Is the neural tangent kernel of PINNs deep learning general partial differential equations always convergent ?

Zijian Zhou and Zhenya Yan*

Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. In this paper, we study the neural tangent kernel (NTK) for general partial differential equations (PDEs) based on physics-informed neural networks (PINNs). As we all know, the training of an artificial neural network can be converted to the evolution of NTK. We analyze the initialization of NTK and the convergence conditions of NTK during training for general PDEs. The theoretical results show that the homogeneity of differential operators plays a crucial role for the convergence of NTK. Moreover, based on the PINNs, we validate the convergence conditions of NTK using the initial value problems of the sine-Gordon equation and the initial-boundary value problem of the KdV equation.

Keywords: Deep learning; physics-informed neural networks; Neural tangent kernel; partial differential equations; convergence condition

1 Introduction

In the past decade, artificial intelligence (AI) has witnessed widespread applications across various domains, encompassing computer vision, natural language processing, equation solving, and diverse business sectors [1, 2]. Some researchers began exploring the utilization of neural networks to study PDEs as early as the 1990s [3, 4] based on the approximation theory [5,6]. Recently, with the advent of remarkable advances in computational power, some scholars have renewed their focus on leveraging neural networks for PDE applications. A series of deep learning approaches have been successively proposed and achieved significant breakthroughs in the aspect of learning PDEs. Among these, the most prevalent approach involves approximating the solutions of a PDE using neural networks, which encompasses methods, such as deep galerkin method (DGM) [7], physics-informed neural networks (PINNs) [8], and deep Ritz method [9]. Another approach is centered around approximating the solution map of a PDE using neural networks. PDE-Net [10, 11], along with other studies [12, 13], combine traditional numerical methods with neural networks. Similarly, DeepONet [14] and Fourier neural operator (FNO) [15], among others [16, 17], employ neural networks to approximate the solution map of a PDE in a black box manner.

The aforementioned neural network methods find extensive applications in diverse fields. For instance, the transitions between two metastable states were studied in a high-dimensional probability distribution [18]. A fermionic neural network (FermiNet) was proposed to compute solutions to the many-electron Schrödinger equation [19]. DeepONet was employed to predict crack paths in quasi-brittle materials [20]. A two-stage training method is used to deal with training the loss function that contains both equations and conservation laws [21]. An improved PINNs method based on Miura transformation is proposed, which realizes unsupervised learning solutions of nonlinear PDEs [22]. The third-order nonlinear wave equations were studied [23, 24]. Moreover, variable coefficient PDEs are considered [25]. Dynamics of the one-dimensional quantum droplets are studied in [26]. Nonlinear dispersive equations were studied to explore peakon and periodic peakon solutions [27]. Additionally, Refs. [28–30] delved into the study of bright solitons, breathers, and rogue wave solutions of the nonlinear Schrödinger-type equations. These examples demonstrate the successful application of neural network methods across various fields.

The universality of neural networks is one of their most significant properties. The pioneering theoretical result regarding the approximation capabilities of neural networks was introduced by Cybenko in 1989 [5]. In the context of solving PDEs, Physics-informed neural networks (PINNs) utilize neural networks as approximators for PDE solutions, with the first theoretical analysis presented in [33]. Additionally, [34] investigates over-parameterized two-layer networks and presents convergence analysis for gradient descent in the context of second-order linear PDEs.

Simultaneously, various theoretical works of neural networks were also analyzed. The NTK theory [35] demonstrated that the training dynamics of supervised learning models can be interpreted as kernel regression. As the width of the neural network tends to infinity, the kernel converges to a deterministic kernel. This provides a novel analytical tool for theoretical analyses of neural networks. In scenarios where the kernel approaches a constant kernel,

*Corresponding author. *Email address*: zyyan@mmrc.iss.ac.cn

certain pathologies can be analyzed, as explored in [36]. Additionally, Ref. [37] considered the finite-width corrections for the limit NTK. However, NTK computation can be challenging for large-scale data. To address this, Ref. [38] focused on optimizing the solver for kernel methods in the context of significant scale problems. Furthermore, [39] extended the NTK theory to convolutional neural networks.

Recently, there has been a considerable body of research that has applied the NTK theory to diverse domains. Notably, Ref. [32] empirically demonstrated the consistent superiority of kernel regression employing a 14-layer CNTK over ResNet-34 trained with standard hyperparameters. This performance advantage is observed on a randomly selected subset of CIFAR-10, containing a maximum of 640 samples. Furthermore, Ref. [40] extended the implications of NTK theory, as presented in [35], to the realm of partial PINNs models. Moreover, Ref. [41] employed the NTK theory to establish that standard neural networks, both theoretically and practically, struggle to capture high-frequency information. Parallelly, Ref. [42] elucidated the characteristic of low-frequency initial learning in PINNs by leveraging the NTK framework. In this paper, we will provide a succinct overview of the primary steps involved in employing NTK to explicate the behavior of PINNs.

A fundamental difference between the NTK analysis of PINNs and the standard NTK approach lies in the incorporation of a differential operator and its associated kernel into the loss function. To be specific, Ref. [35] considered the following boundary condition problem in a bounded domain $\Omega \in \mathbb{R}^d$ (for the time-dependent problems, t can be thought of as a part of $\boldsymbol{x}$):

$$\begin{cases} \mathcal{F}[q](\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ q(\boldsymbol{x}) = g(\boldsymbol{x}), & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{1}$$

where $\mathcal{F}$ denote a differential operator (Poisson equation and wave equation were considered in [35]), $q(\boldsymbol{x})$ is the solution of $\mathcal{F}[q](\boldsymbol{x}) = f(\boldsymbol{x})$ with $\boldsymbol{x} = (x_1, x_2, ..., x_d)$. For time-dependent problems, t can be regarded as an additional variable of $\boldsymbol{x}$.

The basic idea of the PINNs method [8] is to use a neural network to approximate the solution of the PDE. Usually, the standard choice for the fully-connected neural network (FCNN) with L layers ($L-1$ hidden layers) is defined recursively as

$$\begin{aligned} \boldsymbol{q}^{(0)}(\boldsymbol{x}) =& \frac{1}{\sqrt{N_0}} \boldsymbol{W}^{(0)} \cdot \boldsymbol{x} + \boldsymbol{b}^{(0)}, \\ \boldsymbol{g}^{(i)}(\boldsymbol{x}) =& \sigma(\boldsymbol{q}^{(i-1)}(\boldsymbol{x})) \in \mathbb{R}^{N_i}, \\ \boldsymbol{q}^{(i)}(\boldsymbol{x}) =& \frac{1}{\sqrt{N_i}} \boldsymbol{W}^{(i)} \cdot \boldsymbol{g}^{(i)}(\boldsymbol{x}) + \boldsymbol{b}^{(i)}, \quad i = 1, \cdots, L, \end{aligned} \tag{2}$$

where $\boldsymbol{W}^{(i)} (\in \mathbb{R}^{N_{i+1} \times N_i})$ and $\boldsymbol{b}^{(i)} (\in \mathbb{R}^{N_{i+1}})$ are the weight matrices to be trained, N_i is the width of the i-th layer of neural network (i.e., number of neurons in the i-th layer), and σ is a coordinate-wise activation function (σ is usually chosen as ReLu, Sigmoid or Tanh). In the usual initialization of NTK, all the weights and biases are initialized to be independent and identically distributed (i.i.d.) as standard normal distribution $\mathcal{N}(0,1)$. The final output of neural network, $\boldsymbol{q}^{(L)}(\boldsymbol{x})$, can be defined as the neural network solution $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = \{\boldsymbol{W}^{(i)}, \boldsymbol{b}^{(i)}\}_{i=0}^{L}$. The appropriate weight matrices of the neural network can be obtained by optimizing the PINNs loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{j=1}^{N_b} \left| q(\boldsymbol{x}_b^j, \boldsymbol{\theta}) - g(\boldsymbol{x}_b^j) \right|^2 + \frac{1}{2} \sum_{j=1}^{N_f} \left| \mathcal{F}[q](\boldsymbol{x}_f^j, \boldsymbol{\theta}) - f(\boldsymbol{x}_f^j) \right|^2 \tag{3}$$

where $\{\boldsymbol{x}_f^j\}_{j=1}^{N_f}$, $\{\boldsymbol{x}_b^j, g(\boldsymbol{x}_b^j)\}_{j=1}^{N_b}$ indicate the physical information training points and boundary data set, respectively. For the above optimization problem, if the gradient descent method (GD) with a minimal learning rate is selected, the optimization process can be converted to a gradient flow model:

$$\frac{d\boldsymbol{\theta}}{dt} = -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \tag{4}$$

After that, the gradient flow model can be rewritten as following kernel gradient descent

$$\begin{bmatrix} \mathcal{F}_t[q](\boldsymbol{x}_f, \boldsymbol{\theta}) \\ q_t(\boldsymbol{x}_b, \boldsymbol{\theta}) \end{bmatrix} = - \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fb} \\ \mathbf{K}_{bf} & \mathbf{K}_{bb} \end{bmatrix} \begin{bmatrix} \mathcal{F}[q](\boldsymbol{x}_f, \boldsymbol{\theta}) - f(\boldsymbol{x}_f) \\ q(\boldsymbol{x}_b, \boldsymbol{\theta}) - g(\boldsymbol{x}_b) \end{bmatrix}, \tag{5}$$

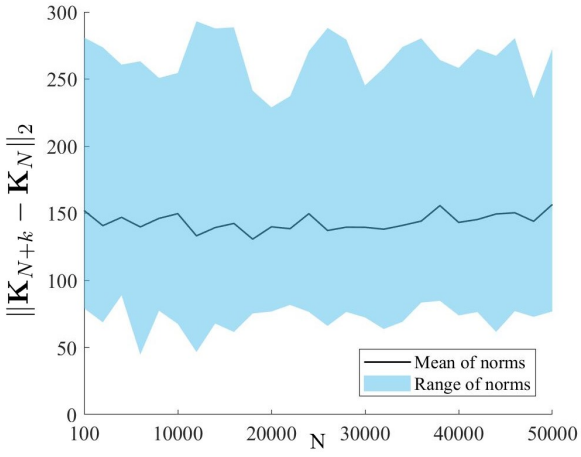

Figure 1: The trends of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$ of the initial value problem of the sine-Gordon equation . The blue area is the variation range of the results of 50 independent experiments, and the dark line is the mean of these experiments.

where

$$\begin{aligned}
\mathbf{K}_{ff} &= \Big(\Big(\nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^j, \boldsymbol{\theta}) \Big)_{i,j=1,\ldots,N_f}, \\
\mathbf{K}_{fb} = \mathbf{K}_{bf}^T &= \Big(\Big(\nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} q(x_b^j, \boldsymbol{\theta}) \Big)_{i=1,\ldots,N_f; j=1,\ldots,N_b}, \\
\mathbf{K}_{bb} &= \Big(\Big(\nabla_{\boldsymbol{\theta}} q(x_b^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} q(x_b^j, \boldsymbol{\theta}) \Big)_{i,j=1,\ldots,N_b},
\end{aligned} \tag{6}$$

$\mathbf{K} \triangleq \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fb} \\ \mathbf{K}_{bf} & \mathbf{K}_{bb} \end{bmatrix}$ can be called NTK. The main theoretical results of NTK for PINNs including:

- Under the certain initialization of parameters, the initialization matrix of $\mathbf{K}(0)$ converges to a deterministic kernel $\mathbf{K}^*$ in probability [35] when $N \to \infty$ (N is the width of the hidden layer):

$$\mathbf{K}(0) \xrightarrow{\mathcal{P}} \mathbf{K}^*. \tag{7}$$

- $\mathbf{K}$ stays asymptotically constant ($\mathbf{K}(0)$) during training (when $N \to \infty$):

$$\lim_{N \to \infty} \sup_{t \in [0,T]} \|\boldsymbol{K}(t) - \boldsymbol{K}(0)\|_2 = 0. \tag{8}$$

By consolidating the aforementioned two outcomes, we obtain the following:

$$\mathbf{K}(t) \approx \mathbf{K}(0) \approx \mathbf{K}^*, \quad \forall t > 0. \tag{9}$$

However, it is important to note that the original theoretical findings solely focus on the Poisson equation and do not account for general PDEs. In practice, we have observed certain discrepancies in the NTK examples with respect to these results. Next, let's introduce an example:

We consider the initial value problem (IVP) of the sine-Gordon equation [43]:

$$\begin{cases} q_{tt}(x,t) - q_{xx}(x,t) = \sin(q(x,t)), & (x,t) \in [-5,5] \times [0,5], \\ q(x,0) = q_0(x), & x \in [-5,5], \end{cases} \tag{10}$$

where $q_0(x)$ is the initial condition. An exact shock wave solution of the sine-Gordon equation would be set as the initial condition:

$$q(x,0) = q_0(x) = 4\arctan(e^{\sqrt{2}x}). \tag{11}$$

We randomly select 50 initial sampling points and 100 spatial sampling points, denoted as $N_b = 50$ and $N_f = 100$, respectively. Consequently, the NTK matrix corresponds to $\mathbb{R}^{150\times150}$. To assess the convergence behavior of the initialized NTK (7), we investigate a series of two-layer fully connected neural networks (FCNNs) (2) with a smooth activation function, specifically the hyperbolic tangent function, denoted as $\tanh(\cdot)$. The widths of these FCNNs are chosen as $\{100, 2000, 4000, 6000, ..., 50000\}$, respectively. For each width of the neural network, the weights are initialized independently and identically distributed (i.i.d.) according to a standard normal distribution, $\mathcal{N}(0,1)$, while the biases are initialized as zero. Subsequently, the initialized NTK is calculated using (6).

Our objective is to examine whether $\mathbf{K}_N(0)$ converges to $\mathbf{K}^*$ in probability as N tends to infinity. However, computing the deterministic limiting kernel $\mathbf{K}^*$ for a specific example poses challenges. Therefore, we investigate the behavior of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$ instead of $\|\mathbf{K}_N - \mathbf{K}^*\|_2$, where N denotes the width of the neural network and k represents the interval in the sequence (in this case, $k = 2000$). Although the convergence of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$ and $\|\mathbf{K}_N - \mathbf{K}^*\|_2$ may not be strictly equivalent, they still provide insights into the divergence of $\|\mathbf{K}_N - \mathbf{K}^*\|_2$.

Figure 1 displays the trend of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$. The horizontal axis N corresponds to the number of neurons in the hidden layer. In order to demonstrate convergence or divergence in probability, the results are aggregated from 50 independent experiments. The blue shaded region in Fig. 1 represents the outcomes of these experiments, while the black line represents the mean value across these experiments. It is evident that the sequence of initialized NTKs exhibits no discernible trend of convergence.

To gain insights into the lack of convergence of initialized NTK towards a deterministic kernel, a further discussion on NTK is warranted. The rest of this paper are organized as follows: In Section 2, we provide a detailed analysis of NTK theory for the PDEs. In Section 3, we present additional experimental results that shed light on the impact of the coefficient s on the convergence of NTK.

2 The neural tangent kernel for general PDEs

In this section, we would like to consider the NTK for the boundary problems of general PDEs (GPDEs):

$$\begin{cases} \mathcal{F}[q](x,\boldsymbol{\theta}) = f(x), & x \in \Omega, \\ q_{mx}(x,\boldsymbol{\theta}) = g(x), & x \in \partial\Omega, \end{cases} \tag{12}$$

where $\mathcal{F}[\cdot]$ is a continuous nonlinear operator, $q_{mx}(x,\boldsymbol{\theta})$ represent some m-order derivative of $q(x,\boldsymbol{\theta})$ with respect with x. In order to consider the general case, scaling parameter $1/\sqrt{N}$ will be changed to $1/N^s$. The subsequent analysis reveals that the convergence of NTK is influenced by the convergence coefficient, denoted as s. It is worth noting that the two-layer FCNN is initialized in the following manner:

$$\boldsymbol{q}(x,\boldsymbol{\theta}) = \frac{1}{N^s}\boldsymbol{W}^{(1)} \cdot \sigma(\boldsymbol{W}^{(0)} \cdot x + \boldsymbol{b}^{(0)}) + \boldsymbol{b}^{(1)}, \tag{13}$$

where $\boldsymbol{W}^{(0)} \in \mathbb{R}^{N_1\times1}, \boldsymbol{b}^{(0)} \in \mathbb{R}^{N_1\times1}, \boldsymbol{W}^{(1)} \in \mathbb{R}^{1\times N_1}$ and $\boldsymbol{b}^{(1)} \in \mathbb{R}$, It is worth mentioning that the coefficient $1/N^s$ only adds to the output layer. Because the nonlinear activation function σ is non-homogeneous, it will affect the convergence of initialized NTK (It will be discussed in detail in the proof of Theorem 2.2). The loss function is chosen as

$$\mathcal{L}(\boldsymbol{\theta}) = \alpha\mathcal{L}_f(\boldsymbol{\theta}) + \beta\mathcal{L}_b(\boldsymbol{\theta}) = \frac{\alpha}{2}\sum_{i=1}^{N_f}\left(\mathcal{F}[q](x_f^i,\boldsymbol{\theta}) - f(x_f^i)\right)^2 + \frac{\beta}{2}\sum_{i=1}^{N_b}\left(q_{mx}(x_b^i,\boldsymbol{\theta}) - g(x_b^i)\right)^2. \tag{14}$$

where α, β are added to balance the loss of different terms.

In the same way in Sec. 1, the optimization process can be changed to the kernel gradient descent.

Lemma 2.1. Given the data points $\{x_f^i, f(x_f^i)\}_{i=1}^{N_f}$, $\{x_b^i, g(x_b^i)\}_{i=1}^{N_b}$ and the gradient flow (4). $\mathcal{F}[q](x,\boldsymbol{\theta}) \in \mathbb{R}^{N_f\times1}$, obey the following matrix evolution equation

$$\begin{bmatrix} \mathcal{F}_t[q](x_f,\boldsymbol{\theta}) \\ q_{mx,t}(x_b,\boldsymbol{\theta}) \end{bmatrix} = -\begin{bmatrix} \alpha\mathbf{K}_{ff} & \beta\mathbf{K}_{fb} \\ \alpha\mathbf{K}_{bf} & \beta\mathbf{K}_{bb} \end{bmatrix}\begin{bmatrix} \mathcal{F}[q](x_f,\boldsymbol{\theta}) - f(x_f) \\ q_{mx}(x_b,\boldsymbol{\theta}) - g(x_b) \end{bmatrix} \tag{15}$$

where

$$\begin{aligned}
\mathbf{K}_{ff} &= \Big(\Big(\nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^j, \boldsymbol{\theta}) \Big)_{i,j=1,\dots,N_f}, \\
\mathbf{K}_{fb} = \mathbf{K}_{bf}^T &= \Big(\Big(\nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x_f^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} q_{mx}(x_b^j, \boldsymbol{\theta}) \Big)_{i=1,\dots,N_f;\, j=1,\dots,N_b}, \\
\mathbf{K}_{bb} &= \Big(\Big(\nabla_{\boldsymbol{\theta}} q_{mx}(x_b^i, \boldsymbol{\theta}) \Big)^T \nabla_{\boldsymbol{\theta}} q_{mx}(x_b^j, \boldsymbol{\theta}) \Big)_{i,j=1,\dots,N_b}.
\end{aligned} \tag{16}$$

Proof. The proof of lemma 2.1 is given in Appendix A. □

When employing gradient descent (GD) with a learning rate that tends to 0, Lemma 2.1 establishes that the training process can be reformulated as a kernel gradient descent problem. Consequently, the investigation of training processes can be conducted by analyzing the behavior of the kernel matrix.

The first theoretical finding affirms that the initialized NTK converges, in probability, to a deterministic kernel matrix as the width of the neural network, denoted as N, approaches infinity.

Theorem 2.2. For the kernel (15) of the boundary problem of PDE, where $\mathcal{F}$ is a continuous differential operator, when the width of the neural network, denoted as N, tends to infinity, the kernel associated with the neural network (13) converges in probability to the following deterministic limiting kernel:

$$\mathbf{K}(0) = \begin{bmatrix} \alpha \mathbf{K}_{ff}(0) & \beta \mathbf{K}_{fb}(0) \\ \alpha \mathbf{K}_{bf}(0) & \beta \mathbf{K}_{bb}(0) \end{bmatrix} \xrightarrow{\mathcal{P}} \mathbf{K}^*, \tag{17}$$

where $\mathbf{K}_{ff}(0), \mathbf{K}_{fb}(0), \mathbf{K}_{bf}(0), \mathbf{K}_{bb}(0)$ are defined in Lemma 2.1, and $\xrightarrow{\mathcal{P}}$ represent convergence by probability.

Proof. The proof of Theorem 2.2 is given in Appendix B.

Remark 1. The crucial aspect determining kernel convergence is the balance between the number of neurons and the convergence coefficient, represented by N^{-s}. The convergence of $\mathbf{K}_{bb}$ occurs in probability when $s \geq \frac{1}{2}$. In the case where $F_0[q, q_x, q_{xx}, ..., q_{nx}] \cdot F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \neq 0$ ($F_0[q, q_x, q_{xx}, ..., q_{nx}] \neq 0$), or if there exists a non-homogeneous monomial in the polynomial $\{F_i[q, q_x, q_{xx}, ...q_{nx}]\}_{i=1}^n$ (referred to as case A), the convergence of $\mathbf{K}_{ff}$ ($\mathbf{K}_{bf}$) occurs in probability when $s \geq 1$. On the other hand, if $F_0[q, q_x, q_{xx}, ..., q_{nx}] \cdot F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] = 0$ ($F_0[q, q_x, q_{xx}, ..., q_{nx}] = 0$), or if every monomial in the polynomials $\{F_i[q, q_x, q_{xx}, ..., q_{nx}]\}_{i=1}^n$ is homogeneous (referred to as case B), the convergence of $\mathbf{K}_{ff}$ ($\mathbf{K}_{bf}$) occurs in probability when $s \geq s_1$ (as defined in Eq. (39)) ($s \geq s_2$ as defined in Eq. (47)).

In other words, in case A, $\mathbf{K}$ convergence by probability when $s \geq 1$. In case B, $\mathbf{K}$ convergence by probability when $s \geq s_1 (> s_2)$. This means the past initialization coefficient $N^{-1/2}$ does not universally guarantee convergence. It depends on the homogeneity of $\{F_i[q, q_x, q_{xx}, ...q_{nx}](x, \boldsymbol{\theta})\}_{i=1}^n$ and the value of $F_0[q, q_x, q_{xx}, ...q_{nx}](x, \boldsymbol{\theta})$. In fact, in the examples in the first and third sections, we observe instances where the initialized NTK diverges.

Remark 2. The significance of the non-homogeneous term F_i in the analysis of Theorem 2 is evident. In a multi-layer network, the nonlinear activation function also exhibits non-homogeneity. When the convergence coefficient N^{-s} is introduced to the hidden layer, the balance between the number of neurons and the convergence coefficient becomes challenging.

Motivated by Theorem 2.2, a comprehensive examination of the NTK range during training is undertaken. Remarkably, the NTK remains a constant matrix (referred to as the initialized NTK) throughout the training process when $N \to \infty$ and $s > 1/4$.

Theorem 2.3 For the loss function (3), if the following assumptions are satisfied for any $T > 0$:

(i) For $t \in T$, all parameters of the network are uniformly bounded, i.e., there exists a constant $C > 0$ (independent on N) such that

$$\sup_{t \in [0,T]} \|\boldsymbol{\theta}(t)\|_\infty \leq C, \tag{18}$$

(ii) The derivatives of the equation are uniformed bounded, i.e. there exists a constant $C > 0$ (independent on n) such that

$$\sup_{i \in \{0,1,\dots,n\}} \|F_i[q, q_x, q_{xx}, ..., q_{nx}]\|_\infty \leq C, \tag{19}$$

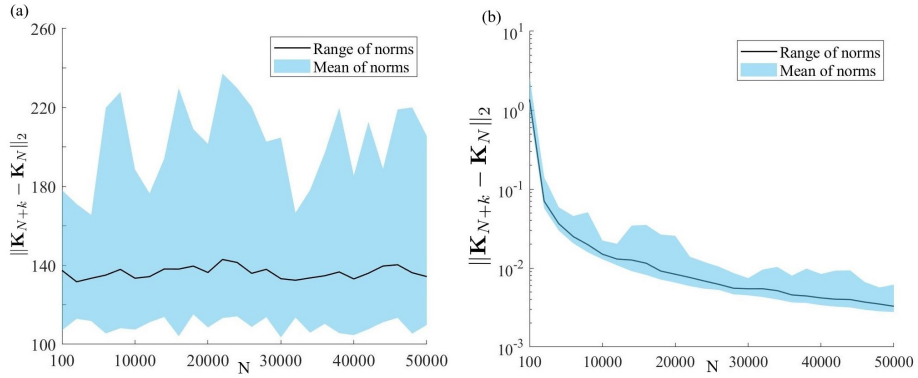

Figure 2: The trends of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$ of the initial value problem of the sine-Gordon equation (10) when (a) $s = 0.5$, (b) $s = 1$. The blue area is the variation range of the results of 50 independent experiments, and the black line denotes the mean value of these experiments.

(iii) There exists a constant $C > 0$,

$$\begin{aligned} &\int_0^T \left| \sum_{i=1}^{N_b} \left(\frac{\partial^m q}{\partial x^m} \left(x_b^i, \boldsymbol{\theta}(\tau) \right) - f\left(x_b^i \right) \right) \right| d\tau \leq C, \\ &\int_0^T \left| \sum_{i=1}^{N_f} \left(\mathcal{F}[q](x_f^j, \boldsymbol{\theta}(\tau)) - g(x_b^i) \right) \right| d\tau \leq C, \end{aligned} \tag{20}$$

(iv) The activation function σ is smooth and its k-order derivatives are bounded (i.e. $|\sigma^{(i)}| \leq C$, $0 \leq i \leq k$), then when $s > 1/4$ we have

$$\lim_{N \to \infty} \sup_{t \in [0,T]} \|\boldsymbol{K}(t) - \boldsymbol{K}(0)\|_2 = 0. \tag{21}$$

Proof. The proof of Theorem 2.3 is given in Appendix C.

Remark 3. Condition (ii) imposes the requirement that the derivatives of the equation are uniformly bounded. This condition shows that if $F_i[q, q_x, q_{xx}, ..., q_{nx}]$ is unbounded, the convergence of $\boldsymbol{K}(t)$ may not be guaranteed. One example of such an unbounded term is the logarithmic function.

Remark 4. Theoretical analysis indicates that the Neural Tangent Kernel (NTK) remains a constant matrix when $s > 1/4$. This finding extends the range of initialized neural networks, which was previously established for the case of $s = 1/2$. The convergence of the NTK during training for values of $s < 1/2$ is verified in next section. In next section, we present two examples where the NTK remains a constant matrix throughout the training process for $s = 1/4$.

In the preceding analysis, we provide a detailed discussion of the original conclusion, introducing relaxations or additional constraints to the theorem. These derived conclusions serve to expand or restrict the applicability of the NTK theory. The subsequent numerical examples serve to substantiate the validity of such an analysis.

3 Some examples

In this section, we present two illustrative examples to elucidate the influence of the coefficient s on the convergence of the initialized NTK and the NTK during training. Specifically, we examine the initial value problem of the non-homogeneous nonlinear sine-Gordon equation represented in Sec. 3.1, and the initial boundary value problem of the homogeneous nonlinear KdV equation depicted in Sec. 3.2.

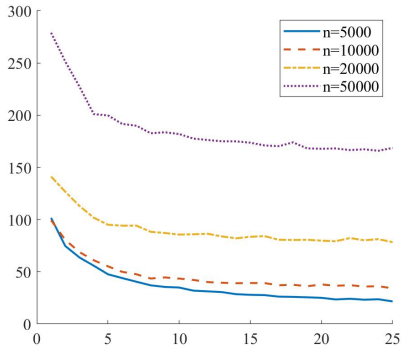

Figure 3: The trends of $\sup_{t\in[0,n]} \|\mathbf{K}_N(t) - \mathbf{K}_N(0)\|_2$ of the initial value problem of the sine-Gordon equation (10) when $s = 1/4$, where n represents the maximal training step and N the width of the neural networks.

3.1 NTK convergence of sine-Gordon equation

In the first experiment, we continue to examine the initial value problem given by Eq. (10). As discussed in Remark 2 of Theorem 2.2, the convergence of the NTK is influenced by the nonlinear activation function when the convergence coefficient s is not sufficiently large. Consequently, in the subsequent experiments, we only apply the convergence coefficient to the output layer (13). We employ the same series of neural networks as in Section 2 to validate the theoretical findings. Given that the initialized NTK converges probabilistically to a deterministic kernel, we independently run the program 50 times and subsequently plot the average and range of the results in Fig.2.

Figure 2 presents the experimental results for two different values of s: $s = 0.5$ (Fig. 2a) and $s = 1$ (Fig. 2b). It is evident that the initialized NTK diverges when $s = 0.5$, but converges when $s = 1$. This observation highlights the influence of the parameter s on the convergence and divergence of the initialized NTK. Specifically, it demonstrates that only when s is sufficiently large, the initialized NTK for the sine-Gordon equation is guaranteed to converge to a deterministic kernel. As analyzed in Theorem 2.2, the presence of non-homogeneous terms such as $\sin(\cdot)$ leads to the divergence of the initialized NTK.

We further investigate the convergence of NTK during training with a convergence coefficient of $s = 1/4$. A series of two-layer FCNNs with varying widths, ranging from $200, 400, \ldots, 5000$, are trained using the standard gradient descent method with the fixed weighted Loss function (14). For each width of the FCNN, we train the models for different numbers of steps, specifically 5000, 10000, 20000, and 50000 steps. The results are plotted as separate curves in Figure 3. To calculate the loss function, we randomly select 50 initial sampling points and 100 spatial sampling points. Each point on the graph in Fig. 3 represents the upper bound of the 2-norm difference between the NTK and the initial NTK over the entire training process.

Figure 3 presents the training results under these conditions. As the width N increases, the quantity $\sup_{t\in[0,n]} \|\mathbf{K}_N(t) - \mathbf{K}_N(0)\|_2$ consistently decreases, regardless of the maximum number of training steps. This observation indicates that the NTK remains a constant matrix throughout the training process when $s = 1/4$. Consequently, this finding suggests that NTK theory can be applied to a broader range of FCNN initialization scenarios.

3.2 NTK convergence of KdV equation

In the second experiment, we investigate the initial boundary value problem of the homogeneous nonlinear KdV equation [43]:

$$\begin{cases} q_t(x,t) + 6q(x,t)q_x(x,t) + q_{xxx}(x,t) = 0, \quad (x,t) \in (-5,5) \times (0,5), \\ q(-5,t) = q(5,t), \quad t \in [0,5], \\ q(x,0) = q_0(x), \quad x \in [-5,5]. \end{cases} \tag{22}$$

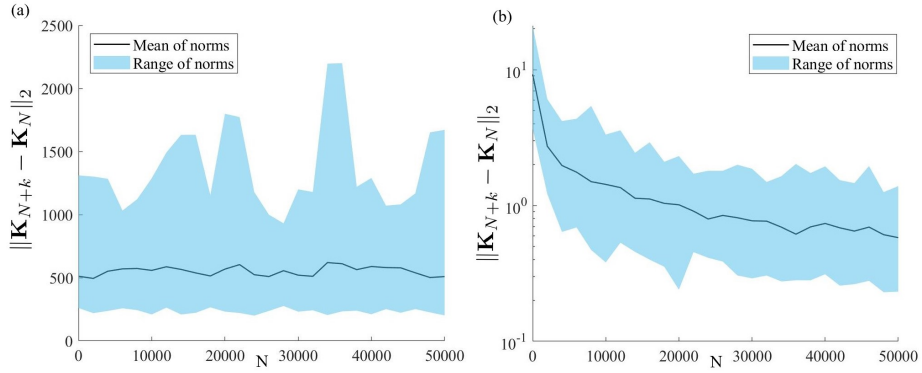

Figure 4: The trends of $\|\mathbf{K}_{N+k} - \mathbf{K}_N\|_2$ of the initial value problem of the KdV equation (22) when (a) $s = 0.5$, (b) $s = 1$. The blue area is the variation range of the results of 50 independent experiments, and the black line is the mean value of these experiments.

Periodic boundary conditions are imposed by incorporating a boundary loss term into the overall Loss function (14):

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) =& \frac{1}{2}\sum_{j=1}^{N_i}\left|q(x_i^j, 0, \boldsymbol{\theta}) - q_0(\boldsymbol{x}_i^j)\right|^2 + \frac{1}{2}\sum_{j=1}^{N_b}\left|q(-5, t_b^j, \boldsymbol{\theta}) - q(5, t_b^j, \boldsymbol{\theta})\right|^2 \\ &+ \frac{1}{2}\sum_{j=1}^{N_f}\left|\mathcal{F}[q](x_f^j, t_f^j, \boldsymbol{\theta}) - f(x_f^j, t_f^j)\right|^2.\end{aligned} \tag{23}$$

And the diagonal of the kernel (15) will be made up of $\mathbf{K}_{ii}$, $\mathbf{K}_{bb}$ and $\mathbf{K}_{ff}$. We take an exact soliton solution as the initial value of this problem:

$$q_0(x) = 2b^2\mathrm{sech}(bx)^2, \quad x \in [-5, 5]. \tag{24}$$

and 50 initial sampling points, 50 boundary sampling points and 100 spatial sampling points are ramdomly selected to computing the Loss function (i.e. $\mathbf{K} \in \mathbb{R}^{200\times 200}$).

Fig.4 displays the convergence of initialized NTK of the KdV equation. A series of two-layer FCNNs with a width of $\{100, 2000, 4000, ..., 50000\}$ are used to verify the conclusions of section 3. As N increases, NTK tends to converge at $s = 1$ and tends to diverge at $s = 0.5$. Since there is a product term $6uu_x$ in the KdV equation, F_i (37) is not all equal to constant. And by the analysis in the proof of Theorem 2.2, $s_1 = 3/4$ (39) and $s_2 = 2/3$ (47). Only when $s \geq \max\{s_1, s_2\} = 3/4$, initialized NTK is convergent.

Figure 4 depicts the convergence behavior of the initialized NTK for the KdV equation. A series of two-layer FCNNs with varying widths, namely $\{100, 2000, 4000, ..., 50000\}$, are employed to investigate the conclusions outlined in Section 2. The trend observed is that as the width, N, increases, the NTK tends to converge when $s = 1$, while it tends to diverge when $s = 0.5$. This behavior can be attributed to the presence of the product term $6uu_x$ in the KdV equation, which results in non-constant values for F_i given by Eq. (37). Based on the analysis presented in the proof of Theorem 2.2, it can be determined that $s_1 = 3/4$ given by Eq. (39) and $s_2 = 2/3$ given by Eq. (47). Consequently, only when $s \geq \max s_1, s_2 = 3/4$, the initialized NTK exhibits convergence.

Figure 5 illustrates the convergence of NTK during the training process. Each FCNN width is trained using the standard gradient descent method with a fixed step size of 10^{-5}. The difference between the NTK and initial NTK tends to zero at different training steps. In this experiment, the convergence coefficient s is set to 0.3, thereby validating the result presented in Theorem 3.

4 Conclusions and discussions

In conclusion, we have found some divergent cases of initialized NTK under normal conditions. The convergence of NTK is discussed in more detail, and some results were changed. We add some restrictions in the convergence of initialized NTK (Theorem 2.2), and the convergence condition of NTK during training is relaxed (Theorem 2.3). These findings contribute to a clearer understanding of NTK theory for the general PINNs models, enabling its application to a broader range of problem domains.

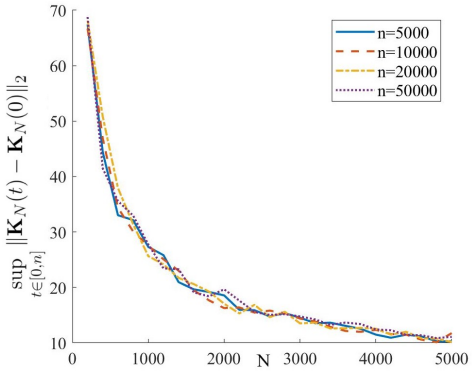

Figure 5: The trends of $\sup_{t\in[0,n]} \|\mathbf{K}_N(t)-\mathbf{K}_N(0)\|_2$ of the initial value problem of the KdV equation when $s=0.2$, where n represents the maximal train step and N the width of the neural networks.

Appendix A. Proof of Lemma 2.1

Proof. In case A, the loss function is

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_f(\boldsymbol{\theta}) + \mathcal{L}_b(\boldsymbol{\theta}) = \frac{\alpha}{2}\sum_{i=1}^{N_f}\left(\mathcal{F}[q](x_f^i,\boldsymbol{\theta}) - f(x_f^i)\right)^2 + \frac{\beta}{2}\sum_{i=1}^{N_b}\left(q_{mx}(x_b^i,\boldsymbol{\theta}) - g(x_b^i)\right)^2, \tag{25}$$

where α, β are weights.

And the gradient flow is

$$\frac{d\boldsymbol{\theta}}{dt} = -\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = -\alpha\sum_{i=1}^{N_f}\left(\mathcal{F}[q](x_f^i,\boldsymbol{\theta}) - f(x_f^i)\right)\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta}) - \beta\sum_{i=1}^{N_b}\left(q_{mx}(x_b^i,\boldsymbol{\theta}) - g(x_b^i)\right)\nabla_{\boldsymbol{\theta}}q_{mx}(x_b^i,\boldsymbol{\theta}). \tag{26}$$

Which is a $(3N+1)$-dimensional equation set. For $i=1,...,N_f$

$$\begin{aligned}
\frac{d\mathcal{F}[q](x_f^i,\boldsymbol{\theta})}{dt} =& \nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta})^T\cdot\frac{d\boldsymbol{\theta}}{dt}\\
=& -\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta})^T\cdot\Bigg[\sum_{j=1}^{N_f}\alpha\left(\mathcal{F}[q](x_f^j,\boldsymbol{\theta}) - f(x_f^j)\right)\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^j,\boldsymbol{\theta})\\
&+\sum_{j=1}^{N_b}\beta\left(q_{mx}(x_b^j,\boldsymbol{\theta}) - g(x_b^j)\right)\nabla_{\boldsymbol{\theta}}q_{mx}(x_b^j,\boldsymbol{\theta})\Bigg]\\
=& -\sum_{j=1}^{N_f}\alpha\left(\mathcal{F}[q](x_f^j,\boldsymbol{\theta}) - f(x_f^j)\right)\left(\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta})\right)^T\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^j,\boldsymbol{\theta})\\
&-\sum_{j=1}^{N_b}\beta\left(q_{mx}(x_b^j,\boldsymbol{\theta}) - g(x_b^j)\right)\left(\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta})\right)^T\nabla_{\boldsymbol{\theta}}q_{mx}(x_b^j,\boldsymbol{\theta}),
\end{aligned} \tag{27}$$

that is,

$$\frac{d\mathcal{F}[q](\boldsymbol{x},\boldsymbol{\theta})}{dt} = -\begin{bmatrix}\mathbf{K}_{ff}, & \mathbf{K}_{fb}\end{bmatrix}\begin{bmatrix}\mathcal{F}[q](\boldsymbol{x}_f,\boldsymbol{\theta}) - f(\boldsymbol{x}_f)\\ q_{mx}(\boldsymbol{x}_b,\boldsymbol{\theta}) - g(\boldsymbol{x}_b)\end{bmatrix}. \tag{28}$$

Similarly, we have

$$\frac{dq_{mx}(\boldsymbol{x}_b,\boldsymbol{\theta})}{dt} = -\begin{bmatrix}\mathbf{K}_{bf}, & \mathbf{K}_{bb}\end{bmatrix}\begin{bmatrix}\mathcal{F}[q](\boldsymbol{x}_f,\boldsymbol{\theta}) - f(\boldsymbol{x}_f)\\ q_{mx}(\boldsymbol{x}_b,\boldsymbol{\theta}) - g(\boldsymbol{x}_b)\end{bmatrix}. \tag{29}$$

□

Appendix B. Proof of Theorem 2.2

Proof. We divide the proof into 3 parts: convergence of $\mathbf{K}_{ff}$, $\mathbf{K}_{bb}$ and $\mathbf{K}_{bf}$.
(1) We first consider the convergence of $\mathbf{K}_{ff}$: We assume $\mathcal{F}[q](x,\boldsymbol{\theta}) = F[q, q_x, q_{xx}, ...q_{nx}](x,\boldsymbol{\theta}) = f(x)$, where $\mathcal{F}[q]$ is a continuous linear or nonlinear function and n is the number of the max order derivative. The partial derivative of $F[q, q_x, q_{xx}, ...q_{nx}](x,\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be written as

$$\frac{\partial F[q, q_x, q_{xx}, ...q_{nx}](x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ...q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{30}$$

where $F_i[q, q_x, q_{xx}, ...q_{nx}](x,\boldsymbol{\theta})$ represent the derivative with respect to q_{ix}. Then for any two data points $x, \hat{x}$, we have

$$\begin{aligned}
\mathbf{K}_{ff} =& \Big\langle \nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x,\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} \mathcal{F}[q](\hat{x},\boldsymbol{\theta}) \Big\rangle \\
=& \Big(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big)^T \Big(\sum_{j=0}^{n} F_j[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Big) \\
=& \sum_{k=0}^{3N+1} \Big(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big)^T \Big(\sum_{j=0}^{n} F_j[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big) \\
=& \sum_{k=0}^{3N} \Big(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big)^T \Big(\sum_{j=0}^{n} F_j[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big) \\
&+ \Big(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(1)}} \Big)^T \Big(\sum_{j=0}^{n} F_j[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(1)}} \Big) \\
=& \sum_{k=0}^{3N} \Big(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big)^T \Big(\sum_{j=0}^{n} F_j[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \Big) \\
&+ F_0[q, q_x, q_{xx}, ..., q_{nx}] F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}],
\end{aligned} \tag{31}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^{(0),T}, \boldsymbol{b}^{(0),T}, \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}\} \in \mathbb{R}^{3N+1}$. Recall that

$$\begin{aligned}
q(x,\boldsymbol{\theta}) =& \frac{1}{N^s} \sum_{k=1}^{N} \boldsymbol{W}_k^{(1)} \sigma_k + \boldsymbol{b}^{(1)}, \\
q_{ix}(x,\boldsymbol{\theta}) =& \frac{1}{N^s} \sum_{k=1}^{N} \boldsymbol{W}_k^{(1)} \sigma_k^{(i)} \boldsymbol{W}_k^{(0)i}, \quad i = 1, ..., n,
\end{aligned} \tag{32}$$

where $\sigma_k^{(i)}$ represent $\sigma^{(i)}\left(\boldsymbol{W}_k^{(0)} x + \boldsymbol{b}_k^{(0)}\right)$. In order to discuss the convergence of $\mathbf{K}_{ff}$, the coefficient $\frac{1}{N^s}$ will be discussed.

(a) As $F_0[q, q_x, q_{xx}, ..., q_{nx}] F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \neq 0$ or exist a monomial in $\{F_i[q, q_x, q_{xx}, ...q_{nx}](x,\boldsymbol{\theta})\}_{i=1}^n$ is non-homogeneous.

Definition: (homogeneous) If $F_i[q, q_x, q_{xx}, ...q_{nx}]$ is a homogeneous operator, there exist $\{t_i\}_{i=0}^n \subset \mathbb{N}$, such that for any $\{m_i\}_{i=0}^n \subset \mathbb{R}$:

$$F[m_0 q, m_1 q_x, m_2 q_{xx}, ..., m_n q_{nx}] = m_0^{t_i} m_1^{t_i} m_2^{t_i} ... m_n^{t_i} F[q, q_x, q_{xx}, ..., q_{nx}] \tag{33}$$

To ensure the convergence of this term, the exponential s should be equal or greater than 1 according to the law of large numbers. When $s = 1$, we have

$$\begin{aligned}
q(x,\boldsymbol{\theta}) =& \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{W}_k^{(1)} \sigma_k + \boldsymbol{b}^{(1)} \xrightarrow{\mathcal{P}} \mathbb{E}[\boldsymbol{W}_k^{(1)} \sigma_k] + \boldsymbol{b}^{(1)} =: \mathbb{E}[q], \\
q_{ix}(x,\boldsymbol{\theta}) =& \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{W}_k^{(1)} \sigma_k^{(i)} \boldsymbol{W}_k^{(0)i} \xrightarrow{\mathcal{P}} \mathbb{E}[\boldsymbol{W}_k^{(1)} \sigma_k^{(i)} \boldsymbol{W}_k^{(0)i}] =: \mathbb{E}[q_i], \quad i = 1, ..., n
\end{aligned} \tag{34}$$

Then by the law of large numbers, we have

$$\mathbf{K}_{ff} \xrightarrow{\mathcal{P}} F_0\left[\mathbb{E}[q], \mathbb{E}[q_1], \mathbb{E}[q_2], ..., \mathbb{E}[q_n]\right] F_0\left[\mathbb{E}[\hat{q}], \mathbb{E}[\hat{q}_1], \mathbb{E}[\hat{q}_2], ..., \mathbb{E}[\hat{q}_n]\right]. \tag{35}$$

The first $3N$ terms in above formula have a coefficient $\frac{1}{N^{2s}}$. Therefore, when $s > \frac{1}{2}$, the first $3N$ terms converge to 0.

(b) As every monomial in $\{F_i[q, q_x, q_{xx}, ..., q_{nx}](x, \boldsymbol{\theta})\}_{i=0}^n$ is homogeneous, and

$$F_0[q, q_x, q_{xx}, ..., q_{nx}]F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] = 0, \tag{36}$$

then the convergence is determined by the maximum order term of F_i (The order is defined by the sum of homogeneous parameters $\sum_{l=0}^{n} t_{il}$ (33).). We assume the number of maximum order is n_i for F_i, and we rewrite the F_i as

$$F_i = F_i^0 + F_i^1, \tag{37}$$

where F_i^0 represent the all max order terms in F_i, and F_i^1 is the remainder terms of F_i. Then, for any two data points $x, \hat{x}$, we have

$$\begin{aligned}\mathbf{K}_{ff} &= \sum_{k=0}^{3N+1}\left(\sum_{i=0}^{n}\left(F_i^0 + F_i^1\right)\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\right)\left(\sum_{j=0}^{n}\left(\bar{F}_j^0 + \bar{F}_j^1\right)\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\right)\\ &= \sum_{k=0}^{3N+1}\left(\sum_{i,j=0}^{n}\left(F_i^0\bar{F}_j^0 + F_i^1\bar{F}_j^0 + F_i^0\bar{F}_j^1 + F_i^1\bar{F}_j^1\right)\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\right)\end{aligned} \tag{38}$$

When

$$s = \frac{2\sum_{l=0}^{n} t_{i'l} + 1}{2\sum_{l=0}^{n} t_{i'l} + 2} =: s_1 \tag{39}$$

where $\sum_{l=0}^{n} t_{i'l} = \max_{i=0,...,n}\left\{\sum_{l=0}^{n} t_{il}\right\}$. By the law of big numbers, we have

$$\begin{aligned}\mathbf{K}_{ff} =& \left(N^{1-2s}\sum_{i,j=0}^{n}\left(F_i^0\bar{F}_j^0 + F_i^1\bar{F}_j^0 + F_i^0\bar{F}_j^1 + F_i^1\bar{F}_j^1\right)N^{2s-1}\sum_{k=0}^{3N}\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}\right)\\ \xrightarrow{\mathcal{P}} & F_{i'}^0\bar{F}_{j'}^0\left(\mathbb{E}\left[\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(0)}}\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(0)}}\right] + \mathbb{E}\left[\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(1)}}\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(1)}}\right] + \mathbb{E}\left[\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{b}_k^{(0)}}\frac{\partial q_{jx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{b}_k^{(0)}}\right]\right)\end{aligned} \tag{40}$$

(2) $\mathbf{K}_{bb}$: we consider $\mathbf{K}_{bb}$ when $s = \frac{1}{2}$:

$$\begin{aligned}\mathbf{K}_{bb} =& \left\langle\frac{\partial q_{mx}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T, \frac{\partial q_{mx}(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right\rangle\\ =& \left\langle\frac{\partial q(x,\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(0)}}^T, \frac{\partial q(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(0)}}\right\rangle + \left\langle\frac{\partial q(x,\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(1)}}^T, \frac{\partial q(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(1)}}\right\rangle\\ &+ \left\langle\frac{\partial q(x,\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(0)}}^T, \frac{\partial q(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(0)}}\right\rangle + \left\langle\frac{\partial q(x,\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(1)}}^T, \frac{\partial q(\hat{x},\boldsymbol{\theta})}{\partial \boldsymbol{b}^{(1)}}\right\rangle\end{aligned} \tag{41}$$

(a) when $m = 0$. Recall that

$$\boldsymbol{q}(x,\boldsymbol{\theta}) = \frac{1}{N^s}\boldsymbol{W}^{(1)}\cdot\sigma(\boldsymbol{W}^{(0)}\cdot x + \boldsymbol{b}^{(0)}) + \boldsymbol{b}^{(1)} \tag{42}$$

for any two points x and $\hat{x}$

$$\begin{aligned}\mathbf{K}_{bb} =& \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{W}_k^{(1)2}\sigma_k'\hat{\sigma}_k' x\hat{x}\right) + \frac{1}{N}\sum_{k=1}^{N}\sigma_k\hat{\sigma}_k + \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{W}_k^{(1)2}\sigma_k'\hat{\sigma}_k'\right) + 1\\ \xrightarrow{\mathcal{P}} & \mathbb{E}\left[\boldsymbol{W}_k^{(1)2}\sigma_k'\hat{\sigma}_k'\right]x\hat{x} + \mathbb{E}\left[\sigma_k\hat{\sigma}_k\right] + \mathbb{E}\left[\boldsymbol{W}_k^{(1)2}\sigma_k\hat{\sigma}_k\right] + 1\end{aligned} \tag{43}$$

(b) when $m > 0$. Recall that

$$q_{mx}(x,\boldsymbol{\theta}) = \frac{1}{N^s}\sum_{k=1}^{N}\boldsymbol{W}_k^{(1)}\sigma_k^{(m)}\boldsymbol{W}_k^{(0)m}, \tag{44}$$

$$
\begin{aligned}
\mathbf{K}_{bb} =& \frac{1}{N^{2s}} \sum_{k=1}^{N} \Big(\boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)2m} x\hat{x} + m^2 \boldsymbol{W}_k^{(1)2} \sigma_k^{(m)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)2(m-1)} \\
& + m \boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)(2m-1)} x + m \boldsymbol{W}_k^{(1)2} \sigma_k^{(m)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)(2m-1)} \hat{x} \Big) \\
& + \frac{1}{N^{2s}} \sum_{k=1}^{N} \left(\sigma_k^{(m)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)2m} \right) + \frac{1}{N^{2s}} \sum_{k=1}^{N} \left(\boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)2m} \right) + 0 \\
\xrightarrow{\mathcal{P}} & \mathbb{E}\left[\boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)2m} \right] x\hat{x} + \mathbb{E}\left[m^2 \boldsymbol{W}_k^{(1)2} \sigma_k^{(m)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)2(m-1)} \right] \\
& + \mathbb{E}\left[m \boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)(2m-1)} \right] x + \mathbb{E}\left[m \boldsymbol{W}_k^{(1)2} \sigma_k^{(m)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)(2m-1)} \right] \hat{x} \\
& + \mathbb{E}\left[\sigma_k^{(m)} \hat{\sigma}_k^{(m)} \boldsymbol{W}_k^{(0)2m} \right] + \mathbb{E}\left[\boldsymbol{W}_k^{(1)2} \sigma_k^{(m+1)} \hat{\sigma}_k^{(m+1)} \boldsymbol{W}_k^{(0)2m} \right]
\end{aligned} \tag{45}
$$

(3) $\mathbf{K}_{fb} (= \mathbf{K}_{bf}^T)$: discussion similar to (1):

(a) When $F_0[q, q_x, q_{xx}, ...q_{nx}] \neq 0$ or exist a monomial in the polynomial in $\{F_i[q, q_x, q_{xx}, ...q_{nx}](x, \boldsymbol{\theta})\}_{i=1}^n$ is non-homogeneous. To ensure the convergence of this term, the exponential s should be equal or greater than 1 according to the law of large numbers. Therefore, when $s = 1$, for any two data points x, $\hat{x}$, we have

$$
\begin{aligned}
\mathbf{K}_{fb} =& \Big\langle \nabla_{\boldsymbol{\theta}} \mathcal{F}[q](x, \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} q_{mx}(\hat{x}, \boldsymbol{\theta}) \Big\rangle \\
=& \sum_{k=0}^{3N} \left(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right) \left(\frac{\partial q_{mx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right) \\
& + F_0[q, q_x, q_{xx}, ...q_{nx}] \frac{\partial q_{mx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{b}^{(1)}} \\
\xrightarrow{\mathcal{P}} & \begin{cases} F_0\big[\mathbb{E}[q], \mathbb{E}[q_1], \mathbb{E}[q_2], ..., \mathbb{E}[q_n]\big], & m = 0, \\ 0, & m > 0. \end{cases}
\end{aligned} \tag{46}
$$

(b) When $F_0[q, q_x, q_{xx}, ...q_{nx}] = 0$ and every monomial in the polynomials in $\{F_i[q, q_x, q_{xx}, ..., q_{nx}](x, \boldsymbol{\theta})\}_{i=1}^n$ is homogeneous. To balance the coefficient of the neural network, we set

$$
s = \frac{\sum_{l=0}^{n} t_{i'l} + 1}{\sum_{l=0}^{n} t_{i'l} + 2} =: s_2 \tag{47}
$$

($\sum_{l=0}^{n} t_{i'l} = \max_{i=0,...,n} \left\{ \sum_{l=0}^{n} t_{il} \right\}$ and $\{t_{il}\}$ are defined in (33)). And for any $i, j \in \{0, 1, ..., n\}$, we define

$$
\begin{aligned}
N^{2s-1} \left(\frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \left(\frac{\partial q_{jx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) =& \frac{1}{N} \sum_{k=0}^{3N+1} N^s \left(\frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right)^T N^s \left(\frac{\partial q_{jx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right) \\
& \xrightarrow{\mathcal{P}} \mathbb{E}[q_i, q_j]
\end{aligned} \tag{48}
$$

Then, we have

$$
\begin{aligned}
\mathbf{K}_{fb} =& \sum_{k=0}^{3N} \left(\sum_{i=0}^{n} F_i[q, q_x, q_{xx}, ..., q_{nx}] \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right) \left(\frac{\partial q_{mx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right) \\
=& \sum_{i=0}^{n} N^{1-2s} (F_i^0 + F_i^1) N^{2s-1} \sum_{k=0}^{3N} \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \frac{\partial q_{mx}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \\
\xrightarrow{\mathcal{P}} & F_{i'}^0 \mathbb{E}[q_i, q_j]
\end{aligned} \tag{49}
$$

In summary, $\mathbf{K}_{bb}$ convergence by probability when $s \geq \frac{1}{2}$. If

$$
F_0[q, q_x, q_{xx}, ..., q_{nx}] F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] \neq 0 \tag{50}
$$

($F_0[q, q_x, q_{xx}, ..., q_{nx}] \neq 0$) or exist a monomial in the polynomial in $\{F[q, q_x, q_{xx}, ...q_{nx}](x, \boldsymbol{\theta})\}_{i=1}^n$ is non-homogeneous (case A), $\mathbf{K}_{ff}$ ($\mathbf{K}_{bf}$) convergence by probability when $s \geq 1$. And if $F_0[q, q_x, q_{xx}, ..., q_{nx}]F_0[\hat{q}, \hat{q}_x, \hat{q}_{xx}, ..., \hat{q}_{nx}] = 0$ ($F_0[q, q_x, q_{xx}, ..., q_{nx}] = 0$) or every monomial in the polynomials in $\{F_i[q, q_x, q_{xx}, ..., q_{nx}](x, \boldsymbol{\theta})\}_{i=1}^n$ is homogeneous (case B), $\mathbf{K}_{ff}$ ($\mathbf{K}_{bf}$) convergence by probability when $s \geq s_1$(39) ($s \geq s_2$(47)).

In other words, in case A, $\mathbf{K}$ convergence by probability when $s \geq 1$. In case B, $\mathbf{K}$ convergence by probability when $s \geq s_1$. □

Appendix C. Proof of Theorem 2.3

Before we prove Theorem 2.3, we first prove a couple of lemmas. We add a discussion of the general PDE based on the original proof.

Lemma C.1 under the condition of Theorem 2.3, we have

$$\begin{aligned} \sup_{t\in[0,T]} \left\| \frac{\partial q_{mx}}{\partial \boldsymbol{\theta}} \right\|_\infty =& \mathcal{O}(\frac{1}{N^s}) \\ \sup_{t\in[0,T]} \left\| \frac{\partial \mathcal{F}[q]}{\partial \boldsymbol{\theta}} \right\|_\infty =& \mathcal{O}(\frac{1}{N^s}) \end{aligned} \tag{51}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}^{(0),T}, \boldsymbol{b}^{(0),T}, \boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}\}$.

Proof. We still assume $\mathcal{F}[q](x, \boldsymbol{\theta}) = F[q, q_x, q_{xx}, ...q_{nx}](x, \boldsymbol{\theta}) = f(x)$. recall that

$$\begin{aligned} q(x, \boldsymbol{\theta}) =& \frac{1}{N^s} \sum_{k=1}^N \boldsymbol{W}_k^{(1)} \sigma\left(\boldsymbol{W}_k^{(0)} x + \boldsymbol{b}_k^{(0)}\right) + \boldsymbol{b}^{(1)}, \\ q_{mx}(x, \boldsymbol{\theta}) =& \frac{1}{N^s} \sum_{k=1}^N \boldsymbol{W}_k^{(1)} \sigma^{(m)} \boldsymbol{W}_k^{(0)m}, \end{aligned} \tag{52}$$

and

$$\frac{\partial \mathcal{F}[q](x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \sum_{i=0}^n F_i[q, q_x, q_{xx}, ...q_{nx}] \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}. \tag{53}$$

By the Uniformly boundedness of weight, derivative of activation function $\sigma^{(k)}$ and F_i (assumptions (i), (ii), (iv)), We have

$$\begin{aligned} \sup_{t\in[0,T]} \left\| \frac{\partial \mathcal{F}[q]}{\partial \boldsymbol{\theta}_k} \right\|_\infty =& \sup_{t\in[0,T]} \left\| \sum_{i=0}^n F_i[q, q_x, q_{xx}, ...q_{nx}] \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right\|_\infty \\ \leq& \sum_{i=0}^n \sup_{t\in[0,T]} \|F_i[q, q_x, q_{xx}, ...q_{nx}]\|_\infty \sup_{t\in[0,T]} \left\| \frac{\partial q_{ix}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \right\|_\infty \\ \leq& nC\frac{C}{N^s} = O(\frac{1}{N^s}), \quad k = 1, 2, ..., 3N + 1. \end{aligned} \tag{54}$$

This completes the proof. □

Lemma C.2 under the condition of Theorem 2.3, we have

$$\lim_{N\to\infty} \sup_{t\in[0,T]} \left\| \frac{1}{N^s}(\boldsymbol{\theta}(t) - \boldsymbol{\theta}(0)) \right\|_2 = 0 \tag{55}$$

Proof.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_f(\boldsymbol{\theta}) + \mathcal{L}_b(\boldsymbol{\theta}) = \frac{1}{2} \sum_{j=1}^{N_f} \left| \mathcal{F}[q](x_f^j, \boldsymbol{\theta}) - f(x_f^j) \right|^2 + \frac{1}{2} \sum_{j=1}^{N_b} \left| q_{mx}(x_b^j, \boldsymbol{\theta}) - g(x_b^j) \right|^2, \tag{56}$$

Because of

$$\frac{d\boldsymbol{\theta}}{dt} = -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}), \tag{57}$$

we have

$$\begin{aligned}
&\left\|\frac{1}{N^s}(\boldsymbol{\theta}(t)-\boldsymbol{\theta}(0))\right\|_2 = \left\|\frac{1}{N^s}\int_0^t \frac{d\boldsymbol{\theta}(\tau)}{d\tau}d\tau\right\|_2 = \left\|\frac{1}{N^s}\int_0^t \frac{\partial \mathcal{L}(\boldsymbol{\theta}(\tau))}{\partial \boldsymbol{\theta}}d\tau\right\|_2 \\
&= \left\|\frac{1}{N^s}\int_0^t \left[\sum_{j=1}^{N_f}(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j))\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \sum_{j=1}^{N_b}(\frac{\partial^m q(x_b^j,\boldsymbol{\theta})}{\partial x^m}-g(x_b^j))\frac{\partial q_{mx}(x_b^j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]d\tau\right\|_2 \\
&\leq \boldsymbol{A}_1+\boldsymbol{A}_2
\end{aligned} \tag{58}$$

where

$$\begin{aligned}
\boldsymbol{A}_1 &= \left\|\frac{1}{N^s}\int_0^t \left[\sum_{j=1}^{N_f}(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j))\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]d\tau\right\|_2 \\
&\leq \frac{1}{N^s}\int_0^t \left\|\alpha\sum_{j=1}^{N_f}(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j))\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right\|_2 d\tau \\
&= \frac{1}{N^s}\int_0^t \sqrt{\sum_{k=1}^{3N+1}\left(\sum_{j=1}^{N_f}(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j))\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(l)}}\right)^2}\,d\tau \\
&\leq \frac{1}{N^s}\int_0^t \left\|\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(l)}}\right\|_\infty \sqrt{\sum_{k=1}^{3N+1}\left(\sum_{j=1}^{N_f}(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j))\right)^2}\,d\tau \\
&= \frac{C}{N^{s-1/2}}\int_0^t \left\|\frac{\partial \mathcal{F}[q](x_f^j,\boldsymbol{\theta})}{\partial \boldsymbol{W}_k^{(l)}}\right\|_\infty \left|\sum_{j=1}^{N_f}\left(\mathcal{F}[q](x_f^j,\boldsymbol{\theta})-f(x_f^j)\right)\right| d\tau \\
&= \mathcal{O}(\frac{1}{N^{2s-1/2}}).
\end{aligned} \tag{59}$$

In the same way, we have

$$\boldsymbol{A}_2 = \left\|\frac{1}{N^s}\int_0^t \left[\sum_{j=1}^{N_b}(\frac{\partial^m q(x_b^j,\boldsymbol{\theta})}{\partial x^m}-g(x_b^j))\frac{\partial q_{mx}(x_b^j,\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(l)}}\right]d\tau\right\|_2 = \mathcal{O}(\frac{1}{N^{2s-1/2}}). \tag{60}$$

□

Lemma C.3 Under the condition of Theorem 2.3, when $s > \frac{1}{4}$ we have

$$\lim_{N\to\infty}\sup_{t\in[0,T]}\left\|\frac{1}{N^s}(\sigma^{(k)}|_{t=T}-\sigma^{(k)}|_{t=0})\right\|_2 = 0 \tag{61}$$

Proof. by the mean-value theorem for $\sigma^{(k)}$ and Lemma C.2 . □

Lemma C.4 under the condition of Theorem 2.3, we have

$$\begin{aligned}
\lim_{N\to\infty}\sup_{t\in[0,T]}\left\|\frac{\partial q_{mx}(x,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}}-\frac{\partial q_{mx}(x,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_2 = 0 \\
\lim_{N\to\infty}\sup_{t\in[0,T]}\left\|\frac{\partial \mathcal{F}[q](x,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}}-\frac{\partial \mathcal{F}[q](x,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_2 = 0
\end{aligned} \tag{62}$$

Proof. Recall that

$$\frac{\partial \mathcal{F}[q](x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \sum_{i=0}^{n} F_i[q,q_x,q_{xx},...q_{nx}]\frac{\partial q_{ix}(x,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k}. \tag{63}$$

Take $\boldsymbol{W}_k^{(0)}$ for example, we have

$$
\begin{aligned}
&\sup_{t\in[0,T]}\left\|\frac{\partial\mathcal{F}[q](x,\boldsymbol{\theta}(t))}{\partial\boldsymbol{W}_k^{(0)}}-\frac{\partial\mathcal{F}[q](x,\boldsymbol{\theta}(t))}{\partial\boldsymbol{W}_k^{(0)}}\right\|_2\\
=&\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...q_{nx}]\frac{\partial q_{ix}(x,\boldsymbol{\theta}(t))}{\partial\boldsymbol{W}_k^{(0)}}-\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...q_{nx}]\frac{\partial q_{ix}(x,\boldsymbol{\theta}(0))}{\partial\boldsymbol{W}_k^{(0)}}\right\|_2\\
=&\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\Big(\Big(\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)\boldsymbol{W}_k^{(0)i}(t)x+\frac{i}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i)}(t)\boldsymbol{W}_k^{(0)(i-1)}(t)\Big)\right.\\
&\left.-\Big(\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\boldsymbol{W}_k^{(0)i}(0)x+\frac{i}{N^s}\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i)}(0)\boldsymbol{W}_k^{(0)(i-1)}(0)\Big)\right)\right\|_2\\
\le&\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\Big(\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)\boldsymbol{W}_k^{(0)i}(t)x-\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\boldsymbol{W}_k^{(0)i}(0)x\Big)\right\|_2\\
&+\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\Big(\frac{i}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i)}(t)\boldsymbol{W}_k^{(0)(i-1)}(t)-\frac{i}{N^s}\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i)}(0)\boldsymbol{W}_k^{(0)(i-1)}(0)\Big)\right\|_2\\
=&A_1+A_2.
\end{aligned}
\tag{64}
$$

Then

$$
\begin{aligned}
A_1=&\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\Big(\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)\boldsymbol{W}_k^{(0)i}(t)x-\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\boldsymbol{W}_k^{(0)i}(0)x\Big)\right\|_2\\
\le&\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\frac{1}{N^s}\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)\Big(\boldsymbol{W}_k^{(0)i}(t)-\boldsymbol{W}_k^{(0)i}(0)\Big)x\right\|_2\\
&+\sup_{t\in[0,T]}\left\|\sum_{i=0}^{n}F_i[q,q_x,q_{xx},...,q_{nx}]\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)-\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\Big)\boldsymbol{W}_k^{(0)i}(0)x\right\|_2\\
\le&\sum_{i=0}^{n}\sup_{t\in[0,T]}\left\|F_i[q,q_x,q_{xx},...,q_{nx}]\right\|_2\left\|\boldsymbol{W}_k^{(1)}(t)\right\|_2\left\|\sigma_k^{(i+1)}(t)\right\|_2\left\|\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(0)i}(t)-\boldsymbol{W}_k^{(0)i}(0)\Big)x\right\|_2\\
&+\sum_{i=0}^{n}\sup_{t\in[0,T]}\left\|F_i[q,q_x,q_{xx},...,q_{nx}]\right\|_2\left\|\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)-\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\Big)\right\|_2\left\|\boldsymbol{W}_k^{(0)i}(0)x\right\|_2
\end{aligned}
\tag{65}
$$

Because

$$
\begin{aligned}
&\left\|\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)-\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\Big)\right\|_2\\
\le&\left\|\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(t)-\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(0)\Big)\right\|_2+\left\|\frac{1}{N^s}\Big(\boldsymbol{W}_k^{(1)}(t)\sigma_k^{(i+1)}(0)-\boldsymbol{W}_k^{(1)}(0)\sigma_k^{(i+1)}(0)\Big)\right\|_2\\
\le&\frac{1}{N^s}\left\|\boldsymbol{W}_k^{(1)}(t)\right\|_2\left\|\sigma_k^{(i+1)}(t)-\sigma_k^{(i+1)}(0)\right\|_2+\frac{1}{N^s}\left\|\boldsymbol{W}_k^{(1)}(t)-\boldsymbol{W}_k^{(1)}(0)\right\|_2\left\|\sigma_k^{(i+1)}(0)\right\|_2\\
=&\mathcal{O}(\frac{1}{N^{2s-1/2}}).
\end{aligned}
\tag{66}
$$

Then we have

$$
A_1=\mathcal{O}(\frac{1}{N^{2s-1/2}}).
\tag{67}
$$

By the same way

$$
A_2=\mathcal{O}(\frac{1}{N^{2s-1/2}}).
\tag{68}
$$

and the same method can be used to $\boldsymbol{W}_k^{(1)}, \boldsymbol{b}_k^{(0)}, \boldsymbol{b}_k^{(1)}$. Therefore

$$\begin{aligned}\left\|\frac{\partial \mathcal{F}[q](x,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}} - \frac{\partial \mathcal{F}[q](x,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_2 &= \mathcal{O}(\frac{1}{N^{2s-1/2}}).\\ \left\|\frac{\partial q_{mx}(x,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}} - \frac{\partial q_{mx}(x,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_2 &= \mathcal{O}(\frac{1}{N^{2s-1/2}}).\end{aligned} \tag{69}$$

□

When the above-mentioned four Lemmas are finished, the raw theorem can be proved.

Proof of Theorem 2.3. The Kernel matrix K can be divided into the multiply of the Jacobian matrix:

$$\mathbf{K}(t) = \begin{bmatrix}\mathbf{K}_{ff}(t) & \mathbf{K}_{fb}(t)\\ \mathbf{K}_{bf}(t) & \mathbf{K}_{bb}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{J}_f(t)\\ \mathbf{J}_b(t)\end{bmatrix}\left[\mathbf{J}_f^T(t), \quad \mathbf{J}_b^T(t)\right] := \mathbf{J}(t)\mathbf{J}^T(t), \tag{70}$$

where $\mathbf{J}_f(t)$

$$\begin{aligned}\mathbf{J}_f(t) &= \left(\nabla_{\boldsymbol{\theta}}\mathcal{F}[q](x_f^i,\boldsymbol{\theta})\right)_{i=1,\dots,N_f} \in \mathbb{R}^{N_f\times(3N+1)},\\ \mathbf{J}_b(t) &= \left(\nabla_{\boldsymbol{\theta}} q_{mx}(x_b^i,\boldsymbol{\theta})\right)_{i=1,\dots,N_b} \in \mathbb{R}^{N_b\times(3N+1)},\end{aligned} \tag{71}$$

Then we have

$$\begin{aligned}\|\mathbf{K}(t)-\mathbf{K}(0)\|_2 &= \left\|\mathbf{J}(t)\mathbf{J}^T(t) - \mathbf{J}(0)\mathbf{J}^T(0)\right\|_2\\ &\le \left\|(\mathbf{J}(t)-\mathbf{J}(0))\mathbf{J}^T(t)\right\|_2 + \left\|\mathbf{J}(0)(\mathbf{J}^T(t)-\mathbf{J}^T(0))\right\|_2\\ &\le \|\mathbf{J}(t)-\mathbf{J}(0)\|_2\left\|\mathbf{J}^T(t)\right\|_2 + \|\mathbf{J}(0)\|_2\left\|\mathbf{J}^T(t)-\mathbf{J}^T(0)\right\|_2.\end{aligned} \tag{72}$$

By Lemma C.1, $\left\|\mathbf{J}^T(t)\right\|_F$ and $\|\mathbf{J}(0)\|_F$ are bounded. And because of the equivalence of norm, $\left\|\mathbf{J}^T(t)\right\|_2$ and $\|\mathbf{J}(0)\|_2$ are bounded.

Then, we consider the convergence of $\left\|\mathbf{J}^T(t)-\mathbf{J}^T(0)\right\|_2$. By Lemma C.4, it is easy to see that

$$\begin{aligned}\|\mathbf{J}(t)-\mathbf{J}(0)\|_F^2 =& \sum_{i=1}^{N_f}\left\|\frac{\partial \mathcal{F}[q](x_i,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}} - \frac{\partial \mathcal{F}[q](x_i,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_F^2\\ &+ \sum_{i=1}^{N_b}\left\|\frac{\partial q_{mx}(x,\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}} - \frac{\partial q_{mx}(x,\boldsymbol{\theta}(0))}{\partial \boldsymbol{\theta}}\right\|_F^2\\ =&\mathcal{O}(\frac{1}{N^{4s-1}}).\end{aligned} \tag{73}$$

Thus, $\|\mathbf{J}(t)-\mathbf{J}(0)\|_F$ is converge to 0 when $N\to\infty$. And because of the equivalence of norm, $\|\mathbf{J}(t)-\mathbf{J}(0)\|_2$ is converge to 0 when $N\to\infty$.

Then we have $\|\mathbf{K}(t)-\mathbf{K}(0)\|_2 \to 0$, when $N\to\infty$. □

Acknowledgement

The work was supported by the National Natural Science Foundation of China under Grant Nos. 11925108 and 12226332.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.

[2] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning (MIT Press, 2016).

[3] M. Dissanayake, N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, Commun. Numer. Methods Eng. 10 (1994) 195-201.

[4] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Netw. 9 (1998) 987-1000.

[5] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signal Systems, 2 (1989) 3.
[6] K. Hornik, Approximation Capabilities of Muitilayer Feedforward Networks, Neural Networks, 4 (1991) 251-257.
[7] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339-1364.
[8] M. Raissi, P. Perdikaris, G. E. Karniadakisa, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686-707.
[9] W. E, B. Yu, The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems, Commun. Math. Stat. 6 (2018) 1-12.
[10] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-Net: Learning PDEs from Data, PMLR 80 (2018) 3208-3216.
[11] Z. Long, Y. Lu, B. Dong, PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, J. Comput. Phys. 399 (2019) 108925.
[12] Y. Bar-Sinaia, S. Hoyerb, J. Hickeyb, and M. P. Brennera, Learning data-driven discretizations for partial differential equations, PNAS, 116 (2019) 15344-15349.
[13] Y. Wang, Z. Shen, Z. Long and B. Dong, Learning to Discretize: Solving 1D Scalar Conservation Laws via Deep Reinforcement Learning, (2019) arXiv:1905.11079.
[14] L. Lu, P. Jin, and G. E. Karniadakis, DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, (2019) arXiv:1910.03193.
[15] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier Neural Operator for parametric partial differential equations, (2021) arXiv:2010.08895v2.
[16] K. Wu, D. Xiu, Data-driven deep learning of partial differential equations in modal space, J. Comput. Phys. 408 (2020) 109307.
[17] Y. Khoo, J. Lu and L. Ying, Solving parametric PDE problems with artificial neural networks, Euro. J. Appl. Math. 32 (2021) 421-435.
[18] G. M. Rotskoff, E. Vanden-Eijnden, Learning with rare data: using active importance sampling to optimize objectives dominated by rare events. PMLR 145 (2022) 757-780.
[19] D. Pfau, J. S. Spencer, A. G. Matthews, W. M. C. Foulkes, Ab initio solution of the many electron Schrödinger equation with deep neural networks. Phys. Rev. Res. 2 (2020) 033429.
[20] S. Goswamia, M. Yinb, Y. Yu, G. E. Karniadakis, A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114587.
[21] S. Lin, Y. Chen, A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions, J. Comput. Phys. 457 (2022) 111053.
[22] S. Lin, Y. Chen, Physics-informed neural network methods based on Miura transformations and discovery of new localized wave solutions, Physica D 445 (2023) 133629.
[23] J. Li, Y. Chen, A deep learning method for solving third-order nonlinear evolution equations, Commun. Theor. Phys. 72 (2020) 115003.
[24] J. Li, J. Chen, B. Li, Gradient-optimized physics-informed neural networks (GOPINNs): a deep learning method for solving the complex modified KdV equation, Nonlinear Dyn. 107 (2022) 781-792.
[25] Z. Miao, Y. Chen, VC-PINN: Variable Coefficient Physical Information Neural Network For Forward And Inverse PDE Problems with Variable Coefficient, arXiv preprint arXiv:2305.07479.
[26] J. Pu, Y. Chen, Complex dynamics on the one-dimensional quantum droplets via time piecewise PINNs, Physica D 454 (2023) 133851.
[27] L. Wang, Z. Yan, Data-driven peakon and periodic peakon solutions and parameter discovery of some nonlinear dispersive equations via deep learning, Physica D 428 (2021) 133037.
[28] Z. Zhou, Z. Yan, Deep learning neural networks for the third-order nonlinear Schr?dinger equation: bright solitons, breathers, and rogue waves, Commun. Theor. Phys. 73 (2021) 105006.
[29] J. Song, Z. Yan, Deep learning soliton dynamics and complex potentials recognition for 1D and 2D PT-symmetric saturable nonlinear Schrödinger equations, Physica D 448 (2023) 133729.
[30] M. Zhong, S. Gong, S.-F. Tian, Z. Yan, Data-driven rogue waves and parameters discovery in nearly integrable PT-symmetric Gross-Pitaevskii equations via PINNs deep learning, Physica D 439 (2022) 133430.
[31] Shuning Lin a, Yong Chen, A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions, J. Comput. Phys. (2022)
[32] Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. (2019c). Harnessing the power of infinitely wide deep nets on small-data tasks. arXiv preprint arXiv:1910.01663.
[33] Y. Shin, J. Darbon, G. E. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, Commun. Comput. Phys. 28 (2020) 2042-2074.
[34] T. Luo, H. Yang, Two-layer neural networks for partial differential equations: optimization and generalization theory. (2020) arXiv:2006.15733.
[35] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: convergence and generalization in neural networks. Adv. Neural Inf. Process. Syst. 31 (2018) 8571-8580.
[36] J. Martens, A. Ballard, G. Desjardins, G. Swirszcz, V. Dalibard, J. SohlDickstein, and S. S. Schoenholz, Rapid training of deep neural networks without skip connections or normalization layers using deep kernel shaping. arXiv preprint arXiv:2110.01765, (2021).

[37] E. Dyer and G. Gur-Ari, Asymptotics of wide networks from feynman diagrams. In International Conference on Learning Representations. (2020).

[38] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi, Kernel methods through the roof: handling billions of points efficiently. arXiv preprint arXiv:2006.10350.(2020)

[39] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang, On exact computation with an infinitely wide neural net. In Advances in Neural Information Processing Systems, (2019b) pages 8141-8150.

[40] S.Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: a neural tangent kernel perspective. J. Comput. Phys. 449 (2022) 110768.

[41] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains. arXiv preprint arXiv:2006.10739.(2020).

[42] S. Wang, H. Wang, P. Perdikaris, On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 384 (2021) 113938.

[43] M. J. Ablowitz, P. A. Clarkson, Solitons, Nonlinear Evolution Equations and Inverse Scattering (Cambridge University Press, Cambridge, 1991).