GLOBAL CONVERGENCE OF DEEP GALERKIN AND PINNS METHODS FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

DEQING JIANG*, JUSTIN SIRIGNANO*, AND SAMUEL N. COHEN*

Abstract. Numerically solving high-dimensional partial differential equations (PDEs) is a major challenge. Conventional methods, such as finite difference methods, are unable to solve high-dimensional PDEs due to the curse-of-dimensionality. This is in particular a fundamental challenge for the solution of financial models, which are often inherently high-dimensional. Option pricing, hedging, mean-field financial models, order book models, and dynamic portfolio investment can all require the solution of high-dimensional PDEs. A variety of deep learning methods have been recently developed to try and solve high-dimensional PDEs by approximating the solution using a neural network. These deep learning methods have been widely applied to high-dimensional PDEs in financial engineering. In this paper, we prove global convergence for one of the commonly-used deep learning algorithms for solving PDEs, the Deep Galerkin Method (DGM). DGM trains a neural network approximator to solve the PDE using stochastic gradient descent. We prove that, as the number of hidden units in the single-layer network goes to infinity (i.e., in the "wide network limit"), the trained neural network converges to the solution of an infinite-dimensional linear ordinary differential equation (ODE). The PDE residual of the limiting approximator converges to zero as the training time $\to \infty$. Under mild assumptions, this convergence also implies that the neural network approximator converges to the solution of the PDE. A closely related class of deep learning methods for PDEs is Physics Informed Neural Networks (PINNs), which has been widely-used in a variety of fields (including financial mathematics but also physics and engineering). Using the same mathematical techniques, we can prove a similar global convergence result for the PINN neural network approximators. Both proofs require analyzing a kernel function in the limit ODE governing the evolution of the limit neural network approximator. A key technical challenge is that the kernel function, which is a composition of the PDE operator and the neural tangent kernel (NTK) operator, lacks a spectral gap, therefore requiring a careful analysis of its properties.

Key words. Neural network, partial differential equation, neural tangent kernel, machine learning, gradient flow

MSC codes. 65N75

1. Introduction.

Deep learning methods have become widely used for solving high-dimensional PDEs in financial engineering. Similar deep learning methods have also been used to model physics data governed by PDEs. Although low-dimensional PDEs can be efficiently solved with existing numerical techniques, such as finite difference methods, high-dimensional PDEs are computationally intractable due to the curse of dimensionality. An alternative approach that has been widely employed is to approximate the PDE solution with a neural network and then train the neural network with stochastic gradient descent to satisfy the PDE and its boundary conditions – e.g., the deep Galerkin method (DGM) in [34]. The DGM algorithm – as well as many subsequent algorithms which have further developed the algorithm – has been applied to a wide range of high-dimensional PDEs in financial mathematics. A similar method – physics-informed neural networks (PINNs) in [29] – was developed to model physics data by training a neural network to both satisfy the corresponding governing PDE and match a sparse set of experimental observations. Subsequently, PINNs methods have also been used for a variety of applications in financial mathematics.

There are now hundreds of articles applying deep learning methods to PDE applications in mathematical finance. Examples include [3], [6], [4], [5], [30], [15], [16], [11], [2], [17], [13], [32], and [7]. There are many other important papers which have studied

*Mathematical Institute, University of Oxford (cohens@maths.ox.ac.uk, jiangd@maths.ox.ac.uk, sirignano@maths.ox.ac.uk).

the application of deep learning to solving PDEs in financial engineering, including mathematical analysis of neural networks in PDE applications. A critical gap in the mathematical theory for such methods is a proof that a neural network *trained with gradient descent* to minimize the PDE residual will converge to the solution of the PDE. In this paper, we develop a convergence analysis for DGM and PINNs methods, proving that they converge to the solution of a class of linear PDEs when trained with gradient descent. Our mathematical techniques also provide a foundation to study other classes of PDEs in financial mathematics.

Both the DGM and PINNs methods share a common feature of training a neural network with (stochastic) gradient descent to satisfy an objective function with a PDE. The neural network is trained (with gradient descent) to satisfy the PDE operator in the interior of the domain (i.e., to minimize the PDE residual) and to satisfy the boundary conditions. Numerous other articles in the literature have also explored solving PDEs with neural networks (see [1] for an overview). Solving PDEs with neural network approximators is a natural idea that has been considered in different forms for decades, for instance, [22], [21], [25] and [31]. These papers propose to use neural networks to solve differential equations by estimating neural network solutions on an a priori fixed mesh.

Due to the universal approximation properties of neural networks [18], it is clear that *there exists* a neural network which can approximate the solution to a given PDE (interpreting the PDE solution as a function in the Sobolev space $\mathcal{H}^2$). Universal approximation results have been proven for PINNs in [27] and [9]. Recently, important results were proven by [33] and [10], showing that the global minimizer of PINNs with Hölder and ridge regularization will converge to the solution of the PDE as the scale of the network goes to infinity. However, these results prove that a neural network *exists* which is an arbitrarily-accurate approximation to the PDE solution; they do not prove that a neural network *trained with gradient descent* will converge to the PDE solution. In particular, in the DGM algorithm, the PDE solution is *a priori* unknown. The neural network is therefore trained to satisfy the PDE operator in the interior of the domain and the boundary conditions on the domain boundary. This is a highly non-convex objective function; therefore, it is unclear whether the neural network (trained with gradient descent) will converge to the PDE solution.

Convergence analysis for the optimization of neural network approximators to PDEs must address several mathematical challenges. First, the neural network is non-convex in its parameters, which is further exacerbated by applying a PDE operator to the neural network in the objective function. Consequently, as the number of hidden units $\to \infty$, the standard neural tangent kernel (NTK) does not arise [19]. Instead, the kernel function involves the PDE operator, requiring the development of new mathematical analysis. Finally, as is also true for the standard NTK setting, the kernel lacks a spectral gap, which makes analysis of infinite-dimensional systems (such as approximators to PDEs) challenging. Our proof leverages a careful analysis of the eigendecomposition of the limit ODE and its kernel function.

For both the DGM and PINN algorithms, we prove that, as the number of hidden units in the single-layer network goes to infinity (i.e., in the "wide network limit"), the trained neural network converges to the solution of an infinite-dimensional linear ODE. The PDE residual of the limiting approximator converges to zero as the training time $\to \infty$. Under mild assumptions, this convergence also implies that the neural network approximator converges to the solution of the PDE. [38] prove that, under some assumptions, as the number of neurons goes to infinity, the training process of PINNs will converge to a process characterized by a kernel matrix. However, [38] does

not prove global convergence of the neural network approximator.

Our paper provides a rigorous mathematical analysis of the DGM and PINN training process for solving PDEs with neural networks. In summary, the key contributions of this paper are:

1. We prove that as the number of hidden units in the neural network $\to \infty$ (i.e., in the "wide network limit), the training process of the neural approximator trained to minimize the PDE residual converges to an infinite-dimensional linear ODE characterized by a kernel function.
2. The kernel function is different than the standard NTK kernel and involves the PDE operator.
3. We prove that even though the kernel is only positive semi-definite and there is no spectral gap, the objective function (i.e., the PDE residual of the wide-limit neural network) converges to zero as the training time $t \to \infty$. This result establishes global convergence. Furthermore, under an additional mild assumption, the wide-limit neural network converges to the PDE solution.

For our analysis of PINNs, we study a version of the PINNs algorithm where spatial points are randomly sampled at each training iteration according to a fixed probability measure μ. This leads to a term in the objective function which is an expectation, with respect to the probability measure μ, of the PDE residual of the neural network approximator over the entire domain.

Finally, we review some of the relevant literature on the mathematical analysis of neural network methods for PDEs. In [24], a gradient descent training algorithm for neural net approximations for PDEs is considered, under different assumptions than in our paper and with different mathematical techniques. In particular, for a particular choice of activation functions, assuming non-degeneracy of the Gram matrix associated with the PDE kernel at finitely many (randomly chosen) points, they argue that the gradient descent method will converge as training time $\to \infty$ to a regularized approximation of the PDE. Under the assumption that the PDE source term has bounded Rademacher complexity, the resulting sequence of approximations will converge. However, it is not possible to interchange these two limiting steps (the number of points and number of training steps) using their analysis. Our approach instead proves that the trained neural network converges to an infinite-dimensional evolution equation and then directly proves convergence of the limit network to the PDE solution as the training time $\to \infty$, using eigenfunction decomposition analysis. We do not assume that the eigenvalues of the corresponding NTK kernel are uniformly lower bounded (which is not true as the number of hidden units in the network $\to \infty$) and we do not make assumptions on the Rademacher complexity of the PDE.

A theoretical analysis for solving PDEs in Reproducing Kernel Hilbert Spaces (RKHS) with applications to neural networks is given in [23]. Their analysis requires the assumption that the PDE operator has the same eigenfunctions as the kernel, which is not satisfied by the kernel induced by many common activation functions used in deep learning. As an example, the eigenfunctions of the kernel for ReLU, Sigmoid, and Tanh activation functions are not the eigenfunctions of a simple heat equation (typically sine functions). Therefore, we cannot assume that the neural network kernel eigenfunctions coincide with the PDE operator eigenfunctions. Our proof does not require this assumption and therefore our convergence theorem applies to a broad class of neural network models.

Our paper is organized as follows. In Section 2, we introduce the class of PDEs that will be considered. Section 3 describes the neural network training algorithm for solving PDEs and then proves that the neural network approximator converges to the

limit ODE as the number of hidden units $\to \infty$, and Section 4 presents the analysis on the kernel generated by the PDE and the activation function . Section 5 studies the properties of the kernel function that characterizes the limit ODE. Then, we prove that the PDE residual converges to zero as the training time $\to \infty$. Then, it is proven that – with an additional mild assumption on the PDE – the wide-limit neural network also converges to the PDE solution. In Section 6, we prove global convergence for the PINN algorithm. Lemmas, corollaries, and theorems are presented in the main part of the paper. All mathematical proofs are in the Appendix A.

2. Mathematical Framework. We will study the convergence of neural network algorithms – such as DGM and PINNs – for solving PDEs. In particular, we will analyze the convergence of such algorithms for the following class of second-order linear PDEs with Dirichlet boundary conditions:

$$\begin{cases} \mathcal{A}v &= h, \quad \text{in } \Omega \\ v &= f, \quad \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where $\Omega \in \mathbb{R}^d$ is a compact set with a smooth boundary. We will study strong Sobolev solutions to the PDE (2.1); that is, we are interested in solutions $u \in \mathcal{H}^2$, where for a finite measure μ on Ω,

$$\mathcal{H}^p = \left\{ f \in L^2(\Omega, \mu) : \|f\|_{\mathcal{H}^p} := \Big(\sum_{|\alpha| \le p} \|D_\alpha f\|_{L^2} \Big) < \infty \right\}, \tag{2.2}$$

where Du is the weak derivative of u (see [12]), where (2.1) is taken to hold μ-almost everywhere on the interior of our domain, and in a trace sense on the boundary. We assume μ is equivalent to Lebesgue measure, and the logarithm of its Radon–Nikodym derivative is bounded (which ensures it generates the same $\mathcal{H}^2$ space as Lebesgue measure). For simplicity, in what follows we will simply say 'solution' for 'strong solution'; we note that if a strong solution v is also known to be $\mathcal{C}^2$, then it is also a classical solution. For notational convenience, we will write $\mathcal{H}^2_{(0)} = \mathcal{H}^2 \cap \mathcal{H}^1_0$, representing the $\mathcal{H}^2$ functions with zero value (in trace sense) on the boundary, equipped with the $\mathcal{H}^2$ norm.

We make the following (standard) assumptions on our problem:

ASSUMPTION 2.1 (Smoothness of the boundary Ω). *The boundary $\partial\Omega$ is $C^{3,\alpha}$ for some $\alpha \in (0,1)$; i.e., three times continuously differentiable with α-Hölder continuous derivatives of order 3.*

ASSUMPTION 2.2 (Auxiliary function η). *There exists a (known) function $\eta \in C^3_b(\mathbb{R}^n)$, which satisfies $\eta > 0$ in Ω, and $\eta = 0$ on $\partial\Omega$. Furthermore, its first order derivative does not vanish at the boundary (that is, for $x \in \partial\Omega$ and $\mathbf{n}_x$ an outward unit normal vector at x, we have $\nabla\eta(x) \cdot \mathbf{n}_x \neq 0$).*

Remark 2.3. If the boundary of our domain is the unit sphere, a natural choice of η is given by $\eta(x) = 1 - \|x\|^2$. More generally, a mollification of the distance to the boundary of the domain can usually be used, if this is easy to compute. In other cases the specification of the domain can suggest choices of η, for example, if we are given a domain in the form $\Omega = \{\eta(x) > 0\}$ for some function η, then this provides a natural choice of η provided $\nabla\eta \neq 0$ when $\eta = 0$.

ASSUMPTION 2.4 (Interpolation of the boundary condition function). *There exists a (known) function $\bar{f} \in \mathcal{H}^2$ such that $\bar{f}|_{\partial\Omega} = f$. In the rest of this paper, we identify f with its extension $\bar{f}$ defined on $\overline{\Omega}$ for notational simplicity.*

We can reformulate the PDE as

$$\begin{cases} \mathcal{A}u &= g, \quad \text{in } \Omega \\ u &= 0, \quad \text{on } \partial\Omega, \end{cases} \tag{2.3}$$

where $u := v - f$ and $g := h - \mathcal{A}f$. Finally, we assume that the PDE operator satisfies a certain type of Lipschitz condition:

ASSUMPTION 2.5 (Lipschitz condition). *There exists a constant $k > 0$ such that for any $f_1, f_2 \in \mathcal{H}^2$ and any $x \in \Omega$, the linear operator $\mathcal{A}$ satisfies*

$$|\mathcal{A}f_1(x) - \mathcal{A}f_2(x)| \leq k\Big[\sum_{0\leq|\alpha|\leq 2} |D_\alpha f_1(x) - D_\alpha f_2(x)|\Big]. \tag{2.4}$$

3. Deep Learning Algorithms for Solving PDEs. Deep learning PDE algorithms – such as DGM and PINNs – train a neural network approximator to satisfy the PDE and its boundary conditions using either gradient descent or stochastic gradient descent.

Consider the following single-layer neural network, with N hidden units equipped with activation function σ, S^N:

$$S^N(x;\theta^N) = \frac{1}{N^\beta}\sum_{i=1}^{N} c^i\sigma(w^i x + b^i), \tag{3.1}$$

where $\frac{1}{N^\beta}$ is a normalization factor and $\frac{1}{2} < \beta < 1$. A normalization factor in front of the summation $\sum_{i=1}^N c^i\sigma(w^i x + b^i)$ is necessary to avoid the neural network output diverging as $N \to \infty$. The choice of $\frac{1}{N^\beta}$ with $\beta \in (\frac{1}{2}, 1)$ as a normalization, combined with the learning rate, will lead to an NTK-type limit for the neural network as $N \to \infty$. We train a neural network Q^N to approximate the solution u to the PDE where

$$Q^N(x;\theta^N) := \eta(x)S^N(x;\theta^N) = \eta(x)\cdot\frac{1}{N^\beta}\sum_{i=1}^{N} c^i\sigma(w^i x + b^i). \tag{3.2}$$

Here $\eta(x)$ is a fixed function which vanishes on the boundary $x \in \partial\Omega$; therefore Q^N automatically satisfies the boundary conditions of the PDE (2.3). This method, which was first introduced by [26], simplifies the training of the neural network model. The parameters $\theta^N = (c^i, w^i, b^i)_{i=1}^N$ must be trained using gradient descent to satisfy the PDE in the interior of the domain. Specifically, we will minimize the PDE residual error for the neural network by minimizing the following objective function:

$$J(\theta^N) = \|\mathcal{A}Q^N - g\|^2_{L^2(\mu)} := \int_\Omega [\mathcal{A}Q^N(x;\theta^N) - g(x)]^2 d\mu(x), \tag{3.3}$$

where μ is a sampling measure (satisfying the regularity assumptions stated after (2.2) above). If the residual term $\mathcal{A}Q^N(x;\theta^N) - g(x)$ equals zero for all $x \in \Omega$, then $Q^N = u$ is the solution of the PDE. We will minimize the objective function (3.3) using continuous-time gradient descent with clipping. The parameter updates in continuous-time gradient descent satisfy the system of ODEs in equation (3.5), which can be numerically solved using ODE numerical methods (e.g., Runge–Kutta

or Euler). If an Euler scheme is used, it becomes identical to standard discrete-time gradient descent. The gradient of (3.3) is:

$$\nabla_\theta J(\theta^N) = \int_\Omega [\mathcal{A}Q^N(x;\theta^N) - g(x)]\nabla_\theta \mathcal{A}Q^N(x;\theta^N)d\mu(x). \tag{3.4}$$

Gradient clipping is widely used in deep learning, see for instance [39], [28] and chapters 10 and 11 of [14]. The continuous-time gradient descent training with clipping is given by:

$$\frac{d\theta_t^N}{dt} = -\alpha^N G^N(\theta_t^N), \tag{3.5}$$

where the learning rate is $\alpha^N = N^{2\beta-1}$ and

$$G^N(\theta_t^N) = -\int_\Omega \psi^N(\mathcal{A}Q^N(x;\theta_t^N) - g(x))\Phi^N(\nabla_\theta \mathcal{A}Q^N(x;\theta_t^N))d\mu(x). \tag{3.6}$$

Here Φ^N is a vector function that applies elementwise clipping with the function ϕ^N. For the explicit definition of clipping functions being applied here, see Definition 3.3, and Assumption 3.4. For each entry of vector $\nabla_\theta \mathcal{A}Q^N$, we clip its value with the scalar function ϕ^N.

In practice, (3.5) can be approximated by discretizing in time and, at each time step, generating Monte Carlo samples from the measure μ to approximate the integral, which is highly computationally efficient even for high-dimensional PDEs. This is also equivalent to the stochastic gradient descent version of (3.5). Although not investigated in this paper, standard methods can be used to prove that stochastic gradient – using the correct learning rate – will converge to the continuous-time gradient flow (3.5); for example, weak convergence analysis such as in [36] and [35] could be used.

Let $Q_t^N(x) = Q^N(x;\theta_t^N)$ be the neural network at training time t with N hidden units. We will analyze the trained neural network Q_t^N as the number of hidden units $N \to \infty$ and the training time $t \to \infty$. First, we prove that the trained neural network Q_t^N will converge to the solution of an infinite-dimensional ODE as $N \to \infty$. That is, in the "wide limit" where the number of hidden units $\to \infty$, Q_t^N converges to the solution of an ODE. Then, we prove that the wide-limit neural network converges to the global minimizer of the objective function (with zero PDE residual) as the training time $t \to \infty$. Under additional mild assumptions, this global minimizer is also a solution to the PDE. These convergence results can also be proven for the PINNs algorithm for solving PDEs; see Section 6.

Our convergence results will be proven under the following assumptions on the neural network architecture:

Assumption 3.1 (Activation function). *The activation function $\sigma \in C_b^4(\mathbb{R})$ is non-constant.*

While Assumption 3.1 is potentially restrictive, it is satisfied by commonly used activation functions such as Sigmoid and Tanh functions. In the remainder of this paper, we use both $\sigma(x;w,b)$ and $\sigma_{w,b}(x)$ to denote the activation function.

Assumption 3.2 (Neural network initialization). *The initialization of the parameters θ_0^N, for all $i \in \{1,2,...,N\}$, satisfies:*

- *The parameters c_0^i, w_0^i, b_0^i are i.i.d. random variables.*

- *The random variables c_0^i are bounded, $|c_0^i| < K_0$, and $\mathbb{E}[c_0^i] = 0$.*
- *The distribution of the random variables w_0^i, b_0^i has full support. That is, for any open set $D \subseteq \mathbb{R}^{n+1}$, we have $\mathbb{P}((w_0^i, b_0^i) \in D) > 0$.*
- *The moments $\mathbb{E}[|(w_0^i)_k|^3]$ and $\mathbb{E}[|b_0^i|]$ are bounded where $(w_0^i)_k$ is the k-element of w_0^i.*

DEFINITION 3.3 (Smooth clipping function). *A function class $\{h^N\}_{N\in\mathbb{N}^+}$ forms a family of smooth clipping functions with parameter $\gamma > 0$ if for any $N \in \mathbb{N}^+$*

- *$h^N \in \mathbb{C}_b^2(\mathbb{R})$ is increasing on $\mathbb{R}$.*
- *$|h^N|$ is bounded by $2N^\gamma$.*
- *$h^N(x) = x$ for $x \in [-N^\gamma, N^\gamma]$.*
- *$|(h^N)'| \leq 1$ on $\mathbb{R}$.*

ASSUMPTION 3.4. *Functions $\{\psi^N\}_{N\in\mathbb{N}^+}$ and $\{\phi^N\}_{N\in\mathbb{N}^+}$ are families of smooth clipping functions with parameter δ and $\epsilon - \beta$ where $\epsilon > \delta > 0$, $\beta \in (\frac{1}{2}, 1)$ and $\epsilon + \delta < \frac{1-\beta}{2}$.*

LEMMA 3.5. *There exists a constant k independent of N such that the change of each component of c^i, w^i, b^i from its initial condition (e.g. $|c_t^i - c_0^i|$) is bounded by $ktN^{2\beta-1+\delta+\epsilon-\beta} = ktN^{\beta+\delta+\epsilon-1}$.*

Proof. Notice that in (3.5), three terms α^N, $\psi^N(\mathcal{A}Q^N - g)$ and $\Phi^N(\nabla_\theta \mathcal{A}Q^N)$ are bounded by $N^{2\beta-1}$, N^δ and $N^{\epsilon-\beta}$ respectively. ∎

Remark 3.6. Gradient clipping is a method for preventing gradient explosion (i.e., gradient updates becoming very large/unbounded) and is a standard technique in deep learning. Assumption 3.4 and Lemma 3.5 are introduced to make sure that when $N \to \infty$, the truncation due to the clipping function vanishes while also guaranteeing stability (appropriately bounded parameter updates) during training. From a mathematical perspective, the gradient clipping allows us to prove certain uniform bounds which are useful for the convergence proof that the neural network training converges to the limit ODE as the number of hidden units $N \to \infty$. Since the clipping function vanishes in the limit $N \to \infty$, it does not appear in the limit ODE and therefore it also does not appear in the convergence analysis for when the training time $t \to \infty$. From a numerical perspective, gradient clipping is a standard method in deep learning to reduce instability in training (due to, for example, exploding gradients) and improve training convergence.

3.1. Convergence of the trained neural network as the number of hidden units $N \to \infty$. In this section, we analyze the evolution of the neural network Q_t^N as it is trained to minimize the PDE residual in the objective function $J(\theta^N)$. We can prove that, as the number of hidden units $N \to \infty$, the neural network $Q_t^N(y)$ will converges to the solution $Q_t(y)$ of an infinite-dimensonal linear ODE. By the chain rule, the dynamics of Q^N satisfy

$$\begin{aligned}\frac{dQ_t^N}{dt}(y) &= \nabla_\theta Q_t^N(y) \cdot \frac{d\theta^N}{dt} \\ &= -\int_\Omega \psi^N(\mathcal{A}Q_t^N(x) - g(x))\left[\alpha^N \Phi^N(\nabla_\theta \mathcal{A}Q_t^N(x))\right] \cdot \nabla_\theta Q_t^N(y) d\mu(x).\end{aligned} \tag{3.7}$$

We are interested in studying the limit of the dynamics of the neural network Q_t^N as $N \to \infty$. Specifically, we will prove that Q_t^N will converge to Q_t as $N \to \infty$ where Q_t

satisfies

$$\frac{dQ_t}{dt}(y) = -\int_\Omega [\mathcal{A}Q_t(x) - g(x)]U(x,y)d\mu(x), \quad Q_0 = 0, \tag{3.8}$$

where the function U is

$$U(x,y) := \mathbb{E}_{c,w,b}\Big[\nabla_{c,w,b}\mathcal{A}[\eta(x)c\sigma(x;w,b)] \cdot \nabla_{c,w,b}[\eta(y)c\sigma(y;w,b)]\Big], \tag{3.9}$$

where the random variable (c,w,b) has the same distribution as (c_0^i, w_0^i, b_0^i).

The ODE (3.8) is an infinite-dimensional linear ODE governing the evolution of the wide-limit neural network (i.e., a neural network with an "infinite" number of hidden units) during training. The right-hand side (RHS) of the ODE involves integral over the PDE residual $[\mathcal{A}Q_t(x) - g(x)]$ weighted by a kernel $U(x,y)$. It is important to notice that the kernel $U(x,y)$ is not the standard NTK kernel: it involves the PDE operator $\mathcal{A}$, which significantly complicates its analysis.

One of the consequences of the presence of the PDE operator $\mathcal{A}$ in the kernel $U(x,y)$ is that $U(x,y)$ is asymmetric. This is a key difference from the standard NTK kernel, which is symmetric.

Define the integral operator $\mathcal{U} : L^2 \to \mathcal{H}^2_{(0)} \subset L^2$ by

$$\mathcal{U}f := \int_\Omega f(x)U(x,y)d\mu(x). \tag{3.10}$$

Note that it is straightforward to check that $U(x,y) = 0$ for $y \in \partial\Omega$ and that U is C_b^2 with respect to y, which ensures that $\mathcal{U}f$ takes values in $\mathcal{H}^2_{(0)}$. (This is the main motivation for Assumption 3.1.) Using this notation, the limit ODE (3.8) can be rewritten as a linear equation in $\mathcal{H}^2_{(0)}$:

$$\frac{dQ_t}{dt} = -\mathcal{U}[\mathcal{A}Q_t - g], \quad Q_0 = 0. \tag{3.11}$$

LEMMA 3.7. *The ODE* (3.11) *admits a unique solution in* $\mathcal{H}^2_{(0)}$.

Proof. By Assumptions 2.5, 3.1 and 3.2, the operator $\mathcal{U}$ is Lipschitz in $\mathcal{H}^2$ norm. Therefore it admits a unique solution in $\mathcal{H}^2$. It is easy to verify that, as its initial condition is in $\mathcal{H}^2_{(0)}$, and $\mathcal{U}$ has codomain $\mathcal{H}^2_{(0)}$, the solution lives in the Hilbert subspace $\mathcal{H}^2_{(0)}$. ∎

Now we present one of this paper's main results. The trajectory of Q_t^N during training, in the limit $N \to \infty$, can be characterized by the wide-limit network Q_t which satisfies the infinite-dimensional ODE (3.11).

THEOREM 3.8. *For any* $t \geq 0$, *the neural network* Q_t^N *converges to* Q_t *in* $\mathcal{H}^2$*:*

$$\lim_{N\to\infty} \mathbb{E}[\|Q_t^N - Q_t\|_{\mathcal{H}^2}] = 0. \tag{3.12}$$

The proof of this theorem is presented in the Appendix.

4. Analysis of the kernel function. In order to prove global convergence as $t \to \infty$ for the limit ODE (3.11), we first must prove some key properties for the integral operator. Specifically, we will study the properties of the operator $\mathcal{S} = \mathcal{A}\mathcal{U}$.

DEFINITION 4.1 (Operator $\mathcal{S}$). *The operator $\mathcal{S} : L^2 \to L^2$ is defined by*

$$\mathcal{S}f := \mathcal{A}\mathcal{U}f = \mathcal{A}\Big(\int_\Omega f(x)U(x,\cdot)d\mu(x)\Big) \tag{4.1}$$

DEFINITION 4.2 (Kernel S). *The kernel S is defined by*

$$S(x,y) := \mathbb{E}_{c,w,b}\Big[\nabla_{c,w,b}\mathcal{A}[\eta(x)c\sigma(x;w,b)] \cdot \nabla_{c,w,b}\mathcal{A}[\eta(y)c\sigma(y;w,b)]\Big]. \tag{4.2}$$

By symmetry of second derivatives (Clairaut–Schwarz–Young theorem) we know that $S(x,\cdot) := \mathcal{A}U(x,\cdot)$ and hence $\mathcal{S}f = \int_\Omega f(x)S(x,\cdot)d\mu(x)$.

While $U(x,y)$ is asymmetric, $S(x,y)$ is symmetric. The symmetric kernel S depends upon the interaction of PDE operator $\mathcal{A}$ applied to the activation function σ. It will next be proven (Lemma 4.6) that the operator $\mathcal{S}$ is discriminatory in the image set of $\mathcal{A}$. This is an important property that will later be leveraged in the global convergence proof.

LEMMA 4.3. *The kernel S is uniformly bounded. That is, there exists a constant $k > 0$ such that for any $(x,y) \in \Omega^2$ we know $|S(x,y)| \le k$.*

Proof. By definition

$$\begin{aligned} S(x,y) = \mathbb{E}\Big[&\mathcal{A}[\eta(x)\sigma_{w,b}(x)]\mathcal{A}[\eta(y)\sigma_{w,b}(y)] + \sum_{i=1}^{d}\mathcal{A}[c\eta(x)x_i\sigma'_{w,b}(x)]\mathcal{A}[c\eta(y)y_i\sigma'_{w,b}(y)] \\ &+ \mathcal{A}[c\eta(x)\sigma'_{w,b}(x)]\mathcal{A}[c\eta(y)\sigma'_{w,b}(y)]\Big]. \end{aligned} \tag{4.3}$$

By the fact that $\mathcal{A}$ is Lipschitz, and that η, $\sigma_{w,b}$ and their partial derivatives are all bounded, there exists constant $k_1 > 0$, $k_2 > 0$ such that

$$|\mathcal{A}[\eta(x)\sigma_{w,b}(x)]| \le k\sum_\alpha |D_\alpha\eta(x)\sigma_{w,b}(x)| \le k_1 \sum_{1\le i,j\le d} |w_i| + |w_iw_j| + 1. \tag{4.4}$$

Similarly, since c is bounded and Ω is bounded

$$\begin{aligned} |\mathcal{A}[c\eta(x)\sigma'_{w,b}(x)]| &\le k_1 \sum_{1\le i,j\le d} |w_i| + |w_iw_j| + 1 \\ |\mathcal{A}[c\eta(x)x_i\sigma'_{w,b}(x)]| &\le k_1 \sum_{1\le i,j\le d} |w_i| + |w_iw_j| + 1. \end{aligned} \tag{4.5}$$

Therefore,

$$|S(x,y)| \le \mathbb{E}_{c,w,b}\big[(d+2)k_1^2(\sum_{1\le i,j\le d} |w_i| + |w_iw_j| + 1)^2\big] \le k_2. \tag{4.6}$$

∎

LEMMA 4.4. *The integral operator $\mathcal{S}$ is Hilbert–Schmidt. In addition, $\mathcal{S}$ is self-adjoint and positive semi-definite.*

Proof. Since $S(x,y)$ is uniformly bounded, we have $\iint_{\Omega^2} |S(x,y)|^2 d\mu(x)d\mu(y) < \infty$. Therefore $\mathcal{S}$ is Hilbert–Schmidt. And since $S(x,y)$ is symmetric, the operator $\mathcal{S}$

is self-adjoint with respect to the L^2 inner product. Now it remains to prove that $\mathcal{S}$ is positive semi-definite. For $f \in L^2$

$$\begin{aligned}\langle f, \mathcal{S}f\rangle &= \iint_{\Omega^2} f(x)S(x,y)f(y)d\mu(x)d\mu(y) \\ &= \iint f(x)\mathbb{E}_{c,w,b}\big[\nabla_{c,w,b}\mathcal{A}[\eta(x)c\sigma(x;w,b)]\cdot\nabla_{c,w,b}\mathcal{A}[\eta(y)c\sigma(y;w,b)]\big]f(y)d\mu(x)d\mu(y)\end{aligned} \tag{4.7}$$

By Tonelli's theorem, swapping the order of expectation and the integral gives

$$\begin{aligned}&\langle f, \mathcal{S}f\rangle \\ &= \mathbb{E}\bigg[\iint_{\Omega^2} f(x)\nabla_{c,w,b}\mathcal{A}[\eta(x)c\sigma(x;w,b)]\cdot\nabla_{c,w,b}\mathcal{A}[\eta(y)c\sigma(y;w,b)]f(y)d\mu(x)d\mu(y)\bigg] \\ &\geq \mathbb{E}_{c,w,b}\bigg[\iint_{\Omega^2} f(x)\nabla_{c}\mathcal{A}[\eta(x)c\sigma(x;w,b)]\cdot\nabla_{c}\mathcal{A}[\eta(y)c\sigma(y;w,b)]f(y)d\mu(x)d\mu(y)\bigg] \\ &= \mathbb{E}_{c,w,b}\bigg[\iint_{\Omega^2} f(x)\mathcal{A}[\eta(x)\sigma(x;w,b)]\mathcal{A}[\eta(y)\sigma(y;w,b)]f(y)d\mu(x)d\mu(y)\bigg] \\ &= \mathbb{E}_{c,w,b}\bigg[\Big(\int_{\Omega} f(x)\mathcal{A}[\eta(x)\sigma(x;w,b)]d\mu(x)\Big)^2\bigg] \geq 0,\end{aligned} \tag{4.8}$$

which concludes the proof. ∎

LEMMA 4.5 (Spectral decomposition). *The integral operator $\mathcal{S}$ is compact, in particular, there exists an orthogonal basis of L^2, $\{\varepsilon_i\}_{i\in\mathbb{N}^+}\cup\{\nu_i\}_{i\in\mathbb{N}^+}$ such that*

$$\mathcal{S}\varepsilon_i = \lambda_i\varepsilon_i, \quad \mathcal{S}\nu_i = 0, \tag{4.9}$$

where $\lambda_1 \geq \lambda_2 \geq ... > 0$.

Proof. As $\mathcal{S}$ is a Hilbert–Schmidt integral operator, $\mathcal{S}$ is compact. Since $\mathcal{S}$ is self-adjoint, the spectral theorem applies. From Lemma 4.4, we see that $\mathcal{S}$ is positive semi-definite. Thus, its eigenvalues are real, non-negative, and concentrate only at zero. We use $\{\nu_i\}$ to represent the eigenfunctions of the zero eigenvalue, and $\{\varepsilon_i\}$ for eigenfunctions of positive eigenvalues. ∎

LEMMA 4.6 (Projection on $\ker(\mathcal{S})$). *For $h \in L^2$, if $\mathcal{S}h = 0$ then $\langle h, \mathcal{A}f\rangle = 0$ for any $f \in \mathcal{H}^2_{(0)}$.*

We recall without proof the following technical result.

LEMMA 4.7 (Lemma 5 in [8]). *Given Assumptions 2.1 and 2.2:*

1. *The set of functions $C^3(\overline{\Omega})\cap C_0(\overline{\Omega})$ is dense in $\mathcal{H}^2_{(0)} = \mathcal{H}^2\cap\mathcal{H}^1_0$ (under the $\mathcal{H}^2$ topology).*
2. *For any function $u \in C^3(\overline{\Omega})\cap C_0(\overline{\Omega})$, the function $\tilde{u} = u/\eta$ is in $C^2_b(\Omega) \subset \mathcal{H}^2$.*

We now proceed to the proof of Lemma 4.6.

Proof of Lemma 4.6. By (4.8), we have

$$\langle h, \mathcal{S}h\rangle = \mathbb{E}_{c,w,b}\bigg[\bigg(\int_{\Omega} h(x)\mathcal{A}[\eta(x)\sigma(x;w,b)]d\mu(x)\bigg)^2\bigg] = 0. \tag{4.10}$$

Therefore

$$\int_\Omega h(x)\mathcal{A}[\eta(x)\sigma(x;w,b)]d\mu(x) = 0 \tag{4.11}$$

for any (c, w, b) by the continuity of the objective with respect to parameters c, w, b.

Since μ is a finite measure, from Theorem 4 in [18] we have that the linear span of $\{\sigma(w \cdot x + b)\}_{w,b\in\mathbb{R}}$ is dense in $\mathcal{H}^2$. For a general $f \in C^3(\bar{\Omega}) \cap C_0(\bar{\Omega})$, by Lemma 4.7(2), we can approximate the function f/η within the linear span of $\{\sigma(w \cdot x + b)\}_{w,b\in\mathbb{R}}$. Multiplying by η, by Lemma 4.7(1) it follows that the function class $\{\eta(x)\sigma(x; w, b)\}$ is dense in the function space $\mathcal{H}^2_{(0)}(\Omega)$, which consists of $\mathcal{H}^2$ function with boundary value zero. For any $f \in \mathcal{H}^2_{(0)}$, there exists a function sequence $\{F_N : \eta\sum_{i=1}^N c_i\sigma_{w,b}\}_{N\geq 1}$ such that

$$\lim_{N\to\infty} \|F_N - f\|_{\mathcal{H}^2} = 0. \tag{4.12}$$

Therefore,

$$\langle h, \mathcal{A}f\rangle = \int_\Omega h(x)\mathcal{A}f d\mu(x) = \lim_{N\to\infty}\langle h, \mathcal{A}F^N\rangle = 0. \tag{4.13}$$ ∎

Remark 4.8. If u is a solution to the PDE (2.3), Lemma 4.6 implies that for any $t \geq 0$, the residual of our approximator $\mathcal{A}Q_t - g = \mathcal{A}[Q_t - u]$ has zero projection on the eigenfunction family $\{\nu_i\}$ since $\mathcal{S}\nu_i = 0$.

COROLLARY 4.9. *Assuming the PDE* (2.3) *admits a solution* $u \in \mathcal{H}^2_{(0)}$*, any stationary point* $Q^* \in \mathcal{H}^2_{(0)}$ *of the limit ODE* (3.8) *is a solution of the PDE* (2.3)*. That is, any stationary point of the limit training algorithm* (3.8) *is a global minimizer and a solution of* (2.3)*.*

Proof. Suppose that Q^* is a stationary point. Then we have

$$\mathcal{U}[\mathcal{A}Q^* - g] = 0. \tag{4.14}$$

It follows that

$$\mathcal{S}\mathcal{A}[Q^* - u] = \mathcal{S}[\mathcal{A}Q^* - g] = \mathcal{A}\mathcal{U}[\mathcal{A}Q^* - g] = 0. \tag{4.15}$$

By Lemma 4.6, the inner product term $\langle\mathcal{A}[Q^* - u], \mathcal{A}f\rangle = 0$ for any $f \in \mathcal{H}^2_{(0)}$. Therefore, by taking $f = Q^* - u$, we have $\|\mathcal{A}Q^* - g\|_2^2 = \langle\mathcal{A}[Q^* - u], \mathcal{A}[Q^* - u]\rangle = 0$. At the same time, since $Q^* \in \mathcal{H}^2_{(0)}$ satisfies the zero boundary condition, we conclude that Q^* is a solution of the PDE. ∎

Remark 4.10. The existence of solutions is needed in this result. For example, consider the trivial case where $\mathcal{A}u \equiv 0$. Then the kernel satisfies $\mathcal{U} \equiv 0$, so $\frac{d}{dt}Q_t = 0$, and we must have a fixed point. However, if $g \neq 0$, then $Q_t = Q_0$ does not solve the PDE. Nevertheless, no uniqueness of solutions is needed, and consequently we cannot expect that $Q^* = u$.

5. Global convergence of the limit ODE as $t \to \infty$. By analyzing the PDE residual term's projection on the eigenfunctions of $\mathcal{S}$, we can prove that the PDE residual (which is the objective function that is being minimized) converges to zero as the training time $t \to \infty$.

Applying $\mathcal{A}$ with respect to y on both sides of the limit ODE (3.11) yields

$$\frac{d\mathcal{A}Q_t}{dt} = \frac{d[\mathcal{A}Q_t - g]}{dt} = -\mathcal{S}[\mathcal{A}Q_t - g]. \tag{5.1}$$

THEOREM 5.1 (Convergence of PDE residual under DGM). *Assuming the PDE (2.3) admits a solution in $\mathcal{H}^2_{(0)}$, the PDE residual for the wide-limit neural network Q_t converges to zero:*

$$\lim_{t\to\infty} \|\mathcal{A}Q_t - g\|_{L^2(\mu)} = 0. \tag{5.2}$$

Proof. In (5.1), multiplying $\mathcal{A}Q_t - g$ on both sides and integrating with respect to $\mu(dy)$:

$$\frac{d\|\mathcal{A}[Q_t - u]\|_2^2}{dt} = -2\langle \mathcal{A}[Q_t - u], \mathcal{S}\mathcal{A}[Q_t - u]\rangle \leq 0 \tag{5.3}$$

with strict inequality unless $\|\mathcal{A}Q_t - g\|_2 = 0$, which corresponds to the PDE solution.

Consider $\mathcal{A}Q_t - g = \mathcal{A}[Q_t - u]$ projected on $\{\varepsilon_i\}_{i\in\mathbb{N}^+} \cup \{\nu_i\}_{i\in\mathbb{N}^+}$. By Lemma 4.6, we have

$$\mathcal{A}Q_t - g = \sum_i h_t^i \varepsilon_i + \sum_j 0\nu_j = \sum_i h_t^i \varepsilon_i. \tag{5.4}$$

Therefore $\|AQ_t - g\|_2^2 = \sum_i {h_t^i}^2$. Now consider its projection on each ε_i

$$\begin{aligned} \frac{d}{dt}\langle \mathcal{A}Q_t - g, \varepsilon_i\rangle &= \left\langle \frac{d\mathcal{A}Q_t - g}{dt}, \varepsilon_i \right\rangle = \langle -\mathcal{S}[\mathcal{A}Q_t - g], \varepsilon_i\rangle = \langle \mathcal{A}Q_t - g, -\mathcal{S}\varepsilon_i\rangle \\ &= -\lambda_i \langle \mathcal{A}Q_t - g, \varepsilon_i\rangle. \end{aligned} \tag{5.5}$$

Therefore

$$\frac{d}{dt}\langle \mathcal{A}Q_t - g, \varepsilon_i\rangle = \frac{d}{dt}[h_t^i] = -\lambda_i h_t^i. \tag{5.6}$$

Consequently $h_t^i = h_0^i e^{-\lambda_i t}$ whose absolute value decays exponentially, and $|h_t^i| \leq |h_0^i|$ for any $t \geq 0$ and any i.

Now, by the dominated convergence theorem

$$\lim_{t\to\infty} \|\mathcal{A}Q_t - g\|_2^2 = \lim_{t\to\infty} \sum_i |h_t^i|^2 = \sum_i \lim_{t\to\infty} |h_t^i|^2 = 0. \tag{5.7}$$

□

This result establishes global convergence of the (wide-limit) training algorithm since the objective function is the PDE residual.

Remark 5.2. Due to the lack of a spectral gap for the eigenvalues of the kernel, it is difficult to establish a convergence rate since some eigenfunction projections may be converging very slowly (if their corresponding eigenvalue is close to zero). As shown in our analysis, for an eigenfunction ϵ_i, we can establish an exponential convergence rate for eigenfunction projections of the PDE residual $\langle \epsilon_i, \mathcal{A}Q_t - g\rangle$ where Q_t is the neural network and $\mathcal{A}u = g$ is the PDE we are trying to solve. In particular, $|\langle \epsilon_i, \mathcal{A}Q_t - g\rangle| \leq C_i \exp(-\lambda_i t)$ where λ_i is the eigenvalue corresponding to the eigenfunction ϵ_i.

However, the convergence of the PDE residual to zero does not necessarily guarantee that $\|Q_t - u\|_{L^2(\mu)}$ as $t \to \infty$ where u is the solution of the PDE (2.3). We show that, under a mild additional assumption on the PDE operator $\mathcal{A}$, we can guarantee that Q_t converges to the solution of the PDE u.

ASSUMPTION 5.3 (Bounded inverse of $\mathcal{A}$). *The inverse of $\mathcal{A}$ is a bounded operator on $L^2 \to L^2$. That is, there exists a constant $k > 0$, such that for any $g \in L^2$, the PDE (2.3) has a unique solution $u \in \mathcal{H}^2_{(0)}$ satisfying*

$$\|u\|_2 \leq k\|g\|_{L^2(\mu)}.$$

Remark 5.4. Second-order uniformly elliptic PDEs naturally have this property. For reference, see Theorem 6, Chapter 6 of [12].

THEOREM 5.5 (Convergence of Q_t under DGM). *If $\mathcal{A}$ has a bounded inverse, Q_t converges to the solution u of the PDE:*

$$\lim_{t\to\infty} \|Q_t - u\|_{L^2(\mu)} = 0.$$

Proof. Writing $\mathcal{A}^{-1}$ for the inverse operator of $\mathcal{A}$, we have

$$\|Q_t - u\|_2 = \|\mathcal{A}^{-1}[\mathcal{A}[Q_t - u]]\|_2 \leq k\|\mathcal{A}[Q_t - u]\|_2 = k\|\mathcal{A}Q_t - g\|_2 \to 0. \tag{5.8}$$ ∎

6. Global Convergence of the PINN Algorithm. In many real-world engineering applications, the PDE solution can be observed at a sparse set of points in the interior of the domain Ω. We consider a PINN algorithm which trains a neural network model to predict the solution $u(x)$ for all $x \in \Omega$ by minimizing (via gradient descent) both the PDE residual sampled by μ on a random set of points in each epoch as well as the distance between the neural network and the observations at the finite set of sparse points. It is assumed that the boundary condition is known.

Let us denote these extra observation points in Ω as $\mathbf{x} := \{x_i\}_{i=1,2,\dots M}$ where the observations are $u(x_i) = u_i$ for $1 \leq i \leq M$. Define the measure $\mu_{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M} \delta_{x_i}$ and the kernel function

$$B(x, y) := \mathbb{E}_{c,w,b}\Big[\nabla_{c,w,b}[c\eta(x)\sigma_{w,b}(x)] \cdot \nabla_{c,w,b}[c\eta(y)\sigma_{w,b}(y)]\Big], \tag{6.1}$$

and the corresponding 'integral' operator $\mathcal{B} : L^2(\mu_{\mathbf{x}}) \equiv \mathbb{R}^M \to L^2(\mu)$ by

$$\mathcal{B}\mathbf{v}(y) := \frac{1}{M}\sum_{i=1}^{M} v_i B(x_i, y). \tag{6.2}$$

For any operator $\mathcal{C}$, we define $\bar{\mathcal{C}}$ to be the evaluation of $\mathcal{C}$ on each of our training points,

$$\bar{\mathcal{C}}v = \big[\mathcal{C}\mathbf{v}(x_1), \cdots, \mathcal{C}\mathbf{v}(x_M)\big]^\top \in \mathbb{R}^M,$$

in particular

$$\bar{\mathcal{B}}\mathbf{v} := \Big[\frac{1}{M}\sum_{i=1}^{M} v_i B(x_i, x_1), \cdots, \frac{1}{M}\sum_{i=1}^{M} v_i B(x_i, x_M)\Big]^\top, \tag{6.3}$$

and for generic $g \in L^2(\mu)$ and $\mathbf{v} \in \mathbb{R}^M$, direct calculation shows that

$$\begin{aligned}\langle \bar{\mathcal{U}} g, \mathbf{v} \rangle_{L^2(\mu_{\mathbf{x}})} &= \frac{1}{M} \sum_i (\mathcal{U} g)(x_i) v_i \\ &= \frac{1}{M} \sum_i \int \mathbb{E}\Big[v_i g(x) \nabla_{c,w,b} \mathcal{A}[c\eta(x)\sigma_{w,b}(x)] \cdot \nabla_{c,w,b}[c\eta(x_i)\sigma_{w,b}(x_i)] \Big] d\mu \\ &= \langle g, \mathcal{A}\mathcal{B}\mathbf{v} \rangle_{L^2(\mu)},\end{aligned} \tag{6.4}$$

in other words, $\bar{\mathcal{U}} : L^2(\mu) \to \mathbb{R}^M$ is the adjoint of $\mathcal{A}\mathcal{B} : \mathbb{R}^M \to L^2(\mu)$, so we write $\mathcal{A}\mathcal{B} = \bar{\mathcal{U}}^*$.

The PINN objective function is

$$\begin{aligned} J(\theta^N) &= \|\mathcal{A} Q_t^N - g\|^2_{L^2(\mu)} + \|Q_t^N - u\|^2_{L^2(\mu_{\mathbf{x}})} \\ &= \int_\Omega (\mathcal{A} Q_t^N - g)^2 d\mu + \int_\Omega (Q_t^N - u)^2 d\mu_{\mathbf{x}}. \end{aligned} \tag{6.5}$$

Informally, the wide-limit ODE (as $N \to \infty$) for the neural network trained with continuous-time gradient descent is:

$$\frac{dQ_t}{dt} = -\mathcal{U}[\mathcal{A} Q_t - g] - \mathcal{B}[Q_t - u]. \tag{6.6}$$

Applying the operator $\mathcal{A}$ and recalling $\mathcal{A}\mathcal{B}$ is the adjoint of $\bar{\mathcal{U}} : L^2(\mu) \to \mathbb{R}^M$, we derive that

$$\frac{d}{dt} \begin{bmatrix} \mathcal{A} Q_t - g \\ \mathbf{Q_t} - \mathbf{u} \end{bmatrix} = - \begin{bmatrix} \mathcal{S} & \bar{\mathcal{U}}^* \\ \bar{\mathcal{U}} & \bar{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \mathcal{A} Q_t - g \\ \mathbf{Q_t} - \mathbf{u} \end{bmatrix}, \tag{6.7}$$

where, on the left-hand side of the equation, the first term $\mathcal{A} Q_t - g \in L^2(\mu)$ and the second term $\mathbf{Q_t} - \mathbf{u} \in \mathbb{R}^M$ is

$$\mathbf{Q_t} - \mathbf{u} = \begin{bmatrix} Q_t(x_1) - u(x_1) \\ Q_t(x_2) - u(x_2)) \\ \vdots \\ Q_t(x_M) - u(x_M) \end{bmatrix}. \tag{6.8}$$

In light of this, we define the operator $\mathcal{V} : L^2(\mu) \times \mathbb{R}^M \to L^2(\mu) \times \mathbb{R}^M$

$$\mathcal{V} f := \begin{bmatrix} \mathcal{S} & \bar{\mathcal{U}}^* \\ \bar{\mathcal{U}} & \bar{\mathcal{B}} \end{bmatrix} f. \tag{6.9}$$

Given the properties of $\mathcal{S}$ (Lemma 4.4), it is easy to verify that $\mathcal{V}$ is Hilbert–Schmidt, self-adjoint and positive semi-definite. It will not generally be strictly positive definite. However, using the representation properties of neural networks, we can prove that $(\mathcal{A} Q_t - g, \mathbf{Q_t} - \mathbf{u})^\top$ in (6.7) has zero projection on the kernel set of $\mathcal{V}$.

LEMMA 6.1 (Projection on $\ker(\mathcal{V})$). *For $h \in L^2(\mu) \times \mathbb{R}^M$, if $\mathcal{V} h = 0$ then $\langle h, \mathcal{V} m \rangle = 0$ for any $m = [\mathcal{A} f - g, f(x_1) - u(x_1), \cdots, f(x_M) - u(x_M)]^\top$ where $f \in \mathcal{H}^2_{(0)}$.*

Proof. Let us begin with the following remark.

Remark 6.2. The setting we consider has a small modification as compared to the original PINNs formulation: we consider the limit ODE for an algorithm which samples points at random using the measure μ instead of using an *a priori* fixed set of points. We study this setting because if the PDE residual is minimized only on a fixed sparse set of points, then there is no guarantee that the approximator will converge to the solution – given any function, it is possible to perturb the function locally around each measurement point to construct functions which will have zero residual at these points; such functions will generally have no relation to the solution of the PDE. For similar reasons, we assume that the boundary value of the PDE is perfectly known.

Denote by $\varrho = \begin{bmatrix} \varrho^d \\ \varrho^p \end{bmatrix}$ an eigenfunction of $\mathcal{V}$ which has an eigenvalue zero:

$$\mathcal{V}\begin{bmatrix} \varrho^d \\ \varrho^p \end{bmatrix} = \mathcal{V}\varrho = 0\varrho = \begin{bmatrix} \mathcal{S} & \bar{\mathcal{U}}^* \\ \bar{\mathcal{U}} & \bar{\mathcal{B}} \end{bmatrix}\begin{bmatrix} \varrho^d \\ \varrho^p \end{bmatrix}. \tag{6.10}$$

We now show that

$$\left\langle \varrho, \begin{bmatrix} \mathcal{A}f - g \\ \mathbf{f} - \mathbf{u} \end{bmatrix} \right\rangle = 0, \tag{6.11}$$

for any $f \in \mathcal{H}^2_{(0)}$. This is because, similar to (4.8), we have

$$\left\langle \varrho, \mathcal{V}\varrho \right\rangle \geq \mathbb{E}\left[\left(\int_\Omega \varrho^d(x)\mathcal{A}[\eta(x)\sigma_{w,b}(x)]d\mu(x) + \int_\Omega \varrho^p(x)\eta(x)\sigma_{w,b}(x)d\mu_{\mathbf{x}}(x) \right)^2 \right] \geq 0. \tag{6.12}$$

Therefore, $\mathcal{V}\varrho = 0$ implies that for any (w, b) pairs

$$\int_\Omega \varrho^d(x)\mathcal{A}[\eta(x)\sigma(x; w, b)]d\mu(x) + \int_\Omega \varepsilon^p(x)\eta(x)\sigma(x; w, b)d\mu_{\mathbf{x}}(x) = 0. \tag{6.13}$$

Now, for any $f - u \in \mathcal{H}^2_{(0)}$, there exists a function sequence $\{F_N : \eta \sum_{i=1}^N c_i\sigma_{w,b}\}_{N\geq 1}$ such that

$$\lim_{N\to\infty} \|F_N - (f - u)\|^2_{\mathcal{H}^2} = 0, \tag{6.14}$$

and simultaneously

$$\lim_{N\to\infty} \|F_N - (f - u)\|^2_{L^2(\mu(\mathbf{x}))} = \frac{1}{M}\sum_{i=1}^M [F_N(x_i) - (f(x_i) - u(x_i))]^2 = 0, \tag{6.15}$$

we have

$$\int_\Omega \varrho^d(x)\mathcal{A}F^N(x)d\mu(x) + \int_\Omega \varrho^p(x)F^N(x)d\mu_{\mathbf{x}}(x) = 0. \tag{6.16}$$

This implies that

$$\int_\Omega \varrho^d(x)\mathcal{A}[f - u](x)d\mu(x) + \int_\Omega \varrho^p(x)[f - u](x)d\mu_{\mathbf{x}}(x) = 0, \tag{6.17}$$

and hence (6.11) is proven. ∎

THEOREM 6.3 (Global convergence of PINNs objective). *Assume that the PDE* (2.3) *admits a continuous* $\mathcal{H}^2$ *solution* u. *The objective function* $\|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})}$ *then converges to zero:*

$$\lim_{t\to\infty} \left(\|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})} \right) = 0. \tag{6.18}$$

Proof. From (6.7) we see that

$$\frac{d}{dt}\left[\|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})} \right] = -\begin{bmatrix} \mathcal{A}Q_t - g \\ \mathbf{Q_t} - \mathbf{u} \end{bmatrix}^\top \mathcal{V} \begin{bmatrix} \mathcal{A}Q_t - g \\ \mathbf{Q_t} - \mathbf{u} \end{bmatrix} \leq 0, \tag{6.19}$$

and that the equality holds iff $Q_t = u$. Therefore, the optimization objective is decreasing.

Consider $\tilde{\mathbf{Q}}_\mathbf{t} = [\mathcal{A}Q_t - g, \mathbf{Q_t} - \mathbf{u}]^\top$ projected on $\{\vartheta\}_{i\in\mathbb{N}^+} \cup \{\varrho_i\}_{i\in\mathbb{N}^+}$. By Lemma 6.1, we have

$$\tilde{\mathbf{Q}}_\mathbf{t} = \sum_i h_t^i \vartheta_i + \sum_j 0\varrho_j = \sum_i h_t^i \vartheta_i. \tag{6.20}$$

Therefore $\|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})} = \sum_i {h_t^i}^2$. Now consider its projection on each ϑ_i

$$\frac{d}{dt}\langle \tilde{\mathbf{Q}}_\mathbf{t}, \vartheta_i\rangle = \left\langle \frac{d\tilde{\mathbf{Q}}_\mathbf{t}}{dt}, \vartheta_i \right\rangle = \langle -\mathcal{V}\tilde{\mathbf{Q}}_\mathbf{t}, \vartheta_i\rangle = \langle \tilde{\mathbf{Q}}_\mathbf{t}, -\mathcal{V}\vartheta_i\rangle = -\lambda_i \langle \tilde{\mathbf{Q}}_\mathbf{t}, \vartheta_i\rangle. \tag{6.21}$$

Therefore

$$\frac{d}{dt}\langle \tilde{\mathbf{Q}}_\mathbf{t}, \vartheta_i\rangle = \frac{d}{dt}[h_t^i] = -\lambda_i h_t^i. \tag{6.22}$$

Consequently $h_t^i = h_0^i e^{-\lambda_i t}$ whose absolute value decays exponentially, and $|h_t^i| \leq |h_0^i|$ for any $t \geq 0$ and any i.

Now, by the dominated convergence theorem

$$\lim_{t\to\infty} \|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})} = \lim_{t\to\infty} \sum_i |h_t^i|^2 = \sum_i \lim_{t\to\infty} |h_t^i|^2 = 0. \tag{6.23}$$ ∎

Remark 6.4. The above theorem makes the additional assumption that the solution to our PDE is continuous. This is needed to ensure that the evaluation at the points x_i is meaningful, as strong solutions are generally only defined almost everywhere. It is also worth noting that, if the solution to the PDE is not unique, then Theorem 6.3 allows the user to choose a solution which achieves specific values $u(x_i)$, which can be specified in advance (assuming such a solution exists).

THEOREM 6.5 (Global convergence of PINNs). *If* $\mathcal{A}$ *has an* L^2 *bounded inverse and the PDE* (2.3) *admits a continuous* $\mathcal{H}^2$ *solution, then* Q_t *converges to the solution of the PDE*

$$\lim_{t\to\infty} \left(\|Q_t - u\|^2_{L^2(\mu)} + \|Q_t - u\|^2_{L^2(\mu_{\mathbf{x}})} \right) = 0.$$

Proof. As the existence of an inverse guarantees that the PDE (2.3) admits a (unique) solution, it is clear from Theorem 6.3 that $\|Q_t - u\|_{L^2(\mu_{\mathbf{x}})} \to 0$ and $\|\mathcal{A}Q_t - g\|^2_{L^2(\mu)} \to 0$. As in the proof of Theorem 5.5, the convergence of the residual implies the convergence of Q_t to u, given $\mathcal{A}^{-1}$ is a bounded operator. ∎

582 **7. Conclusion.** In this paper, we develop a convergence theory for neural net-
583 work approximators of PDEs trained with gradient descent (e.g. DGM and PINNs).
584 It is proven that a neural network trained by corresponding algorithms to minimize
585 the PDE residual will converge to an infinite-dimensional linear ODE as the number
586 of hidden units $\to \infty$. The limit ODE's dynamics are characterized by a novel kernel
587 function involving the PDE operator and the neural network activation function. The
588 kernel lacks a spectral gap, making the analysis of the limit ODE challenging. Using
589 an eigendecomposition approach, we are able to prove that the PDE residual of the
590 limit neural network converges to zero. Furthermore, under mild assumptions, the
591 limit neural network converges to the solution of the PDE.

592 In this paper, we do not cover free-boundary problems nor non-linear PDEs,
593 which are both important types of problems in financial mathematics contexts. From
594 a numerical usage perspective, there already exist applications of PINNs/DGM type
595 algorithms on free-boundary problems or non-linear PDEs, see for instance, [37, 20].
596 However, in terms of proving a mathematical convergence for the algorithm, there
597 are challenges that must be addressed. For free boundary problems, it is unlikely to
598 match the boundary condition exactly when the algorithm starts, which is a crucial
599 point in our analysis. For non-linear problems, the kernel will become very different
600 from the kernel for linear PDEs and there will be unique mathematical challenges
601 which must be addressed.

602 **Acknowledgments.** SC acknowledges the support of the UKRI Prosperity Part-
603 nership Scheme (FAIR) under EPSRC Grant EP/V056883/1 and the "Mathemat-
604 ical Foundations of Intelligence: An 'Erlangen Programme' for AI" under grant
605 EP/Y028872/1. DJ's research was supported by the EPSRC Centre for Doctoral
606 Training in Industrially Focused Mathematical Modelling (EP/L015803/1).

Appendix A. Proofs.

A.1. Proof of Theorem 3.8. The main idea of the proof is the split the difference into multiple residual terms and provide a bound for each of them. Then we apply Grönwall's inequality to provide an estimate of the difference between Q^N and Q. In this proof, constant C may vary from line to line, but it remains invariant with the number of neurons N and training time t. Expectations are taken with respect to random initialization of neural network parameters $\{c^i, w^i, b^i\}_{i=1,..,N}$.

We start with the following auxiliary lemmas which provide bounds for residual terms in our estimate. To simplify our notations, we may denote $\mathcal{L}Q := \mathcal{A}Q - g$.

LEMMA A.1. *We define residual term* $M_1 :=$

$$\int_0^T N^{2\beta+\delta-1}\bigg(\int_{\Omega^2}\Big|\Phi^N(\nabla_\theta \mathcal{A}Q_s^N(x))\cdot\sum_\alpha D_\alpha\nabla_\theta(Q_s^N-Q_0^N)(y)\Big|^2 d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}}ds, \tag{A.1}$$

It satisfies

$$\lim_{N\to\infty}\mathbb{E}M_1 = 0. \tag{A.2}$$

Proof. By Jensen's inequality, it suffices to prove that for any indices α

$$\int_0^T N^{2\beta+\delta-1}\Big(\int\mathbb{E}\Big[\Big|\Phi^N(\nabla_\theta\mathcal{A}Q_s^N(x))\cdot D_\alpha\nabla_\theta(Q_s^N-Q_0^N)(y)\Big|^2\Big]d\mu(x)d\mu(y)\Big)^{\frac{1}{2}}ds\to 0. \tag{A.3}$$

Notice that $|\Phi^N|$ is bounded elementwise by $N^{\epsilon-\beta}$. We also notice that by the mean value theorem and that η, σ, and their derivatives are up to polynomial growth

$$\begin{aligned}\|D_\alpha^y(\nabla_\theta Q_s^N-\nabla_\theta Q_0^N)(y)\|_1 \le \frac{C}{N^\beta}\sum_{i=1}^N\bigg(&(\|w_s^i-w_0^i\|+\|b_s^i-b_0^i\|\\&+\|c_s^i-c_0^i\|)f(w_0^i,c_0^i,b_0^i,y)\bigg)\end{aligned} \tag{A.4}$$

where f is a polynomial up to 3rd order. We notice that $\|w_u^i-w_0^i\|+\|b_u^i-b_0^i\|+\|c_u^i-c_0^i\|\le uCN^{\epsilon+\delta+\beta-1}$. Therefore

$$\mathbb{E}\Big[\Big|\Phi^N(\nabla_\theta\mathcal{A}Q_s^N(x))\cdot D_\alpha(\nabla_\theta Q_s^N-\nabla_\theta Q_0^N)(y)\Big|^2\Big]\le CN^{2(2\epsilon+\delta-\beta)}. \tag{A.5}$$

Therefore, the left-hand side of (A.3) is smaller than $CN^{2\delta+2\epsilon+\beta-1}$, which goes to 0 as $N\to\infty$. ∎

LEMMA A.2. *We define residual term*

$$\begin{aligned}M_2 := \int_0^T N^{2\beta+\delta-1}\bigg(\int\Big|(\Phi^N(\nabla_\theta\mathcal{A}Q_s^N(x))-\Phi^N(\nabla_\theta\mathcal{A}Q_0^N(x)))\cdot\sum_\alpha D_\alpha\nabla_\theta Q_0^N(y)\Big|^2\\ d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}}ds\end{aligned} \tag{A.6}$$

It satisfies

$$\lim_{N\to\infty}\mathbb{E}M_2 = 0. \tag{A.7}$$

Proof. Similarly, it suffices to show that for any α that

$$\int_0^T N^{2\beta+\delta-1}\bigg(\int_{\Omega^2}\mathbb{E}\bigg[\big|(\Phi^N(\nabla_\theta \mathcal{A}Q_s^N(x))-\Phi^N(\nabla_\theta \mathcal{A}Q_0^N(x)))\cdot D_\alpha\nabla_\theta Q_0^N(y)\big|^2\bigg] d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}}ds\to 0. \tag{A.8}$$

As, elementwise, $|\phi^N(x)-\phi^N(y)|\le|x-y|$, we have

$$\big|(\Phi^N(\nabla_\theta \mathcal{A}Q_s^N(x))-\Phi^N(\nabla_\theta \mathcal{A}Q_0^N(x))\cdot D_\alpha\nabla_\theta Q_0^N(y)\big|\le\big|\nabla_\theta \mathcal{A}Q_s^N(x)-\nabla_\theta \mathcal{A}Q_0^N(x)\big|\cdot\big|D_\alpha\nabla_\theta Q_0^N(y)\big| \tag{A.9}$$

Then, since

$$\big|(\nabla_\theta \mathcal{A}Q_s^N(x)-\nabla_\theta \mathcal{A}Q_0^N(x)\big|\cdot\big|D_\alpha\nabla_\theta Q_0^N(y)\big|\le\frac{C}{N^{2\beta}}\sum_{i=1}^N\bigg((\|w_s^i-w_0^i\|+\|b_s^i-b_0^i\|+\|c_s^i-c_0^i\|)g(w_0^i,c_0^i,b_0^i,x,y)\bigg) \tag{A.10}$$

where g is a polynomial up to 3rd order. Since $\|w_u^i-w_0^i\|+\|b_u^i-b_0^i\|+\|c_u^i-c_0^i\|\le uCN^{\epsilon+\delta+\beta-1}$, we have

$$\mathbb{E}\bigg[\big|(\Phi^N(\nabla_\theta \mathcal{A}Q_s^N(x))-\Phi^N(\nabla_\theta \mathcal{A}Q_0^N(x)))\cdot D_\alpha\nabla_\theta Q_0^N(y)\big|^2\bigg]\le CN^{2(\epsilon+\delta-\beta)}. \tag{A.11}$$

The left-hand side of (A.8) is bounded by $CN^{\epsilon+\delta+\beta-1}$, which goes to 0 as N goes to infinity. ∎

Lemma A.3. *We define residual term*

$$M_3:=\int_0^T N^{2\beta+\delta-1}\bigg(\int_{\Omega^2}\big|(\Phi^N(\nabla_\theta \mathcal{A}Q_0^N(x))-\nabla_\theta \mathcal{A}Q_0^N(x))\cdot\sum_\alpha D_\alpha\nabla_\theta Q_0^N(y)\big|^2 d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}}ds. \tag{A.12}$$

We have

$$\lim_{N\to\infty}\mathbb{E}M_3=0. \tag{A.13}$$

Proof. By Jensen's inequality, it suffices to show that

$$\int_0^T N^{2\beta+\delta-1}\bigg(\int_{\Omega^2}\mathbb{E}\bigg[\big|(\Phi^N(\nabla_\theta \mathcal{A}Q_0^N(x))-\nabla_\theta \mathcal{A}Q_0^N(x))\cdot D_\alpha\nabla_\theta Q_0^N(y)\big|^2\bigg] d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}}ds\to 0 \tag{A.14}$$

for any indices α where $|\alpha| \leq 2$. Let us denote the k-th unit of the initial approximator Q_0^N as q^k, where $q^k = c_0^k N^{-\beta} \eta \sigma_{w_0^k, b_0^k}$. Then $\mathbb{E}[|(\Phi^N(\nabla_\theta \mathcal{A} Q_0^N(x)) - \nabla_\theta \mathcal{A} Q_0^N(x)) \cdot D_\alpha \nabla_\theta Q_0^N(y)|^2]$

(A.15)

$$
\begin{aligned}
&= \mathbb{E}\bigg[\bigg(\sum_{k=1}^{N} [\Phi^N(\nabla_\theta \mathcal{A} q^k(x)) - \nabla_\theta \mathcal{A} q^k(x)] \cdot \nabla_\theta D_\alpha q^k(y)\bigg)^2\bigg] \\
&= N\mathbb{E}\Big[\big(\Phi^N(\nabla_\theta \mathcal{A} q(x)) - \nabla_\theta \mathcal{A} q(x)] \cdot \nabla_\theta D_\alpha q(y)\big)^2\Big] + N(N-1)\mathbb{E}\Big[\Phi^N(\nabla_\theta \mathcal{A} q(x)) \\
&\quad - \nabla_\theta \mathcal{A} q(x)] \cdot \nabla_\theta D_\alpha q(y)\Big]^2.
\end{aligned}
$$

Then, by definition

(A.16)

$$
\begin{aligned}
&\Phi^N(\nabla_\theta \mathcal{A} q(x)) - \nabla_\theta \mathcal{A} q(x)] \cdot \nabla_\theta D_\alpha q(y) \\
&= [\phi^N(\mathcal{A}[N^{-\beta} \eta \sigma_{w,b}](x)) - \mathcal{A}[N^{-\beta} \eta \sigma_{w,b}](x)] \cdot D_\alpha [N^{-\beta} \eta \sigma_{w,b}](y)] \\
&\quad + \sum_{i=1}^{d} [\phi^N(\mathcal{A}[N^{-\beta} c x_i \eta \sigma'_{w,b}](x)) - \mathcal{A}[N^{-\beta} c x_i \eta \sigma'_{w,b}](x)] \cdot D_\alpha [N^{-\beta} c y_i \eta \sigma'_{w,b}](y)] \\
&\quad + [\phi^N(\mathcal{A}[N^{-\beta} c \eta \sigma'_{w,b}](x)) - \mathcal{A}[N^{-\beta} c \eta \sigma'_{w,b}](x)] \cdot D_\alpha [N^{-\beta} c \eta \sigma'_{w,b}](y)].
\end{aligned}
$$

We introduce a uniform bound for $|D_\alpha[\eta \sigma_{w,b}(x)]|$, $|D_\alpha[c x_i \eta \sigma_{w,b}(x)]|$ and also the term $|D_\alpha[c \eta \sigma_{w,b}(x)]|$ for any indices α and any x. By the fact that $\mathcal{A}$ is Lipschitz,

$$
|\mathcal{A}[\eta \sigma_{w,b}](x)| \leq k_0 \sum_{0 \leq |\alpha| \leq 2} |D_\alpha [\eta \sigma_{w,b}](x)|. \tag{A.17}
$$

Notice that

$$
\begin{aligned}
|D_\alpha[\eta \sigma_{w,b}](x)| &= \bigg| \sum_{\alpha_1 + \alpha_2 = \alpha} D_{\alpha_1} \eta \cdot D_{\alpha_2} \sigma_{w,b}(x) \bigg| \leq k_\eta \bigg| \sum_{\alpha_1 + \alpha_2 = \alpha} D_{\alpha_2} \sigma_{w,b}(x) \bigg| \\
&\leq k_\eta \bigg| \sum_{0 \leq |\alpha_2| \leq 2} D_{\alpha_2} \sigma_{w,b}(x) \bigg| \\
&\leq k_\eta k_\sigma k_\Omega \bigg(1 + \sum_{i=1}^{d} |(w)_i| + \sum_{i,j=1}^{d} |(w)_i (w)_j| \bigg).
\end{aligned} \tag{A.18}
$$

Therefore

(A.19)

$$
|\mathcal{A}[\eta \sigma_{w,b}](x)| \leq k_0 \sum_{0 \leq |\alpha| \leq 2} |D_\alpha [\eta \sigma_{w,b}](x)| \leq k \bigg(1 + \sum_{i=1}^{d} |(w)_i| + \sum_{i,j=1}^{d} |(w)_i (w)_j| \bigg).
$$

Similar results hold for $|D_\alpha[c x_i \eta \sigma_{w,b}]|$ and $|D_\alpha[c \eta \sigma_{w,b}]|$. We define $f(w) := k(1 + \sum_{i=1}^{d} |(w)_i| + \sum_{i,j=1}^{d} |(w)_i (w)_j|)$. Now $|\phi^N(\mathcal{A}[N^{-\beta} \eta \sigma_{w,b}](x)) - \mathcal{A}[N^{-\beta} \eta \sigma_{w,b}](x)|$

$$
\begin{aligned}
&\leq N^{-\beta} \Big[|\mathcal{A}[\eta \sigma_{w,b}](x)| - N^\epsilon \Big] \mathbf{1}_{\{|\mathcal{A}[\eta \sigma_{w,b}](x)| \geq N^\epsilon\}}(x) \\
&\leq N^{-\beta} \big[f(w) - N^\epsilon \big] \mathbf{1}_{\{f(w) \geq N^\epsilon\}}(x).
\end{aligned} \tag{A.20}
$$

Subsequently,

(A.21)
$$|\Phi^N(\nabla_\theta \mathcal{A}q(x)) - \nabla_\theta \mathcal{A}q(x)] \cdot \nabla_\theta D_\alpha q(y)| \leq N^{-2\beta}(d+2)\big[f(w) - N^\epsilon\big]\mathbf{1}_{\{f(w)\geq N^\epsilon\}}(x).$$

Therefore

(A.22)
$$N\mathbb{E}\bigg[\big(\Phi^N(\nabla_\theta \mathcal{A}q(x)) - \nabla_\theta \mathcal{A}q(x)] \cdot \nabla_\theta D_\alpha q(y)\big)^2\bigg] \leq N^{1-4\beta}(d+2)\mathbb{E}[f(w)^2] \leq kN^{1-4\beta}.$$

Meanwhile $\mathbb{E}[\Phi^N(\nabla_\theta \mathcal{A}q(x)) - \nabla_\theta \mathcal{A}q(x)] \cdot \nabla_\theta D_\alpha q(y)]$

(A.23)
$$\begin{aligned}&\leq \mathbb{E}[N^{-2\beta}(d+2)\big[f(w) - N^\epsilon\big]\mathbf{1}_{\{f(w)\geq N^\epsilon\}}(x)] \leq N^{-2\beta}(d+2)\mathbb{E}[f(w)\mathbf{1}_{\{f(w)\geq N^\epsilon\}}(x)]\\ &\leq k_0 N^{-2\beta-\epsilon}\mathbb{E}[f(w)^2] \leq kN^{-2\beta-\epsilon}.\end{aligned}$$

Hence

$$N(N-1)\mathbb{E}\bigg[\Phi^N(\nabla_\theta \mathcal{A}q(x)) - \nabla_\theta \mathcal{A}q(x)] \cdot \nabla_\theta D_\alpha q(y)\bigg]^2 \leq kN^{2-4\beta-2\epsilon}. \tag{A.24}$$

Combining (A.22) and (A.24), we conclude that

$$M_3 \leq k(N^{\delta-1/2} + N^{\delta-\epsilon}) \to 0 \tag{A.25}$$

as $N \to \infty$. ∎

Lemma A.4. *We define residual term*

(A.26)
$$\begin{aligned}M_4 :&= \mathbb{E}\bigg[\int_0^T N^\delta\bigg(\int_{\Omega^2} \big|N^{2\beta-1}\nabla_\theta \mathcal{A}Q_0^N(x) \cdot \sum_\alpha D_\alpha \nabla_\theta Q_0^N(y) - \sum_\alpha D_\alpha^y U(x,y)\big|^2\\ &\quad d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}} ds\bigg].\end{aligned}$$

It satisfies

$$\lim_{N\to\infty} \mathbb{E}M_4 = 0. \tag{A.27}$$

Proof. It suffices to prove that for any indices α

(A.28)
$$\int_0^T N^\delta\bigg(\int \mathbb{E}\bigg[\big|N^{2\beta-1}\nabla_\theta \mathcal{A}Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y) - D_\alpha^y U(x,y)\big|^2\bigg] d\mu(x)d\mu(y)\bigg)^{\frac{1}{2}} ds$$

converges to 0 as $N \to \infty$. By the strong law of large numbers:

$$\lim_{N\to\infty} N^{2\beta-1}\nabla_\theta \mathcal{A}Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y) = D_\alpha^y U(x,y). \tag{A.29}$$

Therefore

(A.30)
$$\begin{aligned}&\mathbb{E}\bigg[\big|N^{2\beta-1}\nabla_\theta \mathcal{A}Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y) - D_\alpha^y U(x,y)\big|^2\bigg]\\ &= \mathrm{Var}\big[N^{2\beta-1}\nabla_\theta \mathcal{A}Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y)\big] = \frac{1}{N}\mathrm{Var}[D_\alpha^y U_{c,w,b}(x,y)].\end{aligned}$$

Consequently, the left-hand side of (A.28) is bounded by $CN^{\delta-1}$ which vanishes as N goes to infinity. ∎

LEMMA A.5. *We have*

$$\lim_{N\to\infty}\int_0^T\int_\Omega \left|\psi^N(\mathcal{L}Q_s(x)) - \mathcal{L}Q_s(x)\right| d\mu(x)ds = 0. \tag{A.31}$$

Proof. Notice that

$$\left|\psi^N(\mathcal{L}Q_s(x)) - \mathcal{L}Q_s(x)\right| \le |\mathcal{L}Q_s(x)|\mathbf{1}_{\{|\mathcal{L}Q_s|>N^\delta\}}(x). \tag{A.32}$$

By the dominated convergence theorem, we conclude our proof. ∎

LEMMA A.6. *We have*

$$\lim_{N\to\infty}\mathbb{E}\left[\left(\int_\Omega \left[\sum_\alpha |D_\alpha(Q_0^N - Q_0)(y)|\right]^2 d\mu(y)\right)^{\frac{1}{2}}\right] = 0. \tag{A.33}$$

Proof. By Jensen's inequality, it suffices to show that for any α

$$\int_\Omega \mathbb{E}\left[D_\alpha(Q_0^N - Q_0)(y)^2\right] d\mu(y) \to 0. \tag{A.34}$$

Since $\mathbb{E}[Q_0^N] = Q_0 = 0$ holds for all y we have

$$\mathbb{E}\left[D_\alpha(Q_0^N - Q_0)(y)^2\right] = \mathrm{Var}[D_\alpha Q_0^N(y)] = N^{1-2\beta}\mathrm{Var}[D_\alpha c\sigma(wy+b)] \le CN^{1-2\beta}. \tag{A.35}$$

Integrating with respect to $\mu(y)$, and letting N go to infinity finish the proof. ∎

Now we present our main proof.

Proof of Theorem 3.8. In the integral form, we have

$$Q_t^N(y) = Q_0^N(y) - \int_0^t\int_\Omega \psi^N(\mathcal{L}Q_t^N(x))\alpha^N\Phi^N(\nabla_\theta \mathcal{A}Q_s^N(x))\cdot\nabla_\theta Q_s^N(y)d\mu(x)ds. \tag{A.36}$$

Similarly, at the same time

$$Q_t(y) = Q_0(y) - \int_0^t\int_\Omega \mathcal{L}Q_t(x)U(x,y)d\mu(x)ds. \tag{A.37}$$

Subtracting (A.37) from (A.36) and taking partial derivative with respect to y with indices α give

$$\begin{aligned}&|D_\alpha(Q_t^N - Q_t)(y)|\\ &\le \int_0^t\int_\Omega \left|\psi^N(\mathcal{L}Q_s^N(x))\alpha^N\Phi^N(\nabla_\theta\mathcal{A}Q_s^N(x))\cdot D_\alpha\nabla_\theta Q_s^N(y) - \mathcal{L}Q_t^N(x)D_\alpha^y U(x,y)\right|\\ &\quad d\mu(x)ds + |D_\alpha(Q_0^N - Q_0)(y)|,\end{aligned} \tag{A.38}$$

where the right-hand side can be further bounded by

(A.39)
$$
\begin{aligned}
&\leq \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s^N(x)) \alpha^N \Phi^N(\nabla_\theta \mathcal{A} Q_s^N(x)) \cdot D_\alpha (\nabla_\theta Q_s^N - \nabla_\theta Q_0^N)(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s^N(x)) \alpha^N (\Phi^N(\nabla_\theta \mathcal{A} Q_s^N(x)) - \Phi^N(\nabla_\theta \mathcal{A} Q_0^N(x)) \cdot \right. \\
&\qquad \left. D_\alpha \nabla_\theta Q_0^N(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s^N(x)) \alpha^N (\Phi^N(\nabla_\theta \mathcal{A} Q_0^N(x)) - \nabla_\theta \mathcal{A} Q_0^N(x)) \cdot D_\alpha \nabla_\theta Q_0^N(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s^N(x)) \left[\alpha^N \nabla_\theta \mathcal{A} Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y) - D_\alpha^y U(x,y) \right] \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega \left| \left[\psi^N(\mathcal{L}Q_s^N(x)) - \psi^N(\mathcal{L}Q_s(x)) \right] D_\alpha^y U(x,y)) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega \left| \left[\psi^N(\mathcal{L}Q_s(x)) - \mathcal{L}Q_s(x)) \right] D_\alpha^y U(x,y)) \right| d\mu(x) ds + |D_\alpha(Q_0^N - Q_0)(y)|.
\end{aligned}
$$

Now by the fact that $|\psi^N| < N^\delta$, $|D_\alpha^y U(x,y)| < C$ we have

(A.40)
$$
\begin{aligned}
&|D_\alpha(Q_t^N - Q_t)(y)| \\
&\leq \int_0^t \int_\Omega N^{2\beta+\delta-1} \left| \Phi^N(\nabla_\theta \mathcal{A} Q_s^N(x)) \cdot D_\alpha (\nabla_\theta Q_s^N - \nabla_\theta Q_0^N)(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega N^{2\beta+\delta-1} \left| (\Phi^N(\nabla_\theta \mathcal{A} Q_s^N(x)) - \Phi^N(\nabla_\theta \mathcal{A} Q_0^N(x)) \cdot D_\alpha \nabla_\theta Q_0^N(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega N^{2\beta+\delta-1} \left| (\Phi^N(\nabla_\theta \mathcal{A} Q_0^N(x)) - \nabla_\theta \mathcal{A} Q_0^N(x)) \cdot D_\alpha \nabla_\theta Q_0^N(y) \right| d\mu(x) ds \\
&\quad + \int_0^t \int_\Omega N^{\delta} \left| N^{2\beta-1} \nabla_\theta \mathcal{A} Q_0^N(x) \cdot D_\alpha \nabla_\theta Q_0^N(y) - D_\alpha^y U(x,y) \right| d\mu(x) ds \\
&\quad + C \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s^N(x)) - \psi^N(\mathcal{L}Q_s(x)) \right| d\mu(x) ds \\
&\quad + C \int_0^t \int_\Omega \left| \psi^N(\mathcal{L}Q_s(x)) - \mathcal{L}Q_s(x)) \right| d\mu(x) ds + |D_\alpha(Q_0^N - Q_0)(y)|.
\end{aligned}
$$

Let us denote $V_t^N(y) := \sum_{|\alpha| \leq 2} |D_\alpha(Q_t^N - Q_t)(y)|$. Then by summing inequality (A.40) with respect to all indices and integrating with respect to $\mu(y)$

(A.41)
$$
\int_\Omega V_t^N(y)^2 d\mu(y) \leq \left(\int_\Omega V_t^N(y)^2 d\mu(y) \right)^{\frac{1}{2}} \left[C \int_0^t \left(\int_\Omega V_s^N(x)^2 d\mu(x) \right)^{\frac{1}{2}} ds + M \right]
$$

where $M := M_1 + M_2 + M_3 + M_4 + M_5 + M_6$ denotes the sum of residual terms where

718 M_1 to M_4 have be defined in previous lemmas and

$$
\begin{aligned}
M_5 &:= \int_0^T \int_\Omega \left|\psi^N(\mathcal{L}Q_s(x)) - \mathcal{L}Q_s(x)\right| d\mu(x) ds, \\
M_6 &:= \left(\int_\Omega \Big[\sum_\alpha |D_\alpha (Q_0^N - Q_0)(y)| \Big]^2 d\mu(y) \right)^{\frac{1}{2}}.
\end{aligned}
\tag{A.42}
$$

720 Then by Grönwall's inequality, since from (A.41) we have

$$
\left(\int_\Omega V_t^N(y)^2 d\mu(y) \right)^{\frac{1}{2}} \leq C \int_0^t \left(\int_\Omega V_s^N(x)^2 d\mu(x) \right)^{\frac{1}{2}} ds + M, \tag{A.43}
$$

722 we derive that

$$
\int_0^t \left(\int_\Omega V_s^N(x)^2 d\mu(x) \right)^{\frac{1}{2}} ds \leq \frac{M}{C} e^{Ct}. \tag{A.44}
$$

724 Taking expectation on both sides, by Lemmas A.1, A.2, A.3, A.4, A.5, A.6 we can
725 control each of the terms in M. Hence

$$
\lim_{N\to\infty} \mathbb{E}\left[\int_0^t \left(\int_\Omega V_s^N(x)^2 d\mu(x) \right)^{\frac{1}{2}} ds \right] = 0. \tag{A.45}
$$

727 By (A.43) we have for any $0 \leq s \leq T$

$$
\lim_{N\to\infty} \mathbb{E}\left[\left(\int_\Omega V_s^N(x)^2 d\mu(x) \right)^{\frac{1}{2}} \right] = 0. \tag{A.46}
$$

729 Notice that $\int_\Omega V_s^N(x)^2 d\mu(x) \geq \|Q_s^N - Q_s\|_{\mathcal{H}^2}$, we conclude that for any $0 \leq s \leq T$

$$
\lim_{N\to\infty} \mathbb{E}\left[\|Q_s^N - Q_s\|_{\mathcal{H}^2} \right] = 0. \tag{A.47}
$$

□

731 REFERENCES

732 [1] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck, *An overview on deep*
733 *learning-based approximation methods for partial differential equations*, arXiv preprint
734 arXiv:2012.12348, (2020).
735 [2] J. Berner, P. Grohs, and A. Jentzen, *Analysis of the generalization error: Empirical risk*
736 *minimization over deep artificial neural networks overcomes the curse of dimensionality in*
737 *the numerical approximation of black-scholes partial differential equations*, SIAM Journal
738 on Mathematics of Data Science, 2 (2020).
739 [3] M. Bichuch and K. Chen, *A deep learning scheme for solving fully nonlinear partial differ-*
740 *ential equation*, Peter Carr Gedenkschrift: Research Advances in Mathematical Finance,
741 (2023), pp. 101–140.
742 [4] R. Carmona and M. Laurière, *Convergence analysis of machine learning algorithms for the*
743 *numerical solution of mean field control and games i: The ergodic case*, SIAM Journal on
744 Numerical Analysis, 59 (2021).
745 [5] ———, *Deep learning for mean field games and mean field control with applications to finance*,
746 arXiv:2107.04568, (2021).
747 [6] ———, *Convergence analysis of machine learning algorithms for the numerical solution of mean*
748 *field control and games: II—the finite horizon case*, The Annals of Applied Probability,
749 32 (2022), pp. 4065–4105.

[7] C.Beck, S.Becker, P.Cheridito, A.Jentzen, and A.Neufeld, *Deep splitting method for parabolic PDEs*, SIAM Journal on Scientific Computing, 43 (2021).

[8] S. N. Cohen, D. Jiang, and J. Sirignano, *Neural Q-learning for solving elliptic PDEs*, arXiv preprint arXiv:2203.17128, (2022).

[9] T. De Ryck, A. D. Jagtap, and S. Mishra, *Error estimates for physics-informed neural networks approximating the navier–stokes equations*, IMA Journal of Numerical Analysis, 44 (2024), pp. 83–119.

[10] N. Doumèche, G. Biau, and C. Boyer, *Convergence and error analysis of PINNs*, arXiv preprint arXiv:2305.01240, (2023).

[11] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab, *DNN expression rate analysis of high-dimensional PDEs: application to option pricing*, Constructive Approximation, 55 (2022).

[12] L. C. Evans, *Partial differential equations*, American Mathematical Society, second ed., 2010.

[13] C. Gao, S. Gao, and Z. Zhu, *Convergence of the backward deep BSDE method with applications to optimal stopping problems*, SIAM Journal on Financial Mathematics, 14 (2023).

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.

[15] P. Grohs, F. Hornung, A. Jentzen, and P. V. Wurstemberger, *A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations*, Memoirs of the American Mathematical Society, 284 (2023).

[16] P. Grohs, A. Jentzen, and D. Salimova, *Deep neural network approximations for solutions of PDEs based on monte carlo algorithms*, Partial Differential Equations and Applications, 3 (2022).

[17] J. Han, A. Jentzen, and W. E, *Solving high-dimensional partial differential equations using deep learning*, Proceedings of the National Academy of Sciences, 115 (2018).

[18] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural networks, 4 (1991), pp. 251–257.

[19] A. Jacot, F. Gabriel, and C. Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, Advances in neural information processing systems, 31 (2018).

[20] X. Jiang, D. Wang, Q. Fan, M. Zhang, C. Lu, and A. P. T. Lau, *Physics-informed neural network for nonlinear dynamics in fiber optics*, Laser & Photonics Reviews, 16 (2022), p. 2100483.

[21] I. E. Lagaris, A. Likas, and D. I. Fotiadis, *Artificial neural networks for solving ordinary and partial differential equations*, IEEE transactions on neural networks, 9 (1998), pp. 987–1000.

[22] H. Lee and I. S. Kang, *Neural algorithm for solving differential equations*, Journal of Computational Physics, 91 (1990), pp. 110–131.

[23] Y. Lu, J. Blanchet, and L. Ying, *Sobolev acceleration and statistical optimality for learning elliptic equations via gradient descent*, Advances in Neural Information Processing Systems, 35 (2022), pp. 33233–33247.

[24] T. Luo and H. Yang, *Two-layer neural networks for partial differential equations: Optimization and generalization theory*, arXiv preprint arXiv:2006.15733, (2020).

[25] A. Malek and R. S. Beidokhti, *Numerical solution for high order differential equations using a hybrid neural network—optimization method*, Applied Mathematics and Computation, 183 (2006), pp. 260–271.

[26] K. S. McFall and J. R. Mahan, *Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions*, IEEE Transactions on Neural Networks, 20 (2009), pp. 1221–1233.

[27] S. Mishra and R. Molinaro, *Estimates on the generalization error of physics-informed neural networks for approximating PDEs.*, IMA Journal of Numerical Analysis, 43 (2023).

[28] R. Pascanu, T. Mikolov, and Y. Bengio, *On the difficulty of training recurrent neural networks*, in International conference on machine learning, Pmlr, 2013, pp. 1310–1318.

[29] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational physics, 378 (2019), pp. 686–707.

[30] A. Reppen, H. Soner, and V. Tissot-Daguette, *Deep stochastic optimization in finance*, Digital Finance, 5 (2022).

[31] K. Rudd, *Solving partial differential equations using artificial neural networks*, PhD thesis, Duke University, 2013.

[32] Y. Saporito and Z. Zhang, *Path-dependent deep Galerkin method: A neural network approach to solve path-dependent partial differential equations*, SIAM Journal on Financial Mathematics, 12 (2021).

[33] Y. Shin, J. Darbon, and G. E. Karniadakis, *On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs*, arXiv preprint arXiv:2004.01806, (2020).
[34] J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, Journal of computational physics, 375 (2018), pp. 1339–1364.
[35] ———, *Mean field analysis of neural networks: A law of large numbers*, SIAM Journal on Applied Mathematics, 80 (2020).
[36] ———, *Asymptotics of reinforcement learning with neural networks*, Stochastic Systems, 12 (2022).
[37] S. Wang and P. Perdikaris, *Deep learning of free boundary and stefan problems*, Journal of Computational Physics, 428 (2021), p. 109914.
[38] S. Wang, X. Yu, and P. Perdikaris, *When and why PINNs fail to train: A neural tangent kernel perspective*, Journal of Computational Physics, 449 (2022), p. 110768.
[39] J. Zhang, T. He, S. Sra, and A. Jadbabaie, *Why gradient clipping accelerates training: A theoretical justification for adaptivity*, arXiv preprint arXiv:1905.11881, (2019).