five aspects: the dimension d, the Fourier transform of the activation function g and its derivative g', the initial parameter distribution w, and the variance of the parameter a. However, interpreting this equation (10) proves challenging since it involves an integral operator combining the different frequencies through an integral. Thus, to give a more interpretable explanation of the frequency dynamics we apply this theorem to the specific case of FF.

4 Dynamics of frequency bias for the Fourier features model

To approximate the model of Fourier Features, we define our neural network as

$$f(\boldsymbol{x}, \Theta(t)) = \frac{1}{\sqrt{2m}} \sum_{k=1}^{m} a_k(t) g_1\left(\boldsymbol{w}_k(t) \cdot \boldsymbol{x}\right) + \frac{1}{\sqrt{2m}} \sum_{k=1}^{m} b_k(t) g_2\left(\boldsymbol{w}_k(t) \cdot \boldsymbol{x}\right)$$
(11)

here $\Theta(t) = \{a(t), b(t), W(t)\}$, where $a \in \mathbb{R}^m$, $b \in \mathbb{R}^m$ and $\mathbf{W}(t) = (\mathbf{w}_1(t), ..., \mathbf{w}_m(t))^T \in \mathbb{R}^d \times \mathbb{R}^m$ are the trainable parameters with initial distribution ρ_a , ρ_b and ρ_w respectively (See figure 1). In this case, the same methodology and analysis as above can be made, where the only difference is the expression of kernel K, which is now given by

$$K(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}_{\theta} \left[a^{2} g'_{1} \left(\boldsymbol{w} \cdot \boldsymbol{x} \right) g'_{1} \left(\boldsymbol{w} \cdot \boldsymbol{x'} \right) \boldsymbol{x} \cdot \boldsymbol{x'} \right] + \mathbb{E}_{\theta} \left[b^{2} g'_{2} \left(\boldsymbol{w} \cdot \boldsymbol{x} \right) g'_{2} \left(\boldsymbol{w} \cdot \boldsymbol{x'} \right) \boldsymbol{x} \cdot \boldsymbol{x'} \right]$$

$$+ \mathbb{E}_{\theta} \left[g_{1} \left(\boldsymbol{w} \cdot \boldsymbol{x} \right) g_{1} \left(\boldsymbol{w} \cdot \boldsymbol{x'} \right) \right] + \mathbb{E}_{\theta} \left[g_{2} \left(\boldsymbol{w} \cdot \boldsymbol{x} \right) g_{2} \left(\boldsymbol{w} \cdot \boldsymbol{x'} \right) \right]$$

$$(12)$$

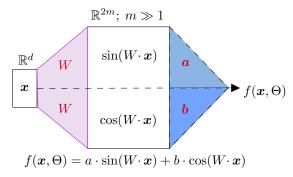


Figure 1: Neural network of 2 layers with Fourier Features model.

Corollary 4.1 (Proved as Cor. B.1). Under the assumptions 2.1. If the NN is given by equation (11) where $g_1(x) = \cos(2\pi x)$, $g_2(x) = \sin(2\pi x)$, $\mathbb{E}[a^2] = \mathbb{E}[b^2] = \sigma_a^2$ and ρ_w is an even function, then dynamics (10) in the frequency space is given by

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = \sigma_a^2 \operatorname{div}\left(\rho_{\boldsymbol{w}}(\boldsymbol{\xi})\nabla\widehat{u}_{\rho_{data}}(\boldsymbol{\xi},\Theta(t))\right) - \rho_{\boldsymbol{w}}(\boldsymbol{\xi})\widehat{u}_{\rho_{data}}(\boldsymbol{\xi},\Theta(t)), \qquad \boldsymbol{\xi} \in \mathbb{R}^d, t > 0, (13)$$

with initial condition

$$\widehat{u}_0(\boldsymbol{\xi}) = \widehat{u}(\boldsymbol{\xi}, \Theta(0)) = \mathcal{F}_{\boldsymbol{x} \to \boldsymbol{\xi}} \left[f(\boldsymbol{x}, \Theta(0)) - \widetilde{f}(\boldsymbol{x}) \right]. \tag{14}$$

Remark 4.2. In the setting of a FF neural network with frozen random weight $\mathbf{W}(t) = \mathbf{W}(0)$, equation (13) becomes

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = -\rho_{\boldsymbol{w}}(\boldsymbol{\xi})\widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi},\Theta(t)). \tag{15}$$

which is equation (13) with $\sigma_a = 0$.

From now on, we assume that the sampling is rich enough, and for the ensuing analysis we approximate $\widehat{u}_{\rho_{\text{data}}} \approx \widehat{u}$.

4.1 Qualitative analysis of the frequency dynamics

FF-specific equation (13) corresponds to a damped heat equation and therefore it provides a quite interpretable expression that is easier to understand. Indeed, the roles played by the initial distributions of the parameters are very explicit: the initial distribution ρ_a only influences the dynamics through the value of its variance σ_a^2 , and σ_a^2 only multiplies the magnitude of the diffusion term. The distribution ρ_w appears in the diffusion coefficient and as the magnitude of the damping term, hence, it is the only element introducing frequency dependency in the dynamics. Notably, from equation (13), we can also observe that the input dimension d does not play any role in any frequency dependent behaviour of the dynamics.

4.1.1 The role of the distribution of ρ_w

The form of equation (13) confirms that the NN may exhibit a frequency bias during learning. For small σ_a or frozen weights **W**, the solution to equation (13) is approximated by the solution of equation (15), which is explicitly:

$$\widehat{u}(\boldsymbol{\xi}, \Theta(t)) = \widehat{u}_0(\boldsymbol{\xi})e^{-\rho_{\boldsymbol{w}}(\boldsymbol{\xi})t}.$$
(16)

For further analysis, we define the frequency learning rate $\kappa(\xi)$ as the slope of the learning dynamics at the beginning of the training process, such that:

$$\log |\widehat{u}(\boldsymbol{\xi}, \Theta(t))| \approx -\kappa(\boldsymbol{\xi})t + \log |\widehat{u}_0(\boldsymbol{\xi})|, \quad \text{for } t \sim 0.$$
 (17)

Then, it stems from equation (16) that the learning rate at each frequency is exactly:

$$\kappa(\boldsymbol{\xi}) = \rho_{\boldsymbol{w}}(\boldsymbol{\xi}). \tag{18}$$

Now, if we aim to mitigate the frequency bias, it is necessary for all frequencies to be learned at the same rate. A first natural choice is to take ρ_w to be *constant* across all frequencies. This corresponds to choose a uniform distribution, as done in [36], that covers all the frequencies of the target function \widetilde{f} . Nonetheless, since the frequency content of \widetilde{f} is not know a priory, we run the risk of leaving some frequencies of \widetilde{f} outside of the learnable range. Therefore, it is more robust to choose ρ_w as a normal distribution with a large enough standard deviation, which is always positive. A disadvantage of choosing ρ_w as a normal distribution with a very large standard deviation, is that the frequency learning rate is inversely proportional to the maximum value of ρ_w . Which also arises as an issue when choosing a uniform distribution with a very large support. We will test these observations in the experiment section (see Figure 3)

4.1.2 The role of the variance σ_a^2

The second form of control is indirectly determined by the distribution ρ_a . In this case, the distribution itself is not crucial: *only its variance* σ_a^2 *appears in* (13). Furthermore, σ_a^2 impacts the diffusion term (first term in (13)), and not the damping term.

In [37], where Fourier Features are introduced, it is discussed that training the frequency parameters in the first layer does not yield any benefits, therefore it is preferable to leave this layer untrained. This observation is explained by our findings: the learning dynamics for frozen $\mathbf{W}(t)$ are practically the same as the dynamics in the case of σ_a small, as is mentioned in the Remark 4.2. Nonetheless, when σ_a is large enough, the diffusive term in equation (13) will play a relevant role in the learning dynamics, and there will be noticeable differences with the training under frozen $\mathbf{W}(t)$. In particular, the non-local nature of the diffusion term should help the NN to learn frequencies beyond the initial range allowed by ρ_w . This claim is validated by empirical experimentation (see figure 4). A negative aspect of choosing a large σ_a is that the initial estimation $f(x, \Theta(0))$ will be far from f, therefore requiring more iterations for the NN to achieve a good estimation of f.

5 Numerical Experiments

To validate our results in this section we analyze how the NNs defined in (11) with m=2000 learns the target function given by $\widetilde{f}(x)=\operatorname{round}(\sin((4.2)\pi x))$, where round is the function that returns the closest integer to its argument (see Figure 2).

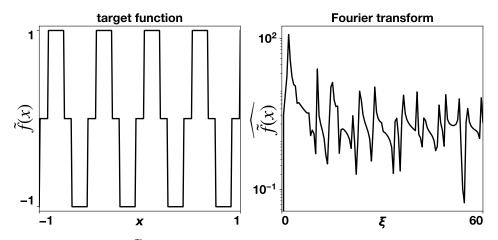


Figure 2: Target function \hat{f} used in our experiments and the magnitude of its Fourier transform.

5.1 NN Training

We generate a training set by evaluating the target function f(x) in an equispaced grid from -1 to 1 using 240 points. The initial parameters are sampled from $\rho_a \sim \mathcal{N}(0, \sigma_a)$ and $\rho_w \sim \mathcal{N}(0, \sigma_w)$ or $\rho_w \sim \mathcal{U}(-R, R)$. The value of σ_a was $2/\sqrt{4000}$ (labeled as 0.03). Additionally, we train without

batching the train set, and we use gradient descent with an optimization learning rate of $10^{-5}/240$. The optimization process is carried out until the 10000th iteration. For our experiments, we train with 100 random initializations and present the results as averages. We use JAX [6] to implement and train the NN and FEniCS [1, 18] to solve the PDE in (13) with the finite element method (FEM). More details regarding the implementation are presented in Appendix C.1. In terms of hardware, we used a personal laptop with an NVIDIA GeForce GTX 3060 with 6GB memory.

5.2 Frequency Learning Rates

For the first experiment, we employed the 100 trained NNs to calculate the frequency learning rate $\kappa(\xi)$ defined in equation (18). We conducted this analysis for both normal and uniform distributions of ρ_w with varying variances (refer to Figure 3).

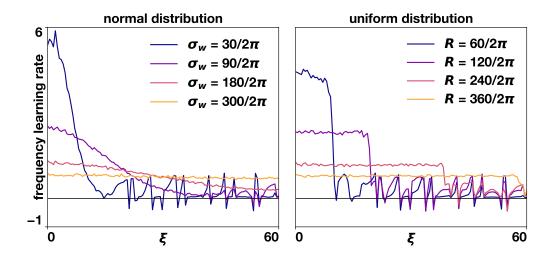


Figure 3: Frequency learning rate $\kappa(\xi)$ for different initialization distributions $\rho_w(\xi)$. Left panel: normal distribution with different standard deviation σ_w , right panel: uniform distribution with different widths R.

Figure 3 validates our results. Indeed, the distribution $\rho_{\boldsymbol{w}}$ is the primary variable for controlling or tuning the frequency bias. For instance, to eliminate spectral bias, the best options are to use a normal distribution with a standard deviation of $\sigma_{\boldsymbol{w}}=300/2\pi$ or larger, or a uniform distribution with R greater than or equal to 60. Furthermore, if the value of $\sigma_{\boldsymbol{w}}$ or R is not large enough, the NN is incapable of learning all frequencies, as demonstrated in the cases with $\sigma_{\boldsymbol{w}}=30/2\pi$ or $\sigma_{\boldsymbol{w}}=90/2\pi$, or more evidently in the case of uniform distribution, where in each case, frequencies above the corresponding value of R are not learned.

The structural noise (oscillations) shown in Figure 3 on both distributions (in the case $\sigma_w=30/2\pi$ and $90/2\pi$ on the Gaussian distribution and $R=60/2\pi, 120/2\pi$ and $240/2\pi$ in the uniform distribution) is due to Discrete Fourier transform trying to approximate the last frequency in the domain of ρ_w . This phenomenon is known as spectral leakage [22] and it is amplified due to the logarithmic scale. For instance, in the case of $\sigma_w=30/2\pi$ (normal distribution), the NN only learns approximately up to the frequency $\xi=15$. Similarly, in the case of uniform distribution, the blue line tells us that the NN is only capable of learning frequencies lower than the frequency $\xi=60/2\pi\approx 9.55$.

5.3 Comparation NN dynamics vs FEM modeling

We perform numerical simulations of the PDE (13) to validate our results. Although Equation (13) is defined over all \mathbb{R} , for the numerical experiments we need to restrict the domain, to $\Omega=(-60,60)$, and impose homogeneous Neumann boundary conditions. With these assumptions, we use FEM for spatial discretization and the backward Euler scheme for time. The FEM basis functions are chosen to be piecewise linear functions with a spatial step size of h=0.5, which take a value of 1 at node i and 0 at the other nodes. The time step is set to 0.1 (see Appendix C for more details).

In the second experiment, we aim to compare the results obtained by NNs with those obtained by FEM simulations when ρ_{w} follows a normal distribution. To do this, we compute the magnitude of the mean of \hat{u} across the 100 initializations. Figure 4 shows that the simulations produced by the FEM (second row) for equation (13) exhibit a similar qualitative behavior to the NNs (first row). Additionally, the figure highlights the diffusive effect of the parameter σ_{a} . Specifically, the values of \hat{u} tend to become smoother, and the NN is capable of improving the learnability of frequencies at which ρ_{w} is small but positive.

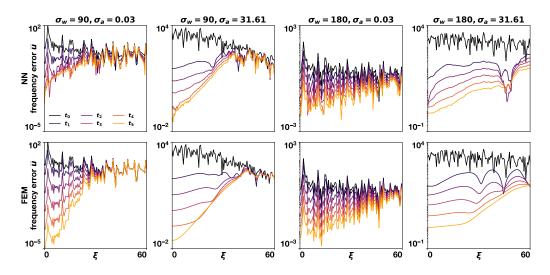


Figure 4: Comparation between FEM simulations of 13 and NN actual dynamics.

5.4 Multilayer case

In addition to our analytical results being based on 2-layer NNs, we aim to demonstrate if these findings also extended to multilayer NNs. The principal assumption that needs to be satisfied is the NTK hypothesis, which posits that the size of the hidden layers tends to infinity. Additionally, the multilayer NN follows the FF model, that is, all the hidden layers have $ReLU(x) = \max(x, 0)$ as an activation function, except the first layer which has cosine and sine activation functions.

To study this setting, we analyze the frequency learning rate $\kappa(\xi)$ given by equation (18) for 3-layer and 4-layer NNs across 100 different initializations of Gaussian distribution (See Figure 5).

Figure 5 provides empirical evidence that our results can be extended to multilayer NNs. Indeed, the figure shows behavior similar to the 2-layer case. In both, the 3-layer and 4-layer NNs, the frequency learning rates resembles half of a Gaussian distribution. However, these plots are not identical to those for 2-layer NNs. This difference is particularly marked in their maximum value.

The results of these experiments are consistent with the findings of [46], which demonstrate that, under the NTK regime, the parameters of the hidden layers tend to remain static. That is, in the infinite-width limit, the hidden layers almost do not change during NN training.

6 Conclusions and future work

We rigorously deduced an equation that models the frequency dynamics of a 2-layer NN in the NTK regime, in terms of directly interpretable quantities. Under the additional hypothesis that the NN follows a Fourier Features architecture, the dynamics is explicitly described as the damped heat equation (13), where the role of the distributions of the initial parameters is understood in a very precise manner. Namely, the frequency bias of the dynamics is mostly determined by the distribution of the parameter ρ_w , through the damping term and the diffusion coefficient, while the variance σ_a^2 of the parameter a, only amplifies or reduces the diffusion term in equation (13).

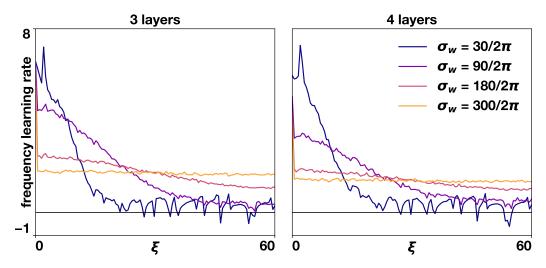


Figure 5: Frequency learning rate $\kappa(\xi)$ on 3-layer and 4-layer neural networks on NTK regime. The hidden layers have all equal widths of 4000.

Our theoretical predictions are then validated with numerical experiments. We find qualitative agreement between the dynamics predicted by the FEM solution of our PDE (13) and the actual FF-NN learning dynamics (figure 4). Also, we very concretely observe the frequency bias behaviour predicted by equation (13) in experimental FF-NN frequency learning rates, for different cases of ρ_w (figure 3). Additionally, the theoretical framework developed here for understanding the dynamics of 2-layer FF-NN, seem to extend to somewhat deeper NN (figure 5).

By understanding how the distributions $\rho_{\boldsymbol{w}}$ and ρ_a affect the behaviour of equation (13), it is possible to predict, and thus eliminate or/and control, frequency bias in the learning dynamics of a FF neural network in the NTK regime. If a target data-set follows a known (or estimated) frequency distribution, a promising direction for future work is to apply our model in order to design specific initial distributions that achieve faster convergence to high accuracy approximation of target functions in specific classes.

Finally, in this work the training of the NN is achieved using gradient descent. Using a different optimization scheme might produce a different learning dynamics, and an interesting continuation of our research is to extend the present analysis to other training mechanisms and develop specific optimization strategies that overcome the issue of spectral bias in NN training.

References

- [1] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. N. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3, 2015. doi: 10.11588/ans.2015.100.20553.
- [2] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.
- [3] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.
- [4] N. Benbarka, T. Höfer, A. Zell, et al. Seeing implicit neural representations as fourier series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2041–2050, 2022.
- [5] B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, 2020.
- [6] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.
- [7] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu. Towards understanding the spectral bias of deep learning. In *30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*, pages 2205–2211. International Joint Conferences on Artificial Intelligence, 2021.
- [8] S. Dahlgaard and J. Evald. Tight hardness results for distance and centrality problems in constant degree graphs. *arXiv* preprint arXiv:1609.08403, 2016.
- [9] K. Dingle, C. Q. Camargo, and A. A. Louis. Input—output maps are strongly biased towards simple outputs. *Nature communications*, 9(1):761, 2018.
- [10] J. Feldman. The simplicity principle in perception and cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(5):330–340, 2016.
- [11] A. Geifman, M. Galun, D. Jacobs, and B. Ronen. On the spectral bias of convolutional neural tangent and gaussian process kernels. *Advances in Neural Information Processing Systems*, 35: 11253–11265, 2022.
- [12] A. Geifman, D. Barzilai, R. Basri, and M. Galun. Controlling the inductive bias of wide neural networks by modifying the kernel's spectrum. *arXiv preprint arXiv:2307.14531*, 2023.
- [13] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv* preprint arXiv:1811.12231, 2018.
- [14] Q. Hong, J. W. Siegel, Q. Tan, and J. Xu. On the activation function dependence of the spectral bias of neural networks. *arXiv preprint arXiv:2208.04924*, 2022.
- [15] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.
- [16] D. Kalimeris, G. Kaplun, P. Nakkiran, B. Edelman, T. Yang, B. Barak, and H. Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in neural information* processing systems, 32, 2019.
- [17] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.
- [18] A. Logg, K.-A. Mardal, G. N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012. doi: 10.1007/978-3-642-23099-8.

- [19] T. Luo, Z. Ma, Z.-Q. J. Xu, and Y. Zhang. Theory of the frequency principle for general deep neural networks. *arXiv preprint arXiv:1906.09235*, 2019.
- [20] T. Luo, Z. Ma, Z. Wang, Z. J. Xu, and Y. Zhang. An upper limit of decaying rate with respect to frequency in linear frequency principle model. In *Mathematical and Scientific Machine Learning*, pages 205–214. PMLR, 2022.
- [21] T. Luo, Z. Ma, Z.-Q. J. Xu, and Y. Zhang. On the exact computation of linear frequency principle dynamics and its generalization. *SIAM Journal on Mathematics of Data Science*, 4(4): 1272–1292, 2022.
- [22] D. A. Lyon. The discrete fourier transform, part 4: spectral leakage. *Journal of object technology*, 8(7), 2009.
- [23] L. E. MacDonald, J. Valmadre, and S. Lucey. On progressive sharpening, flat minima and generalisation. *arXiv preprint arXiv:2305.14683*, 2023.
- [24] S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pages 2388–2464. PMLR, 2019.
- [25] A. Molaei, A. Aminimehr, A. Tavakoli, A. Kazerouni, B. Azad, R. Azad, and D. Merhof. Implicit neural representation in medical imaging: A comparative survey. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2381–2391, 2023.
- [26] D. Morwani, J. Batra, P. Jain, and P. Netrapalli. Simplicity bias in 1-hidden layer neural networks. Advances in Neural Information Processing Systems, 36, 2024.
- [27] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.
- [28] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.
- [29] S. Ramasinghe and S. Lucey. Learning positional embeddings for coordinate-mlps. *arXiv* preprint arXiv:2112.11577, 2021.
- [30] S. Ramasinghe and S. Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *European Conference on Computer Vision*, pages 142–158. Springer, 2022.
- [31] S. Ramasinghe, L. E. MacDonald, and S. Lucey. On the frequency-bias of coordinate-mlps. *Advances in Neural Information Processing Systems*, 35:796–809, 2022.
- [32] B. Ronen, D. Jacobs, Y. Kasten, and S. Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *Advances in Neural Information Processing Systems*, 32, 2019.
- [33] J. Schmidhuber. Discovering problem solutions with low Kolmogorov complexity and high generalization capability. Inst. für Informatik, 1994.
- [34] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.
- [35] A. Sinha and J. C. Duchi. Learning kernels with random features. *Advances in neural information processing systems*, 29, 2016.
- [36] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473, 2020.

- [37] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.
- [38] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In 2015 ieee information theory workshop (itw), pages 1–5. IEEE, 2015.
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [40] S. Wang, H. Wang, and P. Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.
- [41] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.
- [42] Z. J. Xu. Understanding training and generalization in deep learning by fourier analysis. *arXiv* preprint arXiv:1808.04295, 2018.
- [43] Z. J. Xu and H. Zhou. Deep frequency principle towards understanding why deeper learning is faster. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10541–10550, 2021.
- [44] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.
- [45] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao. Training behavior of deep neural network in frequency domain. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26, pages 264–274. Springer, 2019.
- [46] G. Yang and E. J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.
- [47] G. Yang and H. Salman. A fine-grained spectral perspective on neural networks. *arXiv* preprint arXiv:1907.10599, 2019.
- [48] Y. Yang, E. Gan, G. K. Dziugaite, and B. Mirzasoleiman. Identifying spurious biases early in training through the lens of simplicity bias. In *International Conference on Artificial Intelligence and Statistics*, pages 2953–2961. PMLR, 2024.
- [49] Y. Zhang, T. Luo, Z. Ma, and Z.-Q. J. Xu. A linear frequency principle model to understand the absence of overfitting in neural networks. *Chinese Physics Letters*, 38(3):038701, 2021.
- [50] J. Zheng, S. Ramasinghe, and S. Lucey. Rethinking positional encoding. arXiv preprint arXiv:2107.02561, 2021.

A Derivation of spectral bias equation for \hat{u}

The objective of this section is to deduce the frequency dynamics of the two-layer neural network. In summary, under the assumptions 2.1 we arrive at a linearized residual dynamics give by

$$\frac{du}{dt}(\boldsymbol{x}, \Theta(t)) = -\mathbb{E}_{\theta \sim \rho_a \times \rho_w} \left[\int_{\mathbb{R}^d} K_{\theta}(\boldsymbol{x}, \boldsymbol{x'}) u(\boldsymbol{x'}, \Theta(t)) d\rho_{\text{data}}(\boldsymbol{x'}) \right], \tag{19}$$

where $\rho_{\text{data}}(x) := \sum_{i=1}^{N} \delta_{x_i}(x), u(x, \Theta(t)) = f(x, \Theta(t)) - \widetilde{f}(x)$ and

$$K_{\theta}(\boldsymbol{x}, \boldsymbol{x'}) := \nabla_{\theta} f^{0}(\boldsymbol{x}, \theta) \cdot \nabla_{\theta} f^{0}(\boldsymbol{x'}, \theta)$$

$$= \partial_{a} f^{0}(\boldsymbol{x}, \theta) \partial_{a} f^{0}(\boldsymbol{x'}, \theta) + \nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x}, \theta) \cdot \nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x}, \theta). \tag{20}$$

Here,

$$f(x,\Theta) = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} f^{0}(\boldsymbol{x}, \theta_{k}) \quad \text{with} \quad f^{0}(\boldsymbol{x}, \theta_{k}) := a_{k} g(\boldsymbol{w}_{k} \cdot \boldsymbol{x}). \tag{21}$$

for a_k , w_k taken as i.i.d. samples from distributions ρ_a , ρ_w .

The following computations are based on the theory of tempered distributions. Recall that, over \mathbb{R}^d , Schwartz functions are denoted by $\mathcal{S}(\mathbb{R}^d)$ and tempered distributions form the dual space, denoted $\mathcal{S}'(\mathbb{R}^d)$. As customary, we say that a relation holds in the sense of distributions over a space \mathbb{R}^d , provided it holds in duality with Schwartz from $\mathcal{S}(\mathbb{R}^d)$.

Definition A.1. Given a nonzero vector $v \in \mathbb{R}^d$, we define the delta-like distribution $\delta_v \in \mathcal{S}'(\mathbb{R}^d)$, concentrated on the line spanned by v, by requiring

$$\langle \delta_{\boldsymbol{v}}, \phi \rangle = \int_{\mathbb{R}} \phi(y\boldsymbol{v}) dy, \quad \text{for all} \quad \phi \in \mathcal{S}(\mathbb{R}^d).$$
 (22)

The Fourier transforms of functions appearing in (20) can be computed using the next lemma.

Lemma A.2. For any $g \in \mathcal{S}'(\mathbb{R})$ and any $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$, the following hold:

(i) The Fourier transform of $f_1(\mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})$ is the distribution

$$\widehat{f}_1(\boldsymbol{\xi}) = \widehat{g}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^2} \cdot \boldsymbol{\xi}\right) \delta_{\boldsymbol{w}}(\boldsymbol{\xi}). \tag{23}$$

(ii) The Fourier transform of $f_2(\mathbf{x}) = \mathbf{x}g'(\mathbf{w} \cdot \mathbf{x})$ is the distribution

$$\widehat{f}_{2}(\boldsymbol{\xi}) = -\frac{ia}{2\pi} \nabla_{\boldsymbol{\xi}} \left[\widehat{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi} \right) \delta_{\boldsymbol{w}}(\boldsymbol{\xi}) \right]. \tag{24}$$

Proof. See [31] or [21] \Box

Lemma A.3. The dynamics (19) has the following expression in the frequency domain, where $u_{\rho_{data}}(\boldsymbol{x},\Theta) := u(\boldsymbol{x},\Theta)\rho_{data}(\boldsymbol{x})$:

$$\forall \psi \in \mathcal{S}(\mathbb{R}^d), \quad \langle \partial_t \widehat{u}, \psi \rangle = \langle \mathcal{D}[\widehat{u}_{\rho_{data}}], \psi \rangle, \tag{25}$$

where

$$\mathcal{D}[\widehat{u}_{\rho_{data}}](\boldsymbol{\xi}) := -\mathbb{E}_{\theta \sim \rho_a \times \rho_w} \int_{\mathbb{R}^d} \widehat{K}_{\theta} \left(\boldsymbol{\xi}, \boldsymbol{\xi'}\right) \widehat{u}_{\rho_{data}} \left(\boldsymbol{\xi'}, \Theta\right) d\boldsymbol{\xi'} = -\mathbb{E}_{\theta} \left\langle \widehat{K}_{\theta}(\boldsymbol{\xi}, \cdot), \overline{\widehat{u}_{\rho_{data}}}(\cdot) \right\rangle$$
(26)

and, where, in the sense of distributions over $\mathbb{R}^d \times \mathbb{R}^d$, there holds

$$\widehat{K}_{\theta}(\boldsymbol{\xi}, \boldsymbol{\xi'}) := \overline{\mathcal{F}_{\boldsymbol{x'} \to \boldsymbol{\xi'}} \left[\nabla_{\theta} f^{0}(\boldsymbol{x'}, \theta) \right]} (\boldsymbol{\xi'}) \cdot \mathcal{F}_{\boldsymbol{x} \to \boldsymbol{\xi}} \left[\nabla_{\theta} f^{0}(\boldsymbol{x}, \theta) \right] (\boldsymbol{\xi})$$

$$= \overline{\mathcal{F}_{\boldsymbol{x'} \to \boldsymbol{\xi'}} \left[\begin{array}{c} \nabla_{a} f^{0}(\boldsymbol{x'}, \theta) \\ \nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x'}, \theta) \end{array} \right]} (\boldsymbol{\xi'}) \cdot \mathcal{F}_{\boldsymbol{x} \to \boldsymbol{\xi}} \left[\begin{array}{c} \nabla_{a} f^{0}(\boldsymbol{x}, \theta) \\ \nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x}, \theta) \end{array} \right] (\boldsymbol{\xi})$$

$$:= \widehat{K}_{a}(\boldsymbol{\xi}, \boldsymbol{\xi'}) + \widehat{K}_{w}(\boldsymbol{\xi}, \boldsymbol{\xi'}) \tag{27}$$

Proof. See [21]. \Box

Theorem A.4 (cf. Thm. 3.1). *Under the assumptions 2.1, the dynamics (19) can be expressed in frequency space, in the sense of distributions over* \mathbb{R}^d *, as:*

$$\left\langle \frac{d}{dt} \widehat{u}, \psi \right\rangle = -\frac{\sigma_a^2}{4\pi^2} \int_{\mathbb{R}^d} \overline{\nabla \psi(\boldsymbol{\xi})} \cdot \int_{\mathbb{R}^2} \nabla \widehat{u}_{\rho_{data}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \frac{\overline{\widehat{g'}(y')} \widehat{g'}(y)}{|y|^d} dy' dy \, \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) d\boldsymbol{\xi}$$

$$- \int_{\mathbb{R}^d} \overline{\psi(\boldsymbol{\xi})} \int_{\mathbb{R}^2} \widehat{u}_{\rho_{data}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \frac{\overline{\widehat{g}(y')} \widehat{g}(y)}{|y|^d} dy' dy \, \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) d\boldsymbol{\xi}.$$

$$(28)$$

Proof. Extending the computations from [21], we start by writing for $\phi, \psi \in \mathcal{S}(\mathbb{R}^d)$

$$\left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle := \left\langle \left\langle \widehat{K}_{\theta}, \overline{\phi} \right\rangle, \psi \right\rangle = \int_{\mathbb{R}^{2d}} \widehat{K}_{\theta} \left(\boldsymbol{\xi}, \boldsymbol{\xi'} \right) \overline{\psi(\boldsymbol{\xi})} \phi(\boldsymbol{\xi'}) d\boldsymbol{\xi} d\boldsymbol{\xi'}. \tag{29}$$

As $\hat{K}_{\theta} = \hat{K}_{w} + \hat{K}_{a}$ (cf. (27)), we treat the two terms separately. For \hat{K}_{w} we have, using Lemma A.2,

$$\left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle \tag{30}$$

$$= \int_{\mathbb{R}^{2d}} \overline{\mathcal{F}_{\boldsymbol{x}' \to \boldsymbol{\xi}'} \left[\nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x}', \theta) \right]} (\boldsymbol{\xi}') \cdot \mathcal{F}_{\boldsymbol{x} \to \boldsymbol{\xi}} \left[\nabla_{\boldsymbol{w}} f^{0}(\boldsymbol{x}, \theta) \right] (\boldsymbol{\xi}) \overline{\psi(\boldsymbol{\xi})} \phi(\boldsymbol{\xi}') d\boldsymbol{\xi}' d\boldsymbol{\xi}$$

$$= \int_{\mathbb{R}^{2d}} \overline{\mathcal{F}_{\boldsymbol{x}' \to \boldsymbol{\xi}'} \left[a \, \boldsymbol{x}' g'(\boldsymbol{w} \cdot \boldsymbol{x}') \right]} (\boldsymbol{\xi}') \cdot \mathcal{F}_{\boldsymbol{x} \to \boldsymbol{\xi}} \left[a \, \boldsymbol{x} g'(\boldsymbol{w} \cdot \boldsymbol{x}) \right] (\boldsymbol{\xi}) \overline{\psi(\boldsymbol{\xi})} \phi(\boldsymbol{\xi}') d\boldsymbol{\xi}' d\boldsymbol{\xi}$$

$$= \int_{\mathbb{R}^{2d}} \frac{ia}{2\pi} \nabla_{\boldsymbol{\xi}'} \left[\overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi}' \right) \delta_{\boldsymbol{w}} (\boldsymbol{\xi}') \right] \cdot - \frac{ia}{2\pi} \nabla_{\boldsymbol{\xi}} \left[\overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi} \right) \delta_{\boldsymbol{w}} (\boldsymbol{\xi}) \right] \overline{\psi(\boldsymbol{\xi})} \phi(\boldsymbol{\xi}') d\boldsymbol{\xi}' d\boldsymbol{\xi}$$

$$= \frac{a^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \nabla_{\boldsymbol{\xi}'} \left[\overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi}' \right) \delta_{\boldsymbol{w}} (\boldsymbol{\xi}') \right] \phi(\boldsymbol{\xi}') d\boldsymbol{\xi}' \cdot \int_{\mathbb{R}^{d}} \nabla_{\boldsymbol{\xi}} \left[\overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi} \right) \delta_{\boldsymbol{w}} (\boldsymbol{\xi}) \right] \overline{\psi(\boldsymbol{\xi})} d\boldsymbol{\xi}$$

$$= \frac{a^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \delta_{\boldsymbol{w}} (\boldsymbol{\xi}') \overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi}' \right) \nabla_{\boldsymbol{\xi}'} \phi(\boldsymbol{\xi}') d\boldsymbol{\xi}' \cdot \int_{\mathbb{R}^{d}} \delta_{\boldsymbol{w}} (\boldsymbol{\xi}) \overline{g'} \left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^{2}} \cdot \boldsymbol{\xi} \right) \nabla_{\boldsymbol{\xi}} \overline{\psi(\boldsymbol{\xi})} d\boldsymbol{\xi}.$$

Hence.

$$\left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle = \frac{a^2}{4\pi^2} \int_{\mathbb{R}} \overline{\widehat{g'}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^2} \cdot y'\boldsymbol{w}\right)} \nabla \phi(y'\boldsymbol{w}) dy' \cdot \int_{\mathbb{R}} \widehat{g'}\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|^2} \cdot y\boldsymbol{w}\right) \nabla \overline{\psi(y\boldsymbol{w})} dy$$

$$= \frac{a^2}{4\pi^2} \int_{\mathbb{R}^2} \nabla \overline{\psi(y\boldsymbol{w})} \cdot \nabla \phi(y'\boldsymbol{w}) \overline{\widehat{g'}(y')} \widehat{g'}(y) \, dy dy'. \tag{31}$$

Next, taking the expectation over $\theta \sim \rho_a \times \rho_w$, we get:

$$\mathbb{E}_{\theta} \left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle = \frac{\sigma_a^2}{4\pi^2} \int_{\mathbb{R}^d} \int_{\mathbb{R}^2} \nabla \overline{\psi(y\boldsymbol{w})} \cdot \nabla \phi(y'\boldsymbol{w}) \overline{\widehat{g'}(y')} \widehat{g'}(y) \, dy' dy \rho_{\boldsymbol{w}}(\boldsymbol{w}) d\boldsymbol{w}. \tag{32}$$

Next, we use the following change of variable:

$$\boldsymbol{\eta} = y\boldsymbol{w} = y(w_1, \dots, w_d)^T, \quad \left| \det \left(\frac{\partial(w_1, \dots, w_d)}{\partial(\eta_1, \dots, \eta_d)} \right) \right| = \frac{1}{|y|^d},$$

obtaining

$$\mathbb{E}_{\theta} \left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle = \frac{\sigma_{a}^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \int_{\mathbb{R}^{2}} \nabla \overline{\psi(\boldsymbol{\eta})} \cdot \nabla \phi \left(\frac{y'}{y} \boldsymbol{\eta} \right) \overline{\widehat{g'}(y')} \widehat{g'}(y) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\eta} \right) \frac{1}{|y|^{d}} dy' dy d\boldsymbol{\eta} \quad (33)$$

$$= \frac{\sigma_{a}^{2}}{4\pi^{2}} \int_{\mathbb{R}^{2}} \int_{\mathbb{R}^{d}} \nabla \overline{\psi(\boldsymbol{\eta})} \cdot \left(\nabla \phi \left(\frac{y'}{y} \boldsymbol{\eta} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\eta} \right) \right) d\boldsymbol{\eta} \overline{\widehat{g'}(y')} \widehat{g'}(y) \frac{1}{|y|^{d}} dy' dy.$$

Then, applying the divergence theorem, we obtain

$$\mathbb{E}_{\theta} \left\langle \widehat{K}_{\boldsymbol{w}}, \overline{\phi} \otimes \psi \right\rangle = -\frac{\sigma_a^2}{4\pi^2} \int_{\mathbb{R}^d} \overline{\psi(\boldsymbol{\eta})} \int_{\mathbb{R}^2} \operatorname{div} \left(\nabla \phi \left(\frac{y'}{y} \boldsymbol{\eta} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\eta} \right) \right) \frac{\overline{\widehat{g'}(y')} \widehat{g'}(y)}{|y|^d} dy' dy d\boldsymbol{\eta}.$$
(34)

Similarly, for K_a we calculate

$$\left\langle \widehat{K}_{a}, \overline{\phi} \otimes \psi \right\rangle = \int_{\mathbb{R}^{2d}} \overline{\mathcal{F}_{x' \to \xi'} \left[\nabla_{a} f^{0}(x', \theta) \right]} (\xi') \cdot \mathcal{F}_{x \to \xi} \left[\nabla_{a} f^{0}(x, \theta) \right] (\xi) \overline{\psi(\xi)} \phi(\xi') d\xi' d\xi \quad (35)$$

$$= \int_{\mathbb{R}^{2d}} \overline{\mathcal{F}_{x' \to \xi'} \left[g(w \cdot x') \right]} (\xi') \cdot \mathcal{F}_{x \to \xi} \left[g(w \cdot x) \right] (\xi) \overline{\psi(\xi)} \phi(\xi') d\xi' d\xi$$

$$= \int_{\mathbb{R}^{2d}} \overline{\widehat{g} \left(\frac{w}{\|w\|^{2}} \cdot \xi' \right)} \delta_{w}(\xi') \widehat{g} \left(\frac{w}{\|w\|^{2}} \cdot \xi \right) \delta_{w}(\xi) \phi(\xi) \overline{\psi(\xi')} d\xi' d\xi$$

$$= \int_{\mathbb{R}^{2}} \overline{\psi(y'w)} \phi(yw) \overline{\widehat{g}(y')} \widehat{g}(y) dy' dy.$$

Applying the expectation with respect to θ and using the same change of variable as before, we get

$$\mathbb{E}_{\theta} \left\langle \widehat{K}_{a}, \overline{\phi} \otimes \psi \right\rangle = \int_{\mathbb{R}^{2}} \int_{\mathbb{R}^{d}} \overline{\psi(\boldsymbol{\eta})} \phi\left(\frac{y'}{y} \boldsymbol{\eta}\right) \rho_{\boldsymbol{w}}\left(\frac{1}{y} \boldsymbol{\eta}\right) d\boldsymbol{\eta} \ \overline{\widehat{g}(y')} \widehat{g}(y) \ \frac{1}{|y|^{d}} dy' dy. \tag{36}$$

Combining equations (34) and (36) and taking $\phi = \hat{u}_{\rho_{\text{data}}}$, we obtain

$$\begin{split} \left\langle \frac{d}{dt} \widehat{u}, \psi \right\rangle &= - \, \mathbb{E}_{\theta} \left\langle \widehat{K}_{\theta}, \overline{\widehat{u}_{\rho_{\text{data}}}} \otimes \psi \right\rangle = - \mathbb{E}_{\theta} \left\langle \widehat{K}_{\boldsymbol{w}} + \widehat{K}_{a}, \overline{\widehat{u}_{\rho_{\text{data}}}} \otimes \psi \right\rangle \\ &= \frac{\sigma_{a}^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \overline{\psi(\boldsymbol{\eta})} \int_{\mathbb{R}^{2}} \operatorname{div} \left(\nabla \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\eta} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\eta} \right) \right) \frac{\overline{\widehat{g'}(y')} \widehat{g'}(y)}{|y|^{d}} dy' dy d\boldsymbol{\eta} \\ &- \int_{\mathbb{R}^{d}} \overline{\psi(\boldsymbol{\eta})} \int_{\mathbb{R}^{2}} \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\eta} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\eta} \right) \frac{\overline{\widehat{g}(y')} \widehat{g}(y)}{|y|^{d}} dy' dy d\boldsymbol{\eta}, \end{split}$$

which concludes the proof.

B Derivation of spectral bias for the Fourier Features Model

In the case of a Fourier Features model, recall that our neural network is defined by

$$f(\boldsymbol{x}, \theta(t)) = \frac{1}{\sqrt{2m}} \sum_{k=1}^{m} a_k(t) g_1\left(\boldsymbol{w}_k(t) \cdot \boldsymbol{x}\right) + \frac{1}{\sqrt{2m}} \sum_{k=1}^{m} b_k(t) g_2\left(\boldsymbol{w}_k(t) \cdot \boldsymbol{x}\right), \tag{37}$$

where $\theta(t) = \{a(t), b(t), W(t)\}$, with $a \in \mathbb{R}^m$, $b \in \mathbb{R}^m$ and $W(t) = (w_1(t), \dots, w_m(t))^T \in \mathbb{R}^d \times \mathbb{R}^m$, are the trainable parameters having initial distributions ρ_a , ρ_b and ρ_w , respectively. In this case, the NTK calculation using Lemma A.2 gives

$$K(\boldsymbol{x}, \boldsymbol{x'}) = \mathbb{E}_{\theta} \nabla_{\theta} f^{0}(\boldsymbol{x}, \theta) \cdot \nabla_{\theta} f^{0}(\boldsymbol{x'}, \theta)$$

$$= \mathbb{E}_{\theta} \left[a^{2} g'_{1}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{1}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'} + b^{2} g'_{2}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{2}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'} \right]$$

$$+ abg'_{1}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{2}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'} + abg'_{2}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{1}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'}$$

$$+ g_{1}(\boldsymbol{w} \cdot \boldsymbol{x}) g_{1}(\boldsymbol{w} \cdot \boldsymbol{x'}) + g_{2}(\boldsymbol{w} \cdot \boldsymbol{x}) g_{2}(\boldsymbol{w} \cdot \boldsymbol{x'}) \right]$$

$$= \mathbb{E}_{\theta} \left[a^{2} g'_{1}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{1}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'} \right] + \mathbb{E}_{\theta} \left[b^{2} g'_{2}(\boldsymbol{w} \cdot \boldsymbol{x}) g'_{2}(\boldsymbol{w} \cdot \boldsymbol{x'}) \boldsymbol{x} \cdot \boldsymbol{x'} \right]$$

$$+ \mathbb{E}_{\theta} \left[g_{1}(\boldsymbol{w} \cdot \boldsymbol{x}) g_{1}(\boldsymbol{w} \cdot \boldsymbol{x'}) \right] + \mathbb{E}_{\theta} \left[g_{2}(\boldsymbol{w} \cdot \boldsymbol{x}) g_{2}(\boldsymbol{w} \cdot \boldsymbol{x'}) \right]$$

Corollary B.1 (Cf. Cor. 4.1). Under the assumption 2.1. If the NN is given by equation (11) where $g_1(x) = \cos(2\pi x)$, $g_2(x) = \sin(2\pi x)$, $\mathbb{E}[a^2] = \mathbb{E}[b^2] = \sigma_a^2$ and ρ_w is an even function, then the dynamics (28) in the frequency space is given by

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = \sigma_a^2 \operatorname{div}\left(\rho_{\boldsymbol{w}}(\boldsymbol{\xi})\nabla\widehat{u}_{\rho_{data}}\left(\boldsymbol{\xi},\Theta(t)\right)\right) - \rho_{\boldsymbol{w}}(\boldsymbol{\xi})\widehat{u}_{\rho_{data}}\left(\boldsymbol{\xi},\Theta(t)\right)$$
(39)

Proof. It directly follows that

$$\left\langle \frac{d}{dt} \widehat{u}, \psi \right\rangle = -\mathbb{E}_{\theta} \left\langle \widehat{K}_{a, \boldsymbol{w}} + \widehat{K}_{b, \boldsymbol{w}} + \widehat{K}_{a} + \widehat{K}_{b}, \overline{\widehat{u}_{\rho_{\text{data}}}} \otimes \psi \right\rangle \tag{40}$$

$$= -\frac{\sigma_{a}^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \overline{\nabla \psi(\boldsymbol{\xi})} \cdot \int_{\mathbb{R}^{2}} \nabla \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\overline{g'_{1}(y')} g'_{1}(y)}{|y|^{d}} dy' dy d\boldsymbol{\xi}$$

$$-\frac{\sigma_{b}^{2}}{4\pi^{2}} \int_{\mathbb{R}^{d}} \nabla \overline{\psi(\boldsymbol{\xi})} \cdot \int_{\mathbb{R}^{2}} \nabla \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\overline{g'_{1}(y')} g'_{1}(y)}{|y|^{d}} dy' dy d\boldsymbol{\xi}$$

$$-\int_{\mathbb{R}^{d}} \overline{\psi(\boldsymbol{\xi})} \int_{\mathbb{R}^{2}} \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\overline{g_{1}(y')} g_{1}(y)}{|y|^{d}} dy' dy d\boldsymbol{\xi}$$

$$-\int_{\mathbb{R}^{d}} \overline{\psi(\boldsymbol{\xi})} \int_{\mathbb{R}^{2}} \widehat{u}_{\rho_{\text{data}}} \left(\frac{y'}{y} \boldsymbol{\xi} \right) \rho_{\boldsymbol{w}} \left(\frac{1}{y} \boldsymbol{\xi} \right) \frac{\overline{g_{1}(y')} g_{2}(y)}{|y|^{d}} dy' dy d\boldsymbol{\xi}$$

As

$$\widehat{g}_{1}(z) = \frac{1}{2} \left(\delta(z-1) + \delta(z+1) \right) \quad \text{and} \quad \widehat{g}'_{1}(z) = -\frac{\pi}{i} \left(\delta(z-1) - \delta(z+1) \right)$$

$$\widehat{g}_{2}(z) = \frac{1}{2i} \left(\delta(z-1) - \delta(z+1) \right) \quad \text{and} \quad \widehat{g}'_{2}(z) = \pi \left(\delta(z-1) + \delta(z+1) \right),$$
(41)

then

$$\left\langle \frac{d}{dt}\widehat{u}, \psi \right\rangle$$

$$= -\int_{\mathbb{R}^d} \left[\frac{\sigma_a^2}{4} \nabla \overline{\psi(\boldsymbol{\xi})} \cdot (2\nabla \widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) + 2\nabla \widehat{u}_{\rho_{\text{data}}}(-\boldsymbol{\xi})) + \frac{\sigma_b^2}{4} \nabla \overline{\psi(\boldsymbol{\xi})} \cdot (2\nabla \widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) - 2\nabla \widehat{u}_{\rho_{\text{data}}}(-\boldsymbol{\xi})) \right]$$

$$+ \frac{1}{4} \overline{\psi(\boldsymbol{\xi})} \left(2\widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) - 2\widehat{u}_{\rho_{\text{data}}}(-\boldsymbol{\xi}) \right) + \frac{1}{4} \overline{\psi(\boldsymbol{\xi})} \left(2\widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) + 2\widehat{u}_{\rho_{\text{data}}}(-\boldsymbol{\xi}) \right) \right] \rho_{\boldsymbol{w}}(\boldsymbol{\xi}) d\boldsymbol{\xi},$$

$$(42)$$

therefore

$$\left\langle \frac{d}{dt}\widehat{u}, \psi \right\rangle = -\int_{\mathbb{R}^d} \left(\sigma_a^2 \nabla \overline{\psi(\boldsymbol{\xi})} \cdot \nabla \widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) + \overline{\psi(\boldsymbol{\xi})} \widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi}) \right) \rho_{\boldsymbol{w}}(\boldsymbol{\xi}) d\boldsymbol{\xi}$$
(43)

and

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = \sigma_a^2 \operatorname{div}\left(\rho_{\boldsymbol{w}}(\boldsymbol{\xi})\nabla\widehat{u}_{\rho_{\text{data}}}\left(\boldsymbol{\xi},\Theta(t)\right)\right) - \rho_{\boldsymbol{w}}(\boldsymbol{\xi})\widehat{u}_{\rho_{\text{data}}}\left(\boldsymbol{\xi},\Theta(t)\right)$$

$$\Box$$

Remark B.2. If $g_1(x) = \cos(cx)$ and $g_2(x) = \sin(cx)$, then

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = \left(\frac{2\pi}{c}\right)^{d} \left[\frac{c^{2}\sigma_{a}^{2}}{4\pi^{2}}\operatorname{div}\left(\rho_{\boldsymbol{w}}\left(\frac{2\pi}{c}\boldsymbol{\xi}\right)\nabla\widehat{u}_{\rho_{\text{data}}}\left(\boldsymbol{\xi},\Theta(t)\right)\right) - \rho_{\boldsymbol{w}}\left(\frac{2\pi}{c}\boldsymbol{\xi}\right)\widehat{u}_{\rho_{\text{data}}}\left(\boldsymbol{\xi}\right)\right]. \tag{45}$$

Remark B.3. If ρ_{w} is not even, then the dynamics is given as follows,

$$\frac{d}{dt}\widehat{u}(\boldsymbol{\xi},\Theta(t)) = \sigma_a^2 \operatorname{div}\left[\frac{(\rho_{\boldsymbol{w}}(\boldsymbol{\xi}) + \rho_{\boldsymbol{w}}(-\boldsymbol{\xi}))}{2}\nabla\widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi},\Theta(t))\right] - \frac{(\rho_{\boldsymbol{w}}(\boldsymbol{\xi}) + \rho_{\boldsymbol{w}}(-\boldsymbol{\xi}))}{2}\widehat{u}_{\rho_{\text{data}}}(\boldsymbol{\xi},\Theta(t)).$$
(46)

C Numerical solution of the damped heat equation (13).

In order to validate our findings we experimentally compared the behavior of the exact learning dynamics of a FF neural network, with the theoretical prediction given by equation (13). To solve equation (13) we used a standard approach with FEM and Backward Euler scheme, as mentioned below.

C.1 Finite element approximation

Let $H^1(\Omega)$ be the standard Sobolev space, for $\Omega \subset \mathbb{R}$, with norm

$$\|\phi\|_{H^1} := \left(\int_{\Omega} |\phi(x)|^2 + |\partial_x \phi(x)|^2 dx\right)^{1/2},\tag{47}$$

with derivatives considered in the weak sense (distributional).

If we let $\phi \in H^1(\Omega)$ be a test function, multiplying equation by ϕ and integrating over Ω , using integration by parts and the fact that $\partial_{\xi} u(\xi, t) = 0$ on the boundary, we obtain the variational form of (13):

$$\int_{\Omega} \partial_t \hat{u}(\xi, t) \overline{\phi(\xi)} \, d\xi = -\sigma_a^2 \int_{\Omega} \rho_{\boldsymbol{w}}(\xi) \partial_{\xi} \hat{u}(\xi, t) \partial_{\xi} \overline{\phi(\xi)} \, d\xi - \int_{\Omega} \rho_{\boldsymbol{w}}(\xi) \hat{u}(\xi, t) \overline{\phi(\xi)} \, d\xi. \quad t \ge 0, (48)$$

with an initial condition prescribing the value of $\hat{u}(\xi,0)$. Given T>0, the problem is to find $u\in C^1([0,T];H^1(\Omega))$, such that (48) holds for all $\phi\in H^1(\Omega)$.

Let τ_h be a partition of Ω with diameter h, and define the corresponding standard finite element space $U_h \in H^1$ of dimension N. The numerical scheme goal is to find an approximated solution $u(\xi,t)$ of equation (48) of the form

$$u(\xi,t) \approx \sum_{1 \le j \le N} c_j(t)\varphi_j(\xi),$$
 (49)

where $\{\varphi_i(x)\}_{i=1}^N$ is a real-valued basis of U_h and $c_i(t) \in \mathbb{C}$ for all $t \in [0, T]$. Substituting each $\overline{\phi} = \overline{\varphi}_j = \varphi_j$ in equation (48) we obtain a system of equations given by

$$\sum_{j=1}^{N} c_j'(t) \int_{\Omega} \varphi_j \varphi_i \, d\xi = -\sigma_a^2 \sum_{j=1}^{N} c_j(t) \int_{\Omega} \rho_{\boldsymbol{w}}(\xi) \nabla \varphi_j \nabla \varphi_i \, d\xi - \sum_{j=1}^{N} c_j(t) \int_{\Omega} \rho_{\boldsymbol{w}}(\xi) \varphi_j \varphi_i \, d\xi, \quad (50)$$

for $t \in [0,T]$ and with i = 1,...,N, and where $c_i(t)$ are unknown functions to be determined.

In summary, if we let $c(t) = (c_1(t), c_2(t), \dots, c_N(t))$ be the vector of unknown time coefficients, the system of equations for c can be rewritten as:

$$Lc'(t) = -\sigma_a^2 Mc(t) - Nc(t), \tag{51}$$

where

$$(L)_{i,j} = \int_{\Omega} \varphi_j \varphi_i \, \mathrm{d}\xi, \qquad (M)_{i,j} = \int_{\Omega} \rho_{\boldsymbol{w}} \nabla \varphi_j \nabla \varphi_i \, \mathrm{d}\xi, \quad \text{ and } \quad (N)_{i,j} = \int_{\Omega} \rho_{\boldsymbol{w}} \varphi_j \varphi_i \, \mathrm{d}\xi.$$

Finally, we discretize the time variable using a Backward Euler scheme: let $t_0 = 0 < t_1 < ... < t_{n_t} = T$ a partition of the time interval with into segments of length h_t and let $c_k \approx c(t_k), k = 1, ..., n_t$, then for the fully discrete equation, with our choice of h_t and with h as above, is

$$(L + h_t \sigma_a^2 M + h_t N) c_{k+1} = L c_k \tag{52}$$

for $k = 0, ..., n_t$, where c_0 is obtained similarly to (50) from the initial condition $\hat{u}_0 = \hat{u}(\xi, 0)$.

For our experiment were chose T=500 time step $h_{\rm t}=0.1$ and $\Omega=(-60,60)$ with spatial step h=0.5. For each index i, the basis function c_i for our FEM implementation is piecewise linear, and takes the value 1 at node i and 0 at the other nodes.

To compare to NN training, we use different time snapshots in Figure 4. The NN snapshot is taken every 4000 iterations, whereas the FEM snapshot is taken every 500-time steps.

D Other numerical experiments

D.1 Robustness of assumption: NTK

In this subsection we empirically test the robustness of the NTK assumption, more precisely we test how large must be m to ensure that our results remain qualitatively valid.

In Figure 6 we present a qualitative analysis of our model for deeper architectures and smaller hidden layer widths m. We find that the frequency learning rate κ still depends on the initial distribution despite the fact that m has a low value. Furthermore, the hyperparameter m plays an important role in the learning speed. Indeed, note the change in order of magnitudes of frequency learning rates for different values of m, compared with figure 3, where the learning speed reaches orders of 8 while for small values of m is close to 2. The same holds for the multilayer case (see the 4-layer of the figure 5 vs the 4-layer of the figure 6).

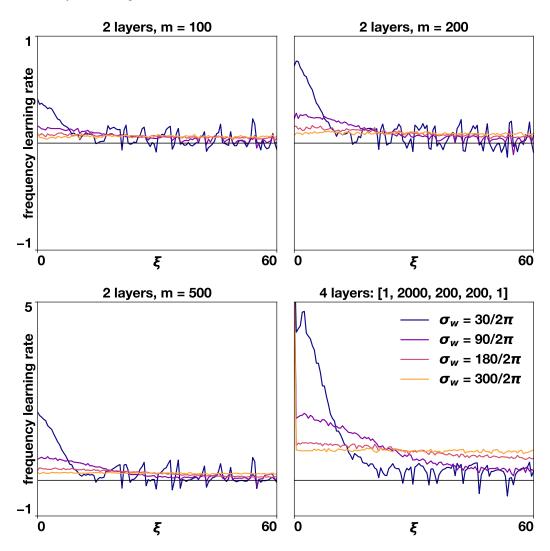


Figure 6: Robustness analysis for the NTK assumption with different values of m and number of layers.