

STRUCTURE-INFORMED OPERATOR LEARNING FOR PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS

FRED ESPEN BENTH, NILS DETERING, LUCA GALIMBERTI

ABSTRACT. In this paper, we present a framework for learning the solution map of a backward parabolic Cauchy problem. The solution depends continuously but nonlinearly on the final data, source, and force terms, all residing in Banach spaces of functions. We utilize Fréchet space neural networks (Benth et al. (2023)) to address this operator learning problem. Our approach provides an alternative to Deep Operator Networks (DeepONets), using basis functions to span the relevant function spaces rather than relying on finite-dimensional approximations through censoring. With this method, structural information encoded in the basis coefficients is leveraged in the learning process. This results in a neural network designed to learn the mapping between infinite-dimensional function spaces. Our numerical proof-of-concept demonstrates the effectiveness of our method, highlighting some advantages over DeepONets.

1. INTRODUCTION

In this paper we provide a framework for learning the solution map of a backward parabolic Cauchy problem incorporating structural information about the final datum, source, and the force term. The solution of the Cauchy problems depends in a nonlinear, but continuous way on the final datum, the source and force terms, which are all functions living in appropriate Banach spaces. We propose to use the recently introduced neural networks in Fréchet spaces (see Benth et al. (2023)), an infinite dimensional neural network structure, to solve this operator learning problem.

Our approach provides an alternative to the operator learning method Deep Operator Networks (DeepONets), studied by Lu et al. (2019) and Lu et al. (2021). DeepONet extends the shallow network for operator learning proposed and analysed in Chen & Chen (1995). The validity of these operator learning approaches is resting on the universal approximation theorem (see Chen & Chen (1995), Lu et al. (2021) and more recently a generalisation by Lanthaler et al. (2022) to measurable operators). Instead of using finite-dimensional neural networks approximating sampled (also called censored) expressions of the input functions in the operator in question, as done in DeepONets, we make use of the information contained in the basis functions spanning the relevant function spaces. We build a neural network which is learning the map between these function spaces expressed by their basis functions. An infinite dimensional activation function allows us to set up a deep neural network that preserves the structural information encoded in the basis coefficients when processing through the layers.

Our approach rests on a truly infinite dimensional neural network, reflecting that we are approximating continuous nonlinear operators between infinite dimensional spaces. Implemented on a computer, we sample a finite set of basis functions in the training rather than censoring the input functions to have finite dimensional approximations. We refer also to (Kovachki et al. 2022, Sect. 2, p. 9) where a similar idea was mentioned but not further explored.

Often linear operators between function spaces can be expressed as integral operators, for example as convolutions. An approximation of such integral operators using graph kernels is proposed in Anandkumar et al. (2019) for operator learning of partial differential equations. The authors take an infinite dimensional perspective in learning operator maps from various parameters into the solution, viewed as continuous mappings between function spaces of Sobolev type. The affine transform-part of the neural network is viewed as an integral kernel

operator, and a discrete version of this is mapped to the next layer by a finite dimensional activation function. These ideas are further expanded into Fourier neural operators (see Li et al. (2021), Kovachki et al. (2022)), where the kernel is represented by the Fourier transform and its inverse to obtain computationally attractive representations for estimating the kernel operator. Cao et al. (2024) propose to use the Laplace transform instead of the Fourier transform, taking advantage of the pole-residue relationship between the input and output space of the operator to be approximated. In Kovachki et al. (2022) a universal approximation theorem is shown for networks learning nonlinear operators between certain Banach spaces of functions with infinite dimensional (integral operator) affine transforms, but finite dimensional activation functions. In a recent paper Li et al. (2024) the Fourier neural operator methodology is applied in conjunction with physics informed learning. We refer the reader to the extensive literature review on operator and physics informed learning learning in Li et al. (2024) (see also more references in Cao et al. (2024)).

In Benth et al. (2023) a universal approximation theorem was shown which ensures that continuous operators from a Fréchet space into a Banach space can be approximated on compacts with Fréchet neural networks. In our analysis of the parabolic Cauchy problem, using the classical theory of e.g. Friedman (1975), the solution map can be shown to be Lipschitz continuous from a product of Sobolev spaces into the continuous functions on compacts. Thus, the universal approximation theorem ensures that we can approximate the operator arbitrary well by the Fréchet neural network. In Lanthaler et al. (2022) the operators they are interested in training using DeepONet are shown to be Lipschitz continuous. For the Fourier neural operators studied in Kovachki et al. (2022), proofs of the integrability properties of the operators seem to be missing for the operators they aim to train that enables them to use of their universal approximation theorem. Cao et al. (2024) do not address the regularity properties for the operators they study numerically.

In a numerical proof-of-concept, we demonstrate that our proposed approach indeed works out. We compare with the DeepONets, and point out some advantages with our approach. We once again would like to emphasise also that we make use of infinite dimensional activation functions, and not finite dimensional ones as in DeepONet and the Fourier neural operator approaches, because for deep learning this allows us to progress structural information from the basis functions throughout the layers.

Our approach and analysis extends the neural network approach to approximate numerically high dimensional partial differential equations by training using synthetic data generated by stochastic differential equations along with the Feynman-Kac formula, as advocated by E et al. (2017) and Han et al. (2018) (see also Beck et al. (2023) for an overview and further references). We can naturally make use of the Feynman-Kac formula also for operator learning problems related to Cauchy problems, as we show in this paper. In passing we remark that Benth et al. (2024) have made use of similar ideas to price options in energy markets which require an infinite dimensional framework.

Learning the operator mapping certain parameter functions into the solution of partial differential equations is a forward problem, often referred to as “many query”. A neural operator method allows for fast and efficient computations of the solution for various specifications of the input parameter functions (see Lanthaler et al. (2022) for a discussion and analysis of the curse of dimensionality in such problems). The inverse problem, where observations of the dynamical system in question is available and one wants to back out parameters, is also of interest, and has been empirically investigated in a Bayesian framework in Li et al. (2021). Rather than learning the operator map for a specific dynamical system, Yang et al. (2023) propose a framework to learn operators connected to a family of differential equations. Empirical evidence demonstrates that commonalities across differential equations reduce the training burden in such an approach.

Our results are presented as follows. In the next section we provide a review of infinite dimensional neural network introduced in Benth et al. (2023), collecting some useful material. Section 3 defines the Cauchy problem relying on the classical analysis of Friedman (1975), and identifies the operator maps that will be the core object of analysis in this paper. To use infinite dimensional neural networks to learn the operator maps, we need continuity properties to hold according to the universal approximation theorem. Continuity of the nonlinear

operator maps are analysed and shown for Sobolev spaces in this Section. A numerical example is given in Section 4, providing a proof-of-concept. Here we are benchmarking our proposed method with the DeepONets-approach, and provide further extensions and perspectives of our proposed structure-informed operator learning approach.

2. A BRIEF INTRODUCTION TO NEURAL NETWORKS IN INFINITE DIMENSIONS

In this Section, we give a brief review of neural networks defined on infinite dimensional spaces, following the approach in Benth et al. (2023). Although Benth et al. (2023) consider networks on Fréchet spaces, we focus on the case of real Banach spaces in the account given here.

Let \mathfrak{X} and \mathfrak{Y} be two real Banach spaces with norms denoted by $\|\cdot\|_{\mathfrak{X}}$ and $\|\cdot\|_{\mathfrak{Y}}$, resp. We are interested in learning continuous nonlinear operators $F : \mathfrak{X} \rightarrow \mathfrak{Y}$ by neural networks. We denote $C(\mathfrak{X}; \mathfrak{Y})$ the space of such continuous operators, equipped with the topology of uniform convergence on compacts. If $\mathfrak{Y} = \mathbb{R}$, we use the simpler notation $C(\mathfrak{X})$ to denote $C(\mathfrak{X}; \mathbb{R})$.

Let us start with defining a one layer real-valued neural network on \mathfrak{X} . Let $A \in \mathcal{L}(\mathfrak{X})$, i.e. a linear and continuous operator $A : \mathfrak{X} \rightarrow \mathfrak{X}$, and $\beta \in \mathfrak{X}$. Thus, $\mathfrak{X} \ni \xi \rightarrow A\xi + \beta \in \mathfrak{X}$ is an *affine* transform on \mathfrak{X} . Introduce an *activation function* $\sigma : \mathfrak{X} \rightarrow \mathbb{R}$ being continuous, and define a *neuron* $\mathcal{N}_{\ell, A, \beta} : \mathfrak{X} \rightarrow \mathbb{R}$ to be the map

$$(1) \quad \mathcal{N}_{\ell, A, \beta}(\xi) = \ell(\sigma(A\xi + \beta))$$

for $\ell \in \mathfrak{X}'$. Here, \mathfrak{X}' is the (topological) dual of \mathfrak{X} , i.e., the space of continuous linear functionals $\ell : \mathfrak{X} \rightarrow \mathbb{R}$. A one-layer neural network of width $M \in \mathbb{N}$ is given as a linear superposition of M such neurons,

$$(2) \quad \mathcal{N}(\xi) := \sum_{j=1}^M \mathcal{N}_{\ell_j, A_j, \beta_j}(\xi).$$

As activation functions σ , we restrict our attention to the "sigmoidal" class. As we operate with infinite dimensional activation functions, the sigmoidal property requires some care. Here is (Benth et al. 2023, Def. 2.6), where a separation property is introduced for continuous $\sigma : \mathfrak{X} \rightarrow \mathbb{R}$:

Definition 2.1 (Separation property). *There exist $\psi \in \mathfrak{X}' \setminus \{0\}$ and $\nu_+, \nu_-, \nu_0 \in \mathbb{R}$ such that either $\nu_+ \notin \text{span}\{\nu_0, \nu_-\}$ or $\nu_- \notin \text{span}\{\nu_0, \nu_+\}$ and such that*

$$\begin{aligned} \lim_{\lambda \rightarrow \infty} \sigma(\lambda\xi) &= \nu_+, \quad \text{if } \xi \in \Psi_+ \\ \lim_{\lambda \rightarrow \infty} \sigma(\lambda\xi) &= \nu_-, \quad \text{if } \xi \in \Psi_- \\ \lim_{\lambda \rightarrow \infty} \sigma(\lambda\xi) &= \nu_0, \quad \text{if } \xi \in \Psi_0, \end{aligned}$$

where $\Psi_+ := \{\xi \in \mathfrak{X} \mid \psi(\xi) > 0\}$, $\Psi_- := \{\xi \in \mathfrak{X} \mid \psi(\xi) < 0\}$ and $\Psi_0 := \ker(\psi)$.

We also need the activation functions to be bounded, in the sense that its image $\sigma(\mathfrak{X})$ is a bounded subset of \mathbb{R} . Indeed, one sees that with continuous and bounded activation functions, the one-layer neural networks that we have introduced are also continuous and bounded.

From (Benth et al. 2023, Example 2.13) we find a specification of a class of bounded and continuous activation functions which enjoy the separation property. Let us recall this example: fix $\psi \in \mathfrak{X}' \setminus \{0\}$ and suppose we have a sequence $(\tilde{\sigma}_j)_{j \in \mathbb{N}}$ of continuous functions $\tilde{\sigma}_j : \mathbb{R} \rightarrow \mathbb{R}$ with the properties $\tilde{\sigma}_j(0) = 0$,

$$\lim_{x \rightarrow \infty} \tilde{\sigma}_j(x) = 1, \quad \lim_{x \rightarrow -\infty} \tilde{\sigma}_j(x) = 0,$$

for all $j \in \mathbb{N}$, where additionally we suppose $\sup_{j \in \mathbb{N}} \|\tilde{\sigma}_j\|_{\infty} < \infty$. Given a summable sequence $(\zeta_j)_{j \in \mathbb{N}} \subset \mathbb{R}$ for which $0 \neq \zeta := \sum_{j=1}^{\infty} \zeta_j$, define the function $\sigma : \mathfrak{X} \rightarrow \mathbb{R}$ by

$$(3) \quad \sigma(\xi) = \sum_{j=1}^{\infty} \tilde{\sigma}_j(\psi(\xi)) \zeta_j$$

One can show (see (Benth et al. 2023, Example 2.13)) that σ is continuous, bounded and separating, and thus suitable as an activation function. Notice that σ is a sum of $\tilde{\sigma}_j(\psi(\xi))\zeta_j$, where $\tilde{\sigma}_j$ are classical sigmoidal activation functions. We first let a global linear functional act on ξ before it goes as input into the activation function $\tilde{\sigma}_j$, having a real value. We use this real value to scale the element ζ_j in the Banach space to build up an activation function which truly maps \mathfrak{X} into itself. In a simple setting, one can choose only one such element in the series (or a finite number of such), defining the activation function $\sigma(\xi) := \tilde{\sigma}(\psi(\xi))\zeta$. We mention in passing that there exist other examples of infinite dimensional activation functions (see Benth et al. (2023)).

To obtain neural networks with values in the Banach space \mathfrak{Y} , we simply aggregate the real-valued neural networks we have defined in (2), scaled by independent unit vectors in \mathfrak{Y} . To be more precise, given d independent unit vectors $\mu_1, \dots, \mu_d \in \mathfrak{Y}$ and neural networks $\mathcal{N}^{(1)}, \dots, \mathcal{N}^{(d)}$, define the \mathfrak{Y} -valued neural network $\mathcal{N}_d : \mathfrak{X} \rightarrow \mathfrak{Y}$ as

$$(4) \quad \mathcal{N}_d(\xi) = \sum_{i=1}^d \mathcal{N}^{(i)}(\xi)\mu_i.$$

For such neural networks we have the following *Universal Approximation Theorem* (see (Benth et al. 2023, Thm. 3.2)):

Theorem 2.2. *Assume $\sigma : \mathfrak{X} \rightarrow \mathfrak{X}$ is a continuous, bounded and separating activation function and suppose that $F \in C(\mathfrak{X}; \mathfrak{Y})$. Then, for given compact set $\mathcal{K} \subset \mathfrak{X}$ and $\epsilon > 0$ there exist $d \in \mathbb{N}$, d independent unit vectors $\mu_1, \dots, \mu_d \in \mathfrak{Y}$ and d real-valued neural networks $\mathcal{N}^{(1)}, \dots, \mathcal{N}^{(d)}$ such that*

$$\sup_{\xi \in \mathcal{K}} \|F(\xi) - \mathcal{N}_d(\xi)\|_{\mathfrak{Y}} < \epsilon,$$

where \mathcal{N}_d is defined in (4).

This universal approximation theorem is the key to the applicability of infinite dimensional neural networks in operator-learning tasks. Remark that if we let $\mathfrak{Y} = \mathbb{R}$, then we can re-state the universal approximation theorem as follows: for given $\epsilon > 0$ and compact subset $\mathcal{K} \subset \mathfrak{X}$, there exists $M \in \mathbb{N}$ such that

$$\sup_{\xi \in \mathcal{K}} |F(\xi) - \mathcal{N}(\xi)| < \epsilon,$$

where \mathcal{N} is defined in (2). By saying that there exists an M , we are in reality also saying that there exist elements $\ell_j \in \mathfrak{X}'$, $A_j \in \mathcal{L}(\mathfrak{X})$ and $\beta_j \in \mathfrak{X}$ for $j = 1, \dots, M$. Obviously, all these elements, including M , depend on ϵ and \mathcal{K} .

To implement the neural network \mathcal{N}_d on a computer we need a finite dimensional version thereof. In dealing with partial differential equations and operator learning tasks, Sobolev spaces appear naturally. These Sobolev spaces are in most cases separable Banach spaces and carry a Schauder basis, which one can exploit to create finite dimensional networks from our infinite dimensional \mathcal{N}_d . Indeed, as basis functions contain structural information about the functions that go into the operator, one can use this information in the training. We provide some more details on this idea which is key to our method.

Suppose \mathfrak{X} is a *separable* Banach space, with Schauder basis denoted by $(e_k)_{k \in \mathbb{N}}$. Thus, for any $\xi \in \mathfrak{X}$ we have unique coefficients $a_k \in \mathbb{R}$, $k = 1, 2, \dots$ such that $\xi = \sum_{k=1}^{\infty} a_k e_k$. Without loss of generality, we can assume that $\|e_k\|_{\mathfrak{X}} = 1$ for all k . We define the canonical linear continuous projectors

$$p_k : \mathfrak{X} \rightarrow \mathbb{R}, \quad \xi \mapsto a_k, \quad k \in \mathbb{N},$$

and the projection operators

$$(5) \quad \Pi_N : \mathfrak{X} \rightarrow \text{span}\{e_1, \dots, e_N\}, \quad \xi \mapsto \sum_{k=1}^N p_k(\xi)e_k,$$

for $N \in \mathbb{N}$. The projection operators are also linear, bounded, and Π_N converges uniformly on compacts $\mathcal{K} \subset \mathfrak{X}$ when $N \rightarrow \infty$ (see (Schaefer 1971, Thm. 9.6, p. 115)).

Given a network \mathcal{N} as in (2) and $N \in \mathbb{N}$, define a finite dimensional network \mathcal{N}_N as

$$(6) \quad \mathcal{N}_N(\xi) = \sum_{j=1}^M (\ell_j \circ \Pi_N)(\sigma(\Pi_N A_j \Pi_N \xi + \Pi_N \beta_j)).$$

Let us briefly explain the reason why this architecture can be seen as a finite dimensional object: first of all, for any $N \in \mathbb{N}$, $\Pi_N \xi \in \mathfrak{X}$ can be identified with its truncated expansion $(p_1(\xi), \dots, p_N(\xi)) \in \mathbb{R}^N$. Secondly, the restrictions of the operators $\Pi_N \circ A_j$ and ℓ_j to $\text{span}\{e_1, \dots, e_N\}$ are finite dimensional, because the action of ℓ_j will be prescribed by the scalars $\ell_j(e_1), \dots, \ell_j(e_N)$, and the action of $\Pi_N \circ A_j \circ \Pi_N$ will be specified by the matrix $\{p_m(A_j e_k)\}_{m,k=1}^N$. The sum above thus resembles a classical neural network. However, instead of the typical one dimensional activation function, the function $\Pi_N \circ \sigma$ restricted to $\text{span}\{e_1, \dots, e_N\}$ is multidimensional. The terms appearing in the sum can now easily be programmed in a computer: we refer to Section 4 for further details.

Assuming additionally that the activation function σ is Lipschitz (which holds if $\tilde{\sigma}_j$ is Lipschitz for each $j \in \mathbb{N}$ and the Lipschitz constants satisfy a uniform bound $\sup_{j \in \mathbb{N}} \text{Lip}(\tilde{\sigma}_j) < \infty$: see (3)), then the Universal Approximation Theorem is valid for \mathcal{N}_N (see (Benth et al. 2023, Prop. 4.1)).

The networks \mathcal{N} and \mathcal{N}_N above can also be extended to deep neural networks. An infinite dimensional network with $n \in \mathbb{N}$ layers can be constructed from neurons of the form

$$(7) \quad \mathcal{N}_{\ell, \mathcal{A}}(\xi) := \ell(\sigma \circ \mathcal{A}_1 \circ \dots \circ \sigma \circ \mathcal{A}_n)(\xi)$$

with $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_n)$ a vector of affine transformation on \mathfrak{X} given as $\mathcal{A}_i(\xi) := A_i \xi + \beta_i$ for $i = 1, \dots, n$. Indeed, the span of such *deep neurons* is dense in the space of continuous operators on \mathfrak{X} , i.e., these deep infinite dimensional neural networks are universal approximators (see (Benth et al. 2023, Prop. 5.2) and (Benth et al. 2024, Prop. 2.10)). Notice that the universal approximation theorems do not provide any quantification on the number of neurons M nor the depth n in the approximating networks.

3. A CONTINUITY ANALYSIS OF THE NONLINEAR OPERATOR MAPS

In this section, we first define the solution operator for the parabolic Cauchy problem that we aim to solve. To achieve this, we embed the parameters of the Cauchy problem into a suitable Banach space of functions. Next, we analyze the continuity of the operator maps, which act as mappings from a set of parameter functions of interest to solutions. This continuity analysis justifies the use of the Fréchet neural network structures introduced in Section 2. Finally, we derive some robustness results for the single-point solution operator.

3.1. The parabolic Cauchy problem. We are going to follow the classical setup provided by Friedman (1975). We aim at solving the following backward parabolic Cauchy problem in \mathbb{R}^n

$$(8) \quad \begin{cases} Lu + \partial_t u = f(x, t), & \text{in } \mathbb{R}^n \times [0, T] \\ u(x, T) = \phi(x), & \text{in } \mathbb{R}^n, \end{cases}$$

where $0 < T < \infty$ and

$$(9) \quad Lu = \frac{1}{2} \sum_{i,j=1}^n a_{ij}(x, t) \partial_{ij}^2 u + \sum_{i=1}^n b_i(x, t) \partial_i u + c(x, t)u.$$

We are going to make the following assumptions on the coefficients of L , the final datum ϕ and the forcing term f .

Assumption 3.1. *The functions a_{ij}, b_i, c, ϕ and f satisfy:*

- (i) *There exists a number $\delta > 0$ such that $\sum_{i,j=1}^n a_{ij}(x, t) y_i y_j \geq \delta |y|^2$, for any $(x, t) \in \mathbb{R}^n \times [0, T], y \in \mathbb{R}^n$;*
- (ii) *a_{ij} and b_i are bounded in $\mathbb{R}^n \times [0, T]$ and Lipschitz continuous in (x, t) in compact subsets of $\mathbb{R}^n \times [0, T]$. The functions a_{ij} are Hölder continuous in x , uniformly with respect to $(x, t) \in \mathbb{R}^n \times [0, T]$;*

- (iii) c is bounded in $\mathbb{R}^n \times [0, T]$ and Hölder continuous in (x, t) in compact subsets of $\mathbb{R}^n \times [0, T]$;
- (iv) $f(x, t)$ is continuous in $\mathbb{R}^n \times [0, T]$, Hölder continuous in x uniformly with respect to $(x, t) \in \mathbb{R}^n \times [0, T]$ and $|f(x, t)| \leq \kappa(1 + |x|)^\gamma$ in $\mathbb{R}^n \times [0, T]$, $\phi(x)$ is continuous in \mathbb{R}^n and $|\phi(x)| \leq \kappa(1 + |x|)^\gamma$, where κ, γ are positive constants.

This set of assumptions ensures that (see (Friedman 1975, Thm 5.3, p. 148)) there exists a unique $u \in C(\mathbb{R}^n \times [0, T]) \cap C^{1,2}(\mathbb{R}^n \times [0, T])$ solution of (8) such that $|u(x, t)| \leq \kappa(1 + |x|)^\gamma$ for some $\kappa > 0$. We remark that κ is here and throughout a generic constant that is allowed to change according to the context. The solution may be represented by the Feynman-Kac formula

$$(10) \quad u(x, t) = \mathbb{E} \left[\phi(X_{x,t}(T)) \exp \left(\int_t^T c(X_{x,t}(s), s) ds \right) \right] \\ - \mathbb{E} \left[\int_t^T f(X_{x,t}(s), s) \exp \left(\int_t^s c(X_{x,t}(r), r) dr \right) ds \right],$$

for $(x, t) \in \mathbb{R}^n \times [0, T]$, where $t \leq s \leq T$, and

$$(11) \quad X_{x,t}(s) = x + \int_t^s b(X_{x,t}(r), r) dr + \int_t^s \eta(X_{x,t}(r), r) dW(r)$$

with $\eta\eta^* = a$. Here, W is an n -dimensional Brownian motion defined on a filtered complete probability space $(\Omega, (\mathcal{F}_t)_{t \in [0, T]}, \mathcal{F}, \mathbb{P})$, $\eta \in \mathbb{R}^{n \times n}$, and $\mathbb{E}[\cdot]$ is the expectation operator with respect to the probability measure \mathbb{P} . Moreover, by (Friedman 1975, p. 112), for any $R > 0, 0 \leq \tau \leq T$, if $|x|, |y| \leq R$, there exists $C_{R,T} > 0$ such that

$$(12) \quad \mathbb{E} \left[\sup_{\tau \leq s \leq T} |X_{x,\tau}(s) - X_{y,\tau}(s)|^2 \right] \leq C_{R,T} |x - y|^2.$$

3.2. Solution maps and continuity. Our goal is to learn the nonlinear operator map

$$(13) \quad (\phi, c, f) \xrightarrow{F^t} u(\cdot, t), \quad 0 \leq t < T$$

with the Fréchet neural network structures presented in Section 2. In order to do so, we first need a space for (ϕ, c, f) and $u(\cdot, t)$ to live in, and then show that the map $(\phi, c, f) \xrightarrow{F^t} u(\cdot, t)$ has the required continuity properties such that we can expect the neural network to approximate the map sufficiently well, in light of the Universal Approximation Theorem 2.2.

We first introduce a couple of spaces of real valued functions that we will need in the following. We refer the reader to Adams & Fournier (2003) for these definitions. For $j \in \mathbb{N}_0$, denote by $C^j(\mathbb{R}^n)$ the space of real valued functions on \mathbb{R}^n whose derivatives up the order j exist. Define

$$(14) \quad C^j(\overline{\mathbb{R}^n}) := \{v \in C^j(\mathbb{R}^n) : D^\alpha v \text{ is bounded and uniformly continuous, } 0 \leq |\alpha| \leq j\}$$

where clearly $D^\alpha v = \partial_{\alpha_1, \dots, \alpha_n} v$ with $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$. Observe that the notation $C^j(\overline{\mathbb{R}^n})$ arises from the fact that for more general spaces \mathcal{X} instead of \mathbb{R}^n , the uniform continuity allows for an extension to the closure $\overline{\mathcal{X}}$. In our case of course $\mathbb{R}^n = \overline{\mathbb{R}^n}$, but $C^j(\overline{\mathbb{R}^n}) \neq C^j(\mathbb{R}^n)$.

The vector space $C^j(\overline{\mathbb{R}^n})$ is a Banach space when endowed with the norm given by

$$\|v\|_{C^j(\overline{\mathbb{R}^n})} := \max_{0 \leq |\alpha| \leq j} \sup_{x \in \mathbb{R}^n} |D^\alpha v(x)|.$$

We further consider for $0 < \lambda \leq 1$ also the spaces $C^{j,\lambda}(\overline{\mathbb{R}^n}) \subset C^j(\overline{\mathbb{R}^n})$ defined by

$$(15) \quad C^{j,\lambda}(\overline{\mathbb{R}^n}) := \{v \in C^j(\overline{\mathbb{R}^n}) : D^\alpha v \text{ satisfies a } \lambda\text{-Hölder condition, } 0 \leq |\alpha| \leq j\}$$

i.e. for any $0 \leq |\alpha| \leq j$ there exists a constant $K \geq 0$ such that

$$|D^\alpha v(x) - D^\alpha v(y)| \leq K |x - y|^\lambda, \quad x, y \in \mathbb{R}^n.$$

The space $C^{j,\lambda}(\overline{\mathbb{R}^n})$ is a Banach space when endowed with the norm given by

$$\|v\|_{C^{j,\lambda}(\overline{\mathbb{R}^n})} := \|v\|_{C^j(\overline{\mathbb{R}^n})} + \max_{0 \leq |\alpha| \leq j} \sup_{x \neq y} \frac{|D^\alpha v(x) - D^\alpha v(y)|}{|x - y|^\lambda}.$$

For brevity reasons we write $C^{j,\lambda}$ rather than $C^{j,\lambda}(\overline{\mathbb{R}^n})$. We also recall the definition of the Sobolev space $W^{k,p} = W^{k,p}(\mathbb{R}^n)$, $k \in \mathbb{N}, 1 \leq p < \infty$: this is the space of all functions $v \in L^p(\mathbb{R}^n)$ such that all their distributional derivatives $D^\alpha v$, $0 < |\alpha| \leq k$ are functions in $L^p(\mathbb{R}^n)$. A natural norm for these spaces is provided by

$$\|v\|_{W^{k,p}} := \left(\sum_{0 \leq |\alpha| \leq k} \|D^\alpha v\|_{L^p}^p \right)^{1/p}.$$

In this way, the spaces become Banach spaces and for $p = 2$ Hilbert spaces. These spaces enjoy the following embedding results, known as Sobolev-Morrey embedding: let $j, m \in \mathbb{N}_0$ and $1 \leq p < \infty$. By part III of Theorem 5.4 and Remark 5.5 (3) (Adams & Fournier 2003, page 98) the following holds:

- If $mp > n > (m-1)p$, then

$$(16) \quad W^{j+m,p} \hookrightarrow C^{j,\lambda}, \quad 0 < \lambda \leq m - n/p$$

with continuity.

- If $n = (m-1)p$, then

$$(17) \quad W^{j+m,p} \hookrightarrow C^{j,\lambda}, \quad 0 < \lambda < 1$$

with continuity. If $n = m-1$ and $p = 1$, then λ can be also equal to 1 in the last equation.

This result has the following consequence, which we are going to use later: by the continuity of the embedding, we have

$$\|v\|_{C^{j,\lambda}} \leq C_{Sob} \|v\|_{W^{j+m,p}}, \quad v \in W^{j+m,p}$$

for some universal constant $C_{Sob} = C_{Sob}(n, j, m, p)$, and so we deduce in particular that

$$(18) \quad \max_{0 \leq |\alpha| \leq j} \sup_{x \neq y} \frac{|D^\alpha v(x) - D^\alpha v(y)|}{|x - y|^\lambda} \leq C_{Sob} \|v\|_{W^{j+m,p}}.$$

From now on, we will assume $j = 0$ and that m and p satisfy one of the two conditions above. Thus, $W^{m,p}$ embeds continuously in $C^{0,\lambda}$ for suitable λ . In this case, (18) becomes

$$(19) \quad |v(x) - v(y)| \leq C_{Sob} \|v\|_{W^{m,p}} |x - y|^\lambda, \quad x, y \in \mathbb{R}^n, v \in W^{m,p}.$$

Remark 3.2. *If we choose $m = 1$, then we need $n < p$ and thus $0 < \lambda < 1 - n/p$.*

Going back to (8), we suppose the following to hold:

Assumption 3.3. *For the functions ϕ, c and f ,*

- $\phi \in W^{m,p}$
- c and f are time-independent and $c, f \in W^{m,p}$.

Under this assumption we can conclude from the embeddings that $\phi, c, f \in C^{0,\lambda}$. In view of these embeddings, there exists a unique solution u of (8). We fix once for all $0 \leq t < T$, and define

$$(20) \quad F^t : W^{m,p} \times W^{m,p} \times W^{m,p} \rightarrow BC(\mathbb{R}^n), \quad (\phi, c, f) \xrightarrow{F^t} u(\cdot, t),$$

where $BC(\mathbb{R}^n)$ denotes the space of bounded continuous functions from \mathbb{R}^n to \mathbb{R} . Observe that indeed $u \in BC(\mathbb{R}^n \times [0, T])$, because ϕ, c and f are bounded and boundedness of u then follows from (10).

We have the following continuity result for the operator F^t in (20) which paves the way for using our neural network in infinite dimension for learning.

Proposition 3.4. *Let $\mathfrak{X} := W^{m,p} \times W^{m,p} \times W^{m,p}$ with m and p satisfying the conditions above. Endow \mathfrak{X} with the natural norm*

$$\|(v_1, v_2, v_3)\|_{\mathfrak{X}} = \|v_1\|_{W^{m,p}} + \|v_2\|_{W^{m,p}} + \|v_3\|_{W^{m,p}}, \quad (v_1, v_2, v_3) \in \mathfrak{X}.$$

Let $0 \leq t < T$. Then, the solution operator F^t defined by (20) is continuous from \mathfrak{X} into $BC(\mathbb{R}^n)$.

Proof. Let

$$\phi_k \xrightarrow{W^{m,p}} \phi, \quad c_k \xrightarrow{W^{m,p}} c, \quad f_k \xrightarrow{W^{m,p}} f$$

as $k \rightarrow \infty$, and let u_k be the corresponding solution of (8). We observe the following elementary facts:

- (1) Since $\phi_k \rightarrow \phi$ uniformly on \mathbb{R}^n by the embedding into $C^{0,\lambda}$, we have $\phi_k(X_{x,t}(T)) \rightarrow \phi(X_{x,t}(T))$ uniformly in $(\omega, x) \in \Omega \times \mathbb{R}^n$ as $k \rightarrow \infty$.
- (2) Similarly, since $c_k \rightarrow c$ uniformly on \mathbb{R}^n , we have $c_k(X_{x,t}(s)) \rightarrow c(X_{x,t}(s))$ uniformly in $(\omega, x, s) \in \Omega \times \mathbb{R}^n \times [t, T]$ as $k \rightarrow \infty$. Therefore,

$$\int_t^T c_k(X_{x,t}(s)) ds \rightarrow \int_t^T c(X_{x,t}(s)) ds$$

uniformly in $(\omega, x) \in \Omega \times \mathbb{R}^n$.

- (3) Using the mean value theorem and the fact that the quantities $\int_t^T c_k(X_{x,t}(s)) ds$ and $\int_t^T c(X_{x,t}(s)) ds$ are bounded, it is immediate to see that

$$\exp \left\{ \int_t^T c_k(X_{x,t}(s)) ds \right\} \rightarrow \exp \left\{ \int_t^T c(X_{x,t}(s)) ds \right\}$$

uniformly in $(\omega, x) \in \Omega \times \mathbb{R}^n$ as $k \rightarrow \infty$.

- (4) Mutatis mutandis, $\int_t^s c_k(X_{x,t}(r)) dr \rightarrow \int_t^s c(X_{x,t}(r)) dr$ uniformly in $(\omega, x, s) \in \Omega \times \mathbb{R}^n \times [t, T]$ and once more

$$\exp \left\{ \int_t^s c_k(X_{x,t}(r)) dr \right\} \rightarrow \exp \left\{ \int_t^s c(X_{x,t}(r)) dr \right\}$$

uniformly in $(\omega, x, s) \in \Omega \times \mathbb{R}^n \times [t, T]$ as $k \rightarrow \infty$.

- (5) $f_k \rightarrow f$ uniformly on \mathbb{R}^n implies $f_k(X_{x,t}(s)) \rightarrow f(X_{x,t}(s))$ uniformly in $(\omega, x, s) \in \Omega \times \mathbb{R}^n \times [t, T]$.

In view of this, we deduce that $\sup_{\mathbb{R}^n} |u_k(x, t) - u(x, t)| \rightarrow 0$ as $k \rightarrow \infty$. We have proved the proposition. \square

We observe that the space \mathfrak{X} defined in Proposition 3.4 above is a separable Banach space, under the natural linear structure of product of spaces. It can be easily supplemented by a Schauder basis, coming from a Schauder basis of $W^{m,p}$ (see e.g. (Heil 2011, Ch. 4) for a general introduction to basis functions in Banach spaces). Indeed, from Triebel (2004) we know that these Sobolev spaces $W^{m,p}$ carry an unconditional Schauder basis given by wavelets. We refer to (Heil 2011, Ch. 12) and Meyer (2009) for more on wavelets. In Section 4 we present a numerical example where we construct a basis instead in terms of Hermite functions for the Hilbert space $W^{1,2}$.

We finally observe that for arbitrary $z \in \mathbb{R}^n$, the Dirac mass δ_z is trivially an element of the topological dual of $BC(\mathbb{R}^n)$. In view of all of this, we conclude that the map $\langle \delta_z, F^t \rangle$ is an element of $C(\mathfrak{X})$. Let us show how we can utilize this to learn the solution map uniformly on small compact subsets of \mathbb{R}^n (see also Corollary 3.7 below).

Proposition 3.5. *Fix $0 \leq t < T$, $\mathcal{K} \subset \mathfrak{X}$ compact and $R > 0$. Let λ be the Hölder constant provided by equation (19). Then there exists a constant $\Gamma = \Gamma(\mathcal{K}, T, R) > 0$ such that*

$$|\langle \delta_x - \delta_y, F^t(\phi, c, f) \rangle| \leq \Gamma(\mathcal{K}, T, R) |x - y|^\lambda$$

for $(\phi, c, f) \in \mathcal{K}$ and $|x| \leq R, |y| \leq R$.

Proof. In the following computations we are going to use repeatedly the Hölder condition in (19) as well as

$$\|v\|_\infty \leq C_{Sob} \|v\|_{W^{m,p}}.$$

We define for convenience the map I ,

$$I : W^{m,p} \rightarrow \mathbb{R}, \quad v \mapsto \exp\{(T-t)\|v\|_\infty\}.$$

We observe that it is a continuous map on $W^{m,p}$, because of the continuity of the embedding and by composition of continuous maps, namely

$$W^{m,p} \ni v \mapsto v \in C^{0,\lambda} \mapsto \|v\|_\infty \mapsto I(v).$$

Let $x, y \in \mathbb{R}^n$, and set $u(\cdot, t) = F^t(\phi, c, f)$. Then, using Feynman-Kac formula (10), and several times the mean value theorem for the terms involving the exponential function, we obtain

$$\begin{aligned}
|u(x, t) - u(y, t)| &\leq \mathbb{E} |\phi(X_{x,t}(T)) - \phi(X_{y,t}(T))| \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] \\
&\quad + \mathbb{E} |\phi(X_{y,t}(T))| \left| \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] - \exp \left[\int_t^T c(X_{y,t}(s)) ds \right] \right| \\
&\quad + \mathbb{E} \int_t^T |f(X_{x,t}(s)) - f(X_{y,t}(s))| \exp \left[\int_t^s c(X_{x,t}(r)) dr \right] ds \\
&\quad + \mathbb{E} \int_t^T |f(X_{y,t}(s))| \left| \exp \left[\int_t^s c(X_{x,t}(r)) dr \right] - \exp \left[\int_t^s c(X_{y,t}(r)) dr \right] \right| ds \\
&\leq C_{Sob} \mathbb{E} \|\phi\|_{W^{m,p}} I(c) |X_{x,t}(T) - X_{y,t}(T)|^\lambda \\
&\quad + \mathbb{E} \|\phi\|_\infty I(c) \left| \int_t^T [c(X_{x,t}(s)) - c(X_{y,t}(s))] ds \right| \\
&\quad + C_{Sob} \mathbb{E} \int_t^T \|f\|_{W^{m,p}} I(c) |X_{x,t}(s) - X_{y,t}(s)|^\lambda ds \\
&\quad + \mathbb{E} \int_t^T \|f\|_\infty I(c) \left| \int_t^s [c(X_{x,t}(r)) - c(X_{y,t}(r))] dr \right| ds.
\end{aligned}$$

Thus, by applying Hölder's inequality repeatedly ($2/\lambda \geq 1$),

$$\begin{aligned}
|u(x, t) - u(y, t)| &\leq C_{Sob} \|\phi\|_{W^{m,p}} I(c) \left[\mathbb{E} |X_{x,t}(T) - X_{y,t}(T)|^2 \right]^{\lambda/2} \\
&\quad + \|\phi\|_\infty I(c) C_{Sob} \|c\|_{W^{m,p}} \mathbb{E} \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^\lambda ds \\
&\quad + C_{Sob} \|f\|_{W^{m,p}} I(c) \mathbb{E} \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^\lambda ds \\
&\quad + C_{Sob} \|f\|_\infty I(c) \|c\|_{W^{m,p}} \mathbb{E} \int_t^T \int_t^s |X_{x,t}(r) - X_{y,t}(r)|^\lambda dr ds \\
&\leq C_{Sob} \|\phi\|_{W^{m,p}} I(c) \left[\mathbb{E} |X_{x,t}(T) - X_{y,t}(T)|^2 \right]^{\lambda/2} \\
&\quad + \|\phi\|_{W^{m,p}} I(c) C_{Sob}^2 \|c\|_{W^{m,p}} \left[\mathbb{E} \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^2 ds \right]^{\lambda/2} (T-t)^{1-\lambda/2} \\
&\quad + C_{Sob} \|f\|_{W^{m,p}} I(c) \left[\mathbb{E} \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^2 ds \right]^{\lambda/2} (T-t)^{1-\lambda/2} \\
&\quad + C_{Sob}^2 \|f\|_{W^{m,p}} I(c) \|c\|_{W^{m,p}} \left[\mathbb{E} \int_t^T \int_t^s |X_{x,t}(r) - X_{y,t}(r)|^2 dr ds \right]^{\lambda/2} \left[\frac{(T-t)^2}{2} \right]^{1-\lambda/2}.
\end{aligned}$$

Let $x, y : |x| \leq R, |y| \leq R$ for some fixed $R > 0$. Using (12) we infer

$$\begin{aligned}
|u(x, t) - u(y, t)| &\leq C_{Sob} \|\phi\|_{W^{m,p}} I(c) C_{R,T}^{\lambda/2} |x - y|^\lambda \\
&\quad + C_{Sob}^2 \|\phi\|_{W^{m,p}} I(c) \|c\|_{W^{m,p}} C_{R,T}^{\lambda/2} (T-t) |x - y|^\lambda \\
&\quad + C_{Sob} \|f\|_{W^{m,p}} I(c) C_{R,T}^{\lambda/2} (T-t) |x - y|^\lambda \\
&\quad + C_{Sob}^2 \|f\|_{W^{m,p}} I(c) \|c\|_{W^{m,p}} C_{R,T}^{\lambda/2} \frac{(T-t)^2}{2} |x - y|^\lambda.
\end{aligned}$$

Therefore, by continuity and Weierstrass theorem, we conclude that there exists a constant $\Gamma = \Gamma(\mathcal{K}, T, R) > 0$ such that

$$|u(x, t) - u(y, t)| \leq \Gamma(\mathcal{K}, T, R) |x - y|^\lambda, \quad (\phi, c, f) \in \mathcal{K}, |x| \leq R, |y| \leq R,$$

i.e.

$$|F^t(\phi, c, f)(x) - F^t(\phi, c, f)(y)| \leq \Gamma(\mathcal{K}, T, R) |x - y|^\lambda, \quad (\phi, c, f) \in \mathcal{K}, |x| \leq R, |y| \leq R.$$

We have proved the proposition. \square

Remark 3.6. *We remark that the result in Proposition 3.5 would hold for subsets $\mathcal{K} \subset \mathfrak{X}$ being bounded only and not necessarily being compact. However, since the Universal Approximation Theorem 2.2 in any case requires working on compact sets, we have formulated the proposition accordingly.*

We have the following important consequence, which in broad strokes tells us that, as long as we stay close to x and willing to accept a slightly higher error, we do not need to change the approximating neural network architecture.

Corollary 3.7. *Fix $0 \leq t < T$, $\mathcal{K} \subset \mathfrak{X}$ compact and $R > 0$. Let $x : |x| \leq R$. Let $\varepsilon > 0$ be arbitrary. Suppose to be given for some $N \in \mathbb{N}$*

$$\mathcal{N}^N = \sum_{j=1}^N \mathcal{N}_{\ell_j, A_j, \beta_j}$$

with $\ell_j \in \mathfrak{X}'$, $A_j \in \mathcal{L}(\mathfrak{X})$ and $\beta_j \in \mathfrak{X}$ such that

$$\sup_{(\phi, c, f) \in \mathcal{K}} |\mathcal{N}^N(\phi, c, f) - \langle \delta_x, F^t(\phi, c, f) \rangle| < \varepsilon.$$

Fix $\varepsilon' > 0$ and set $r = \left(\frac{\varepsilon'}{\Gamma(\mathcal{K}, T, R)}\right)^{1/\lambda}$. Then for any $y \in B_r(x)$, $|y| \leq R$, it holds

$$\sup_{(\phi, c, f) \in \mathcal{K}} |\mathcal{N}^N(\phi, c, f) - \langle \delta_y, F^t(\phi, c, f) \rangle| < \varepsilon + \varepsilon'.$$

Proof. From the previous proposition, we indeed have for any $y \in B_r(x)$, $|y| \leq R$

$$|\langle \delta_x - \delta_y, F^t(\phi, c, f) \rangle| < \varepsilon'$$

for any $(\phi, c, f) \in \mathcal{K}$, and therefore

$$\sup_{(\phi, c, f) \in \mathcal{K}} |\langle \delta_x, F^t(\phi, c, f) \rangle - \langle \delta_y, F^t(\phi, c, f) \rangle| < \varepsilon'.$$

By the triangle inequality we get the claim. \square

Needless to say, everything said until here still holds for possibly different solution operators F^t where some of the “variables” defining the parabolic Cauchy problem are fixed, i.e., given exogenously. However, by relaxing the properties of these fixed variables, we can allow for more flexible specifications but still preserve the continuity of the solution operator. We investigate this next in the case where $f = 0$ and the final datum ϕ is not necessarily in $W^{m,p}$ but is continuous with some polynomial growth (i.e., still satisfying the standard assumptions ensuring well-posedness of (8)). Namely, we consider solutions of this kind:

$$u(x, t) = \mathbb{E} \left[\phi(X_{x,t}(T)) \exp \left(\int_t^T c(X_{x,t}(s)) ds \right) \right], \quad (x, t) \in \mathbb{R}^n \times [0, T]$$

with $c \in W^{m,p}$ and ϕ continuous in \mathbb{R}^n such that $|\phi(x)| \leq \kappa(1 + |x|)^\gamma$, where κ, γ are positive constants. From the general theory in Subsection 3.1 we know that $|u(x, t)| \leq \kappa(1 + |x|)^\gamma$ for some $\kappa > 0$ (possibly different to the above), and thus in general the solution will be unbounded. To overcome this issue, we simply restrict ourselves to a fixed compact subset $K \subset \mathbb{R}^n$, and we will learn the solution here. More precisely, for $0 \leq t < T$, $K \subset \mathbb{R}^n$ compact and ϕ as above we define the following solution operator

$$(21) \quad F^{t,K,\phi} : W^{m,p} \rightarrow C(K), \quad c \mapsto u(\cdot, t) \Big|_K$$

namely the solution of (8) as a function of c restricted to K with $f = 0$, final datum ϕ given. Also in this setting we obtain:

Proposition 3.8. *Let ϕ be continuous in \mathbb{R}^n and such that $|\phi(x)| \leq \kappa(1 + |x|)^\gamma$, where κ, γ are positive constants. Assume $c \in W^{m,p}$, $0 \leq t < T$ and let $K \subset \mathbb{R}^n$ be compact. Then the solution operator $F^{t,K,\phi}$ of the parabolic Cauchy problem*

$$\begin{cases} Lu + \partial_t u = 0, & \text{in } \mathbb{R}^n \times [0, T) \\ u(x, T) = \phi(x), & \text{in } \mathbb{R}^n, \end{cases}$$

is continuous from $W^{m,p}$ into $C(K)$.

If ϕ is assumed additionally to be Hölder continuous with exponent $0 < \tilde{\lambda} \leq 1$, the following holds: given a compact subset $\mathcal{K} \subset W^{m,p}$, there is $\Gamma = \Gamma(\phi, \tilde{\lambda}, T, a_{ij}, b_i, K, \mathcal{K}) > 0$ constant such that

$$|\langle \delta_x - \delta_y, F^{t,K,\phi}(c) \rangle| \leq \Gamma |x - y|^{\tilde{\lambda}}, \quad c \in \mathcal{K}, x, y \in K.$$

Proof. Let us show continuity first. Given $c_k \xrightarrow{W^{m,p}} c$, we know from before that

$$\exp \left\{ \int_t^T c_k(X_{x,t}(s)) ds \right\} \rightarrow \exp \left\{ \int_t^T c(X_{x,t}(s)) ds \right\}$$

uniformly on $\Omega \times K$. Furthermore, from standard SDEs theory (see for instance (Friedman 1975, Thm 2.3 page 107)), we know that for any $h \in \mathbb{N}$

$$\mathbb{E} |X_{x,t}(s)|^h \leq (2 + |x|^h) e^{Cs}, \quad t \leq s \leq T, x \in \mathbb{R}^n$$

where $C = C(h, a_{ij}, b_i, T)$. Thus, in view of the bound satisfied by ϕ , we easily get

$$|\phi(X_{x,t}(T))| \lesssim_\phi (1 + |X_{x,t}(T)|^h), \quad x \in K$$

for some $h = h(\phi) \in \mathbb{N}$, and hence

$$\mathbb{E} |\phi(X_{x,t}(T))| \leq C(\phi, T, a_{ij}, b_i, K)$$

uniformly in $x \in K$. From this we infer

$$\begin{aligned} |u_k(x, t) - u(x, t)| &\leq \mathbb{E} |\phi(X_{x,t}(T))| \left| \exp \left[\int_t^T c_k(X_{x,t}(s)) ds \right] - \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] \right| \\ &\leq \mathbb{E} |\phi(X_{x,t}(T))| \sup_{(\omega, x) \in \Omega \times K} \left| \exp \left[\int_t^T c_k(X_{x,t}(s)) ds \right] - \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] \right| \\ &\leq C(\phi, T, a_{ij}, b_i, K) \sup_{(\omega, x) \in \Omega \times K} \left| \exp \left[\int_t^T c_m(X_{x,t}(s)) ds \right] - \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] \right| \end{aligned}$$

uniformly in $x \in K$. Therefore, $\sup_{x \in K} |u_k(x, t) - u(x, t)| \rightarrow 0$ as $k \rightarrow \infty$, namely $F^{t,K,\phi}$ is continuous on $W^{m,p}$ into $C(K)$.

Let us now assume additionally that ϕ is Hölder continuous for some exponent $0 < \tilde{\lambda} \leq 1$. Arguing as above in the proof of Proposition 3.5, for $x, y \in K$ we now have

$$\begin{aligned}
|u(x, t) - u(y, t)| &\leq \mathbb{E} |\phi(X_{x,t}(T)) - \phi(X_{y,t}(T))| \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] \\
&\quad + \mathbb{E} |\phi(X_{y,t}(T))| \left| \exp \left[\int_t^T c(X_{x,t}(s)) ds \right] - \exp \left[\int_t^T c(X_{y,t}(s)) ds \right] \right| \\
&\leq C_\phi I(c) \mathbb{E} |X_{x,t}(T) - X_{y,t}(T)|^{\tilde{\lambda}} \\
&\quad + I(c) \mathbb{E} |\phi(X_{y,t}(T))| \left| \int_t^T [c(X_{x,t}(s)) - c(X_{y,t}(s))] ds \right| \\
&\leq C_\phi I(c) \left[\mathbb{E} |X_{x,t}(T) - X_{y,t}(T)|^2 \right]^{\tilde{\lambda}/2} \\
&\quad + I(c) C_{Sob} \|c\|_{W^{m,p}} \mathbb{E} |\phi(X_{y,t}(T))| \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^{\tilde{\lambda}} ds.
\end{aligned}$$

By Hölder's and Jensen's inequalities we infer

$$\begin{aligned}
\mathbb{E} |\phi(X_{y,t}(T))| \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^{\tilde{\lambda}} ds &\leq \left[\mathbb{E} |\phi(X_{y,t}(T))|^{2/(2-\tilde{\lambda})} \right]^{1-\tilde{\lambda}/2} \times \\
&\quad \times \left[\mathbb{E} \left[\int_t^T |X_{x,t}(s) - X_{y,t}(s)|^{\tilde{\lambda}} ds \right]^{2/\tilde{\lambda}} \right]^{\tilde{\lambda}/2} \\
&\leq \left[\mathbb{E} |\phi(X_{y,t}(T))|^{2/(2-\tilde{\lambda})} \right]^{1-\tilde{\lambda}/2} \times \\
&\quad \times (T-t)^{1-\tilde{\lambda}/2} \left[\mathbb{E} \int_t^T |X_{x,t}(s) - X_{y,t}(s)|^2 ds \right]^{\tilde{\lambda}/2}.
\end{aligned}$$

From above, we deduce

$$|\phi(X_{y,t}(T))|^{2/(2-\tilde{\lambda})} \lesssim_{\phi, \tilde{\lambda}} (1 + |X_{y,t}(T)|^{2h/(2-\tilde{\lambda})}), \quad y \in K$$

for some $h = h(\phi) \in \mathbb{N}$, and hence, with a different constant,

$$\mathbb{E} |\phi(X_{y,t}(T))|^{2/(2-\tilde{\lambda})} \leq C(\phi, \tilde{\lambda}, T, a_{ij}, b_i, K)$$

uniformly in $y \in K$. Similarly to above, we then obtain

$$\begin{aligned}
|u(x, t) - u(y, t)| &\leq C_\phi I(c) C_{K,T}^{\tilde{\lambda}/2} |x - y|^{\tilde{\lambda}} + \\
&\quad + C_{Sob} I(c) \|c\|_{W^{m,p}} C(\phi, \tilde{\lambda}, T, a_{ij}, b_i, K) C_{H,T}^{\tilde{\lambda}/2} (T-t) |x - y|^{\tilde{\lambda}},
\end{aligned}$$

for $x, y \in K$. Let $\mathcal{K} \subset W^{m,p}$ be a fixed compact subset. By continuity and the Weierstrass Theorem, we conclude that there exists a constant $\Gamma = \Gamma(\phi, \tilde{\lambda}, T, a_{ij}, b_i, K, \mathcal{K}) > 0$ such that

$$|u(x, t) - u(y, t)| \leq \Gamma |x - y|^{\tilde{\lambda}}, \quad c \in \mathcal{K}, x, y \in K,$$

i.e.,

$$|F^{t,K,\phi}(c)(x) - F^{t,K,\phi}(c)(y)| \leq \Gamma |x - y|^{\tilde{\lambda}}, \quad c \in \mathcal{K}, x, y \in K,$$

and the claim follows. \square

As an immediate consequence, we effortlessly obtain the analogous of Corollary 3.7:

Corollary 3.9. *Assume the setting of Proposition 3.8. Let $x \in K$ and $\varepsilon > 0$ be arbitrary. Suppose for $N \in \mathbb{N}$ to be given*

$$\mathcal{N}^N = \sum_{j=1}^N \mathcal{N}_{\ell_j, A_j, \beta_j}$$

with $\ell_j \in (W^{m,p})'$, $A_j \in \mathcal{L}(W^{m,p})$ and $\beta_j \in W^{m,p}$ such that

$$\sup_{c \in \mathcal{K}} |\mathcal{N}^N(c) - \langle \delta_x, F^{t,K,\phi}(c) \rangle| < \varepsilon.$$

Fix $\varepsilon' > 0$ and set $r = \left(\frac{\varepsilon'}{\Gamma}\right)^{1/\bar{\lambda}}$. Then for any $y \in K \cap B_r(x)$, it holds

$$\sup_{c \in \mathcal{K}} |(\mathcal{N}^N(c) - \langle \delta_y, F^{t,K,\phi}(c) \rangle)| < \varepsilon + \varepsilon'.$$

After these theoretical considerations on continuity, verifying the use of the Universal Approximation Theorem for operator-learning, we proceed in the next section with a numerical case study.

4. A NUMERICAL CASE STUDY

We demonstrate our proposed methodology by considering a particular case of the Cauchy problem set on \mathbb{R} and train a neural network to learn the operator mapping the function c into the u solution evaluated in a location. In our proof-of-concept study, we benchmark with respect to the DeepONet approach.

For our purposes, we need to have a set of orthonormal basis functions in $W^{1,2}$. These provide us with structural information that we exploit in the training. Here we propose to construct such a basis from the Hermite functions, which is an orthonormal basis of $L^2(\mathbb{R})$.

4.1. Basis functions. Following e.g. (Schwartz 1950, p. 261), we define the 1-d Hermite polynomials by

$$(22) \quad H_m(x) = (-1)^m 2^{1/4-m} (m!)^{-1/2} \pi^{-m/2} \exp(2\pi x^2) \frac{d^m}{dx^m} \exp(-2\pi x^2)$$

and the associated Hermite functions

$$(23) \quad \mathcal{H}_m(x) = H_m(x) \exp(-\pi x^2).$$

where $x \in \mathbb{R}$, $m \in \mathbb{N}_0$. Then $(\mathcal{H}_m)_{m \in \mathbb{N}_0}$ is an orthonormal system in $L^2(\mathbb{R})$. We first derive $\langle \mathcal{H}_m, \mathcal{H}_n \rangle_{W^{1,2}}$ which we need in order to obtain an orthonormal set of vectors in $W^{1,2}$.

Proposition 4.1. *The following holds:*

(1) for $m, n \in \mathbb{N}$

$$\int_{\mathbb{R}} \mathcal{H}'_m(x) \mathcal{H}'_n(x) dx = \pi(2m+1) \delta_{m,n} - \pi \sqrt{m(m-1)} \delta_{m,n+2} - \pi \sqrt{(m+1)(m+2)} \delta_{m,n-2},$$

(2) for $m \in \mathbb{N}$ and $n = 0$

$$\int_{\mathbb{R}} \mathcal{H}'_m(x) \mathcal{H}'_0(x) dx = -\pi \sqrt{2} \delta_{m,2},$$

(3) and for $m = n = 0$

$$\int_{\mathbb{R}} \mathcal{H}'_0(x) \mathcal{H}'_0(x) dx = \pi.$$

Proof. It follows from (Schwartz 1950, VII, 7; 30) (or by direct computation) that

$$-\mathcal{H}'_m(x) + 2\pi x \mathcal{H}_m(x) = 2\sqrt{\pi(m+1)} \mathcal{H}_{m+1}(x), \quad m \in \mathbb{N}_0$$

and

$$-\mathcal{H}'_m(x) - 2\pi x \mathcal{H}_m(x) = -2\sqrt{\pi m} \mathcal{H}_{m-1}(x), \quad m \in \mathbb{N}.$$

By summing up these two equations, one obtains the recursion

$$\mathcal{H}'_m(x) = \sqrt{\pi m} \mathcal{H}_{m-1}(x) - \sqrt{\pi(m+1)} \mathcal{H}_{m+1}(x), \quad m \in \mathbb{N}.$$

Hence, after appealing to the orthogonality of $(\mathcal{H}_m)_{m \in \mathbb{N}_0}$, we get

$$\begin{aligned} & \int_{\mathbb{R}} \mathcal{H}'_m(x) \mathcal{H}'_n(x) dx \\ &= \pi \left(\sqrt{mn} + \sqrt{(m+1)(n+1)} \right) \delta_{m,n} - \pi \sqrt{m(n+1)} \delta_{m,n+2} - \pi \sqrt{(m+1)n} \delta_{m,n-2} \end{aligned}$$

which proves (1). For $m = 0$ one obtains directly from the definition of \mathcal{H}_0 that

$$\mathcal{H}'_0(x) = -2\pi x \mathcal{H}_0(x).$$

From the definition of H_0 and \mathcal{H}_0 it follows that $H_0(x) = 2^{1/4}$ and $H_1(x) = 2^{5/4}\sqrt{\pi}x$. Now we can re-write $\mathcal{H}'_0(x) = -2\pi x \mathcal{H}_0(x)$ as

$$\mathcal{H}'_0(x) = -2\pi x 2^{1/4} \exp(-\pi x^2) = -\sqrt{\pi} \mathcal{H}_1(x)$$

because $\mathcal{H}_1(x) = 2^{1/4} 2\sqrt{\pi}x \exp(-\pi x^2)$. With this last observation the case (2) with $m = 1$ and case (3) follows. \square

To this end, we write the conclusions of the last proposition in a more concise form, resulting in the following ‘‘multiplication table’’: for $m, n \in \mathbb{N}_0$ it holds:

- $\langle \mathcal{H}_m, \mathcal{H}_n \rangle_{W^{1,2}} = 1 + \pi(2m + 1)$ for $m = n$;
- $\langle \mathcal{H}_m, \mathcal{H}_n \rangle_{W^{1,2}} = -\pi\sqrt{\ell(\ell - 1)}$ for $|m - n| = 2$ with $\ell = \max\{m, n\}$;
- $\langle \mathcal{H}_m, \mathcal{H}_n \rangle_{W^{1,2}} = 0$ otherwise.

With this and the fact that

$$\int_{\mathbb{R}} \mathcal{H}_m(x) \mathcal{H}_n(x) dx = \delta_{m,n},$$

we can now apply the Gram–Schmidt procedure to the vectors $\mathcal{H}_0, \mathcal{H}_1, \dots$ to obtain an orthonormal basis in $W^{1,2}$, which we denote by $(e_k)_{k \in \mathbb{N}_0}$. We remark in passing that we can build basis functions in $W^{1,2}$ for $d > 1$ by tensorising the above Hermite basis.

4.2. Numerical example. We consider the parabolic Cauchy problem in \mathbb{R}

$$(24) \quad \begin{cases} Lu + \partial_t u = 0, & \text{in } \mathbb{R} \times [0, T) \\ u(x, T) = \phi(x), & \text{in } \mathbb{R}, \end{cases}$$

where $0 < T < \infty$ and

$$(25) \quad Lu = \frac{1}{2} \partial_x^2 u + c(x)u.$$

To simplify, we have set $a = 1$, $b = 0$ and the forcing term $f = 0$ in this numerical example. For the final datum ϕ , we choose $\phi(x) = x^2$. With these specifications, we aim for learning the non-linear operator mapping

$$(26) \quad c \mapsto u(\cdot, t).$$

where, by (10) and (11) u is given by

$$(27) \quad u(x, t) = \mathbb{E} \left[\phi(X_{x,t}(T)) \exp \left(\int_t^T c(X_{x,t}(s)) ds \right) \right]$$

and

$$(28) \quad X_{x,t}(s) = x + \int_t^s dW(r).$$

For fixed $x \in \mathbb{R}$, we fit a neural network as introduced in Section 2 to learn the map $c \mapsto u(x, t)$ on a compact subset $\mathcal{K} \subset W^{m,p}$. In our numerics, we let $t = 0$ and $T = 1$.

Instead of using data points $\{(c_i, u_i(x, t))\}_{i=1}^{M_{\text{train}}}$ to train the neural network, we instead fit a neural network to minimize the energy functional

$$(29) \quad g \mapsto \mathbb{E} \left[\int_{W^{1,2}} |\mathcal{X}(c) - g(c)|^2 \mu(dc) \right],$$

with $\mathcal{X}(c)$ given by

$$(30) \quad \mathcal{X}(c) := \phi(X_{x,t}(T)) \exp \left(\int_0^1 c(X_{x,t}(s)) ds \right),$$

and where μ is a measure on \mathcal{K} . This approach allows us to appeal to the Uniform Approximation Theorem since we have continuity of the map $c \mapsto u(x, t)$. We refer to (Beck et al. 2021, Prop. 2.2) where this approach has been used first in the finite dimensional case, and (Benth et al. 2024, Lem. 5.4) for its extension to infinite dimensional spaces.

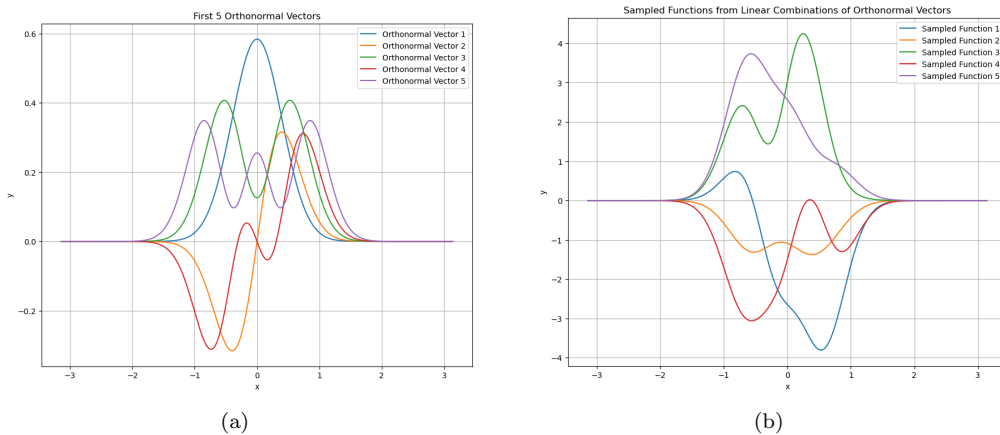


FIGURE 1. First 5 basis functions e_1, \dots, e_5 (left) and 5 random samples of c from \mathcal{K} (right)

Recall from above the basis $(e_k)_{k \in \mathbb{N}_0}$ of Hermite functions for $W^{1,2}$. We choose the compact set $\mathcal{K} \subset W^{1,2}$ by

$$\mathcal{K} := \left\{ c \in W^{1,2}; c = \sum_{k=1}^5 a_k e_k, a_k \in [-5, 5] \right\}.$$

We specify a uniform measure μ on \mathcal{K} canonically from the classical uniform measure on $[-5, 5]^5$, and we trivially extend it to the whole space. In Figure 1 we show the first 5 basis vectors and 5 random samples from \mathcal{K} (Note that the multiple occurrences of 5 is a coincident and not intentional). We fit a Fréchet neural network as introduced in Section 2 with two layers and 15 nodes in each layer. For the activation function σ we follow Example 4.4 in Benth, Detering and Galimberti Benth et al. (2023) and specify $\sigma(x) = \beta(\psi(x))z$ for a $\psi \in \mathcal{L}(W^{1,2}, \mathbb{R})$, a vector $z \in W^{1,2}$ and β is a real-valued Lipschitz continuous function on the real line. In particular, we choose $\beta(y) := \max\{0, 1 - \exp(-y)\}$, $\psi(h) = a_1 \cdot 0.25 + \dots + a_5 \cdot 0.25$ for $h = \sum_{k=1}^{\infty} a_k e_k$ and $z = e_1 + \dots + e_5$.

We build a training set of $M_{\text{train}} = 5,000,000$ datapoints by first sampling uniformly a vector c_i from \mathcal{K} for $i = 1 \dots M_{\text{train}}$, and then, for each i , we make an independent draw from the Brownian motion W appearing in (28). Based on this draw, we calculate

$$(31) \quad \mathcal{X}(c_i) := \phi(X_{x,t}(T)) \exp \left[\int_t^T c_i(X_{x,t}(s)) ds \right].$$

for $x \in \{-1, -0.5, 0, 0.5, 1\}$. The set $(c_i, \mathcal{X}(c_i))$ for $i = 1, \dots, M_{\text{train}}$ constitutes the training set for each $x \in \{-1, -0.5, 0, 0.5, 1\}$. We train the neural network with 25 epochs and a batch size of 10,000. We denote the resulting neural network by \mathcal{N}_σ^x .

To test the accuracy of our network, we generate a test set of size $M_{\text{test}} = 10,000$. For this, we first randomly sample functions $\tilde{c}_i, i = 1, \dots, M_{\text{test}}$ from \mathcal{K} . For each of these samples we now calculate $u(x, t, \tilde{c}_i)$ for $x \in \{-1, -0.5, 0, 0.5, 1\}$ based on Monte Carlo simulation with 10,000 paths. Note in passing that we include \tilde{c}_i in the argument of u to emphasize the dependency, slightly abusing the notation. We consider $(\tilde{c}_i, u(x, t, \tilde{c}_i)), i = 1, \dots, M_{\text{test}}$ as examples from the ground truth. Next, we calculate the mean square error of the neural network predictions with respect to the ground truth given by

$$\frac{1}{M_{\text{test}}} \sum_{i=1}^{M_{\text{test}}} (\mathcal{N}_\sigma(\tilde{c}_i) - u(x, t, \tilde{c}_i))^2.$$

In Table 1, first column, we list the mean squared error for the different values of x . In Figure 2(a) we provide the resulting box plots and in Figure 3(a)- 3(e) the histograms of the distributions of the errors $\mathcal{N}_\sigma(\tilde{c}_i) - u(x, t, \tilde{c}_i)$. We stress that we essentially fix 5 separate neural networks for each x in this proof-of-concept case study.

Next, for comparison, we fit a DeepONet structure to learn the map $c \mapsto u(t, x, c)$ via minimizing the energy functional (29) based on the samples $(c_i, \mathcal{X}(c_i))$ for $i = 1, \dots, M_{\text{train}}$. We use a 2-layer DeepONet with a branch net of 50 nodes and a trunk net of 50 nodes. We choose a ReLU (rectified linear unit) activation function. The DeepONet has 6,301 parameters, comparable to the number of parameters of our Fréchet network (6,500 parameters). Because the DeepONet requires sampling of c_i on a grid, we evaluate each c_i from the training set on an equally spaced grid $\{y_1, \dots, y_{20}\}$ of size 20. The training set for the DeepONet is composed of the set $\{((c_i, x_j), u(t, x_j, c_i))\}_{i=1, \dots, M_{\text{train}}, 1 \leq j \leq 5}$ where $c_i = (c_i(y_1), \dots, c_i(y_{20}))$ and $(x_1, x_2, x_3, x_4, x_5) = (-1, -0.5, 0, 0.5, 1)$. We train the network again with 25 epochs and a batch size of 10,000. We denote the resulting neural network by $\mathcal{N}_\sigma^{\text{DON}}$. Now we take the same test set as before and evaluate it on the grid, i.e., calculate $\{((\tilde{c}_i, x_j), u(t, x_j, c_i))\}_{i=1, \dots, M_{\text{test}}, 1 \leq j \leq 5}$. The mean squared errors are presented in the right column of Table 1. The error distributions are displayed as a box plot in Figure 2(b) and as histograms in Figures 3(f)-3(j).

Overall we observe an error of similar magnitude for both architectures. For all values of x the mean square error is slightly lower for the Fréchet neural network, except for $x = 0.5$, where it is lower for DeepONet. We further observe that the error distribution for the Fréchet neural network is more symmetric while for DeepONet it is heavily skewed. We stress that the DeepONet is basically trained on a training set 5 times larger than the training set for the Fréchet neural network. This is due to the nature of DeepONet approximating the map $c \mapsto u(t, \cdot, c)$, i.e., it learns the entire solution function $u(t, \cdot, c)$. This requires to feed in the argument x , at which $u(t, \cdot, c)$ is to be evaluated, resulting in the training set $\{((c_i, x_j), u(t, x_j, c_i))\}_{i=1, \dots, M_{\text{train}}, 1 \leq j \leq 5}$. In contrast to Fréchet neural network which is separately trained for each $x \in \{-1, -0.5, 0, 0.5, 1\}$, DeepONet can therefore make use of information across different values of x in the learning process.

As evident from the theory covered in Section 2 and 3, it is possible to learn the entire solution $u(t, \cdot, c)$ with the Fréchet neural network structure. In fact, we know from Section 3 that the non-linear and continuous operator $c \mapsto u(t, \cdot, c)$ is a continuous operator from $W^{m,p}$ to $C(K)$. Because $C(K)$ naturally embeds continuously into $L^2(K)$, we can actually see this operator as an operator from $W^{m,p}$ to $L^2(K)$ where it is still continuous by composition. We can now choose any orthonormal basis in $L^2(K)$ to represent the solution $u(t, \cdot, c)$. For example if $K = [0, 1]^n$, then we can use tensor products of sinus and cosinus functions. We can compute these Fourier coefficients simply by evaluating numerical integrals, without computing derivatives. With the Fourier coefficients and the sinus and cosinus basis functions, structural information about the solution $u(t, \cdot, c)$ could be used. We expect a significant improvement in the learning for Fréchet neural networks in this case. We leave a full scale numerical analysis of this approach for future research.

x Value	Fréchet NN	DeepONet
$x = -1$	0.011	0.047
$x = -0.5$	0.040	0.147
$x = 0$	0.061	0.069
$x = 0.5$	0.042	0.023
$x = 1$	0.016	0.033

TABLE 1. Mean Squared Error for various x values across two methods, rounded to 10^{-3} .

Overall the proof-of-concept case study presented here shows already that using the structural information, being the key in our proposed Fréchet neural network architecture, is promising. We believe that our approach can be applied successfully in many situations where an entire set of partial differential equations needs to be solved at once. For example, in mathematical finance one is interested in pricing derivatives like options (see e.g. Björk (2009)). With X being a model for the market and ϕ signifying the payoff function of a derivative at time T , the price dynamics of the derivative can be described by $u(t, x; \phi)$ with $x = X(t)$. By learning the operator map $\phi \mapsto u(t, x; \phi)$, one has available a pricing

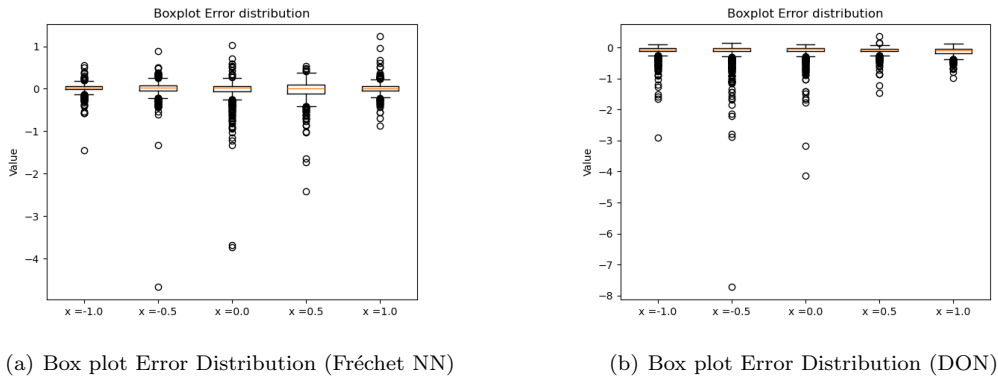


FIGURE 2. Box plots of the error distributions. The left figure shows the result from the Fréchet neural network, and the right figure the results with DeepONet.

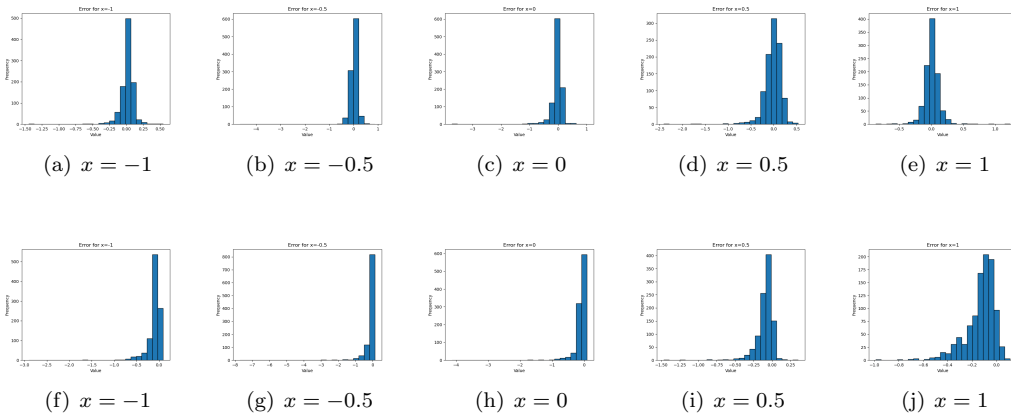


FIGURE 3. Comparison of error distributions for various values of x . The first row shows the result from the Fréchet neural network, and the second row shows the results with DeepONet.

generator for such derivatives and portfolios thereof. The function c is mapping the multivariate process X into an interest rate dynamics in such a context, and from derivatives prices one can consider the inverse problem of re-constructing c from data. Having access to the operator map $(\phi, c) \mapsto u(t, x; \phi, c)$ and its structural representation provides a tool for solving this problem. Moreover, we can price portfolios of derivatives after learning the operator map. A damped L^2 -space is an appropriate space for payoff functions, i.e., $L^2(w)$ with $w(dx) = \exp(-x^2)dx$. Another important problem in option theory is computing the implied volatility. This entails in recovering the covariance function a (the matrix specifying the elliptic operator in (9)) from knowing the prices, given by u . I.e., this is the inverse problem for the operator map $a \mapsto u(t, x; a)$. If one is able to specify a suitable space for a as well as showing continuity of the operator map, we can use our framework for this task.

Random parabolic partial differential equations is another avenue of applications of our operator-learning methodology (see e.g. Nabian & Meidani (2019), who propose a deep neural network architecture to solve high-dimensional random partial differential equations). If one or more of the parameter functions ϕ, c or f are random, by knowing the operator $(\phi, c, f) \mapsto u(t, x; \phi, c, f)$ we can efficiently sample from the solution u by drawing random samples of the input functions. By representing both the input functions and the output map in terms of their basis functions, we are indeed sampling the loadings of the basis expansion of the input, and using the learned network for the output loadings. We believe

this is a fruitful approach for uncertainty quantification, in particular for high dimensional problems.

REFERENCES

- Adams, R. & Fournier, J. (2003), *Sobolev Spaces*, ISSN, Elsevier Science.
URL: <https://books.google.de/books?id=R5A65Koh-EoC>
- Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B. & Stuart, A. (2019), Neural operator: Graph kernel network for partial differential equations, in ‘ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations’.
URL: <https://openreview.net/forum?id=fg2ZFmXFO3>
- Beck, C., Becker, S., Grohs, P., Jaafari, N. & Jentzen, A. (2021), ‘Solving the Kolmogorov PDE by means of deep learning’, *Journal of Scientific Computing* **88**(3).
URL: <http://dx.doi.org/10.1007/s10915-021-01590-0>
- Beck, C., Hutzenthaler, M., Jentzen, A. & Kuckuck, B. (2023), ‘An overview on deep learning-based approximation methods for partial differential equations’, *Discrete and Continuous Dynamical Systems, Series B* **28**(6), 3697–3746.
- Benth, F. E., Detering, N. & Galimberti, L. (2023), ‘Neural networks in Fréchet spaces’, *Annals of Mathematics and Artificial Intelligence* **91**(1), 75–103.
URL: <https://doi.org/10.1007/s10472-022-09824-z>
- Benth, F. E., Detering, N. & Galimberti, L. (2024), ‘Pricing options on flow forwards by neural networks in a Hilbert space.’, *Finance & Stochastics* **28**, 81–121.
- Björk, T. (2009), *Arbitrage Theory in Continuous Time*, Oxford Finance Series, 3rd edn, Oxford University Press.
- Cao, Q., Goswami, S. & Karniadakis, G. E. (2024), ‘Laplace neural operator for solving differential equations’, *Nature Machine Intelligence* pp. Online June 24, 2024.
- Chen, T. & Chen, H. (1995), ‘Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems’, *IEEE Transactions on Neural Networks* **6**(4), 911–917.
- E, W., Han, J. & Jentzen, A. (2017), ‘Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations’, *Communications in Mathematics and Statistics* **5**, 349–380.
- Friedman, A. (1975), *Stochastic Differential Equations and Applications: Volume 1*, Dover.
- Han, J., Jentzen, A. & E, W. (2018), ‘Solving high-dimensional partial differential equations using deep learning’, *Proceedings of the National Academy of Sciences* **115**, 8505–8510.
- Heil, C. (2011), *A Basis Theory Primer*, Applied and Numerical Harmonic Analysis, Springer New York Dordrecht Heidelberg London.
- Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2022), ‘Neural operator: Learning maps between function spaces with applications to pdes’, *Journal of Machine Learning Research* pp. 1–97.
- Lanthaler, S., Mishra, S. & Karniadakis, G. E. (2022), ‘Error estimates for deepoanets: a deep learning framework in infinite dimensions’, *Transactions on Mathematics and its Applications* **6**(1), 1–141.
- Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2021), Fourier neural operator for parametric partial differential equations, in ‘International Conference on Learning Representations’.
URL: <https://openreview.net/forum?id=c8P9NQVtmnO>
- Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K. & Anandkumar, A. (2024), ‘Physics-informed neural operator for learning partial differential equations’, *ACM/JMS Journal of Data Science* **1**(3), Article 9.
URL: <https://doi.org/10.1145/3648506>
- Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. (2019), ‘Deepoanet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators’, *arXiv:1910.03193v3*.
- Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. (2021), ‘Learning nonlinear operators via deepoanet based on the universal approximation theorem of operators’, *Nature*

- Machine Intelligence* **3**(3), 218–229.
URL: <https://doi.org/10.1038/s42256-021-00302-5>
- Meyer, Y. (2009), *Wavelets and Operators*, Cambridge Studies in Advanced Mathematics (37), Cambridge University Press Cambridge.
- Nabian, M. A. & Meidani, H. (2019), ‘A deep learning solution approach for high-dimensional random differential equations’, *Probabilistic Engineering Mechanics* **57**, 14–25.
URL: <https://www.sciencedirect.com/science/article/pii/S0266892018301681>
- Schaefer, H. (1971), *Topological Vector Spaces*, N. Bourbaki, Springer.
- Schwartz, L. (1950), *Théorie des Distributions*, number v. 1 in ‘Actualités scientifiques et industrielles’, Hermann.
URL: <https://books.google.com/books?id=tsgc0AEACAAJ>
- Triebel, H. (2004), ‘A note on wavelet bases in function spaces’, *Banach Center Publications* **64**(1), 193–206.
- Yang, L., Liu, S., Meng, T. & Osher, S. J. (2023), ‘In-context operator learning with data prompts for differential equation problems’, *Proceedings of the National Academy of Sciences* **120**(39), e2310142120.
URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2310142120>

FRED ESPEN BENTH, UNIVERSITY OF OSLO, DEPARTMENT OF MATHEMATICS, P.O. BOX 1053, BLINDERN, N-0316 OSLO, NORWAY
Email address: fredb@math.uio.no

NILS DETERING, HEINRICH HEINE UNIVERSITY DÜSSELDORF, DEPARTMENT OF MATHEMATICS, UNIVERSITÄTSTRASSE 1, 40225 DÜSSELDORF, GERMANY
Email address: nils.detering@hhu.de

LUCA GALIMBERTI, KING’S COLLEGE LONDON, DEPARTMENT OF MATHEMATICS, STRAND BUILDING, WC2R 2LS, LONDON, UK
Email address: luca.galimberti@kcl.ac.uk