Bregman Proximal Viewpoint on Neural Operators

Abdel-Rahim Mezidi, Jordan Frecon, Saverio Salzo, Amaury Habrard,
Massimiliano Pontil, Rémi Emonet, Marc Sebban

Bregman Proximal Viewpoint on Neural Operators

Abdel-Rahim Mezidi[1,*], **Jordan Patracone**[1,*] **Saverio Salzo**[2,*]
Amaury Habrard[1] **Massimiliano Pontil**[3,4] **Remi Emonet**[1] **Marc Sebban**[1]

[1] Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School, Inria, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France.
[2] DIAG, Sapienza University of Rome, 00185 Rome, Italy.
[3] Computational Statistics and Machine Learning, IIT, Genova, Italy.
[4] Departement of Computer Science, UCL, London, United Kingdom.
[*] Equal contribution.

Abstract

We present several advances on neural operators by viewing the action of operator layers as the minimizers of Bregman regularized optimization problems over Banach function spaces. The proposed framework allows interpreting the activation operators as Bregman proximity operators from dual to primal space. This novel viewpoint is general enough to recover classical neural operators as well as a new variant, coined Bregman neural operators, which also includes a skip-like connection. Numerical experiments support the added benefits of the Bregman variant of Fourier neural operators for training deeper and more accurate models.

1 Introduction

Neural operators [14, 15], a recent extension of neural networks, have emerged as a versatile framework for learning mappings between function spaces. These operators have shown great potential in solving partial differential equations (PDEs) and simulating complex dynamical systems. The exploration of neural architectures for the approximation and learning of operators has led to the development of a variety models.
One influential contribution is the Fourier Neural Operator (FNO) [17], which transform encoded input data into frequency components in order to learn intricate relationships in the frequency domain. More recently, the Group-Equivariant FNO (G-FNO) [12] additionally leverages symmetries to design equivariant Fourier layers, thereby enhancing the representation power and robustness of the architecture. The FNO are extended to Wavelet Neural Operators (WNO) [27] by replacing Fourier layers with wavelet layers to further exploit multiscale information. The U-shaped Neural Operator (U-NO) [23] adapts the U-net architecture for neural operators, enabling mapping between function spaces through integral operators, thus broadening the applicability of neural architectures to diverse domains. Differently, the DeepONet architecture [19] comprises two intertwined components: a branch network responsible for encoding discrete input function spaces, and a trunk network dedicated to encoding the domain of output functions. Operating as a conditional model, DeepONet leverages the embedding of inputs and outputs via a dot product operation, facilitating the approximation of complex functions through a structured network topology. Finally, Neural Inverse Operators (NIO) [20] tackle inverse problems by combining DeepONet and FNO architectures to map operators to functions, thereby extending the applicability of neural operators to coefficient estimation tasks.
Some approaches inspired by attention mechanisms, pivotal in image and natural language

Preprint. Under review.

processing, have also been considered in operator learning. LOCA (Learning Operators with Coupled Attention) [13] facilitates robust gradient estimation, particularly in scenarios with limited training data, by combining attention with kernel mechanisms. The General Neural Operator Transformer (GNOT) [11] is a scalable framework based self-attention mechanisms allowing to deal with heterogeneous inputs useful for modeling diverse physical systems. Some physics-informed variants integrating information from PDEs during the learning process or as constraints have been proposed recently enhancing model interpretability and generalization: PI-DeepONet [29] and its Long-Time Integration variant (LTI-PI-DeepONet) [28], PINO (Physics-Informed Neural Operator) [18] a hybrid extension of FNO, or other variations such as V-DeepONet [10] and Modified DeepONet [30].

Contributions. The architectural design of neural operators has primarily followed that of neural networks. However, such models often face challenges related to stability and efficiency, particularly when training deeper networks, as evidenced by the fact that most released models consist of only four layers. In this work, we propose a novel expressive framework for neural operators by conceptualizing the action of operator layers as the minimizers of Bregman regularized optimization problems over Banach function spaces. By interpreting the activation operators as Bregman proximity operators mapping from dual to primal spaces, our approach generalizes existing neural operators and introduces a new variant, termed Bregman neural operators, showing better prediction as the number of layers increases.

Outline. The rest of the paper is organized as follows: Section 2 is dedicated to the presentation of definitions and background knowledge on neural operators and Bregman proximity operator. In Section 3, we introduce the operator layers as the solution of a functional optimization problem. In addition, we show that this new mapping allows recovering the classical neural operators and creating a more general family of so-called Bregman neural operators. In Section 4, we provide a preliminary universal approximation result for Bregman neural operators networks. Finally, in Section 5, we conduct an extended experimental study comparing on some benchmark datasets our Bregman variant with the classical FNO.

2 Background and Definitions

Here, we introduce some definitions required for the understanding of the rest of the paper as well as the necessary background on neural operators and Bregman proximity operator.

Notations. Let Let $\mathcal{V}$ and $\mathcal{V}^*$ be two Banach spaces put in duality via the pairing $\langle\cdot,\cdot\rangle\colon \mathcal{V}\times\mathcal{V}^*\to\mathbb{R}$. If $\Phi\colon \mathcal{V}\to\left]-\infty,+\infty\right]$, we denote by $\operatorname{dom}\Phi=\{v\in\mathcal{V}\,|\,\Phi(v)<+\infty\}$ its *effective domain*. For every proper convex function $\Phi\colon \mathcal{V}\to\left]-\infty,+\infty\right]$, we set its subdifferential

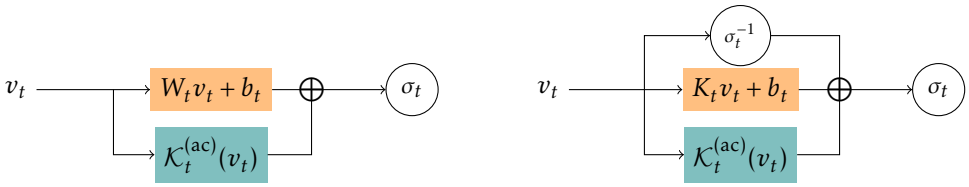
$$\partial\Phi(v)=\{v^*\in\mathcal{V}^*\,|\,\text{for all } u\in\mathcal{V},\ \Phi(u)\geq\Phi(v)+\langle u-v,v^*\rangle\},$$

if $v\in\operatorname{dom}\Phi$, and $\partial\Phi(v)=\varnothing$, otherwise. We set $\operatorname{dom}\partial\Phi=\{v\in\operatorname{dom}\Phi\,|\,\partial\Phi(v)\neq\varnothing\}$ and the *range* $\operatorname{ran}\partial\Phi=\{v^*\in\mathcal{V}^*\,|\,\exists v\in\mathcal{V}\text{ s.t. }v^*\in\partial\Phi(v)\}$. When $\partial\Phi(v)$ is a singleton, we denote by $\tilde{\nabla}\Phi$ its unique element. If $\Phi\colon\mathcal{V}\to\left]-\infty,+\infty\right]$, its *Fenchel conjugate* is the function $\Phi^*\colon\mathcal{V}\to\left]-\infty,+\infty\right]$ such that $\Phi^*(v^*)=\sup_{v\in\mathcal{V}}\langle v,v^*\rangle-\Phi(v)$. We denote by $\Gamma_0(\mathcal{V})$ the set of proper convex and lower-semicontinuous functions on $\mathcal{V}$. The Fenchel-Moreau theorem ensures that $\Phi\in\Gamma_0(\mathcal{V})\ \Rightarrow\ \Phi^*\in\Gamma_0(\mathcal{V}^*)$. We denote by $|\cdot|$ the Euclidean norm in $\mathbb{R}^m$. If $D\subset\mathbb{R}^m$ is a nonempty open bounded set and $p\in[1,+\infty]$, we denote by $L^p(D,\mathbb{R}^m)$ the Lebesgue space of p-integrable functions (essentially bounded functions, if $p=+\infty$) from D to $\mathbb{R}^m$.

2.1 Operator Learning: Application to Learning the Solution Map of PDEs

Operator learning finds significant applications in the context of PDEs in order to efficiently approximate solutions to PDEs without the need to solve them repeatedly from scratch [18, 25, 24]. Given a nonempty bounded open set $D\subset\mathbb{R}^d$, and some time horizon $\tau>0$, we let the generic family of PDEs over $D\times[0,\tau]$ of the form

$$F_a\Big((\partial^\alpha u(x,t))_{\alpha\in\mathbb{N}^{d+1},|\alpha|\leq k}\Big)=f(x,t)\ \text{ on } D\times\left]0,\tau\right]\ \text{ and }\ \begin{cases}u(x,0)=u_0(x)\ \text{ on } D,\\ u(x,t)=u_b(x,t)\ \text{ on } \partial D\times\left]0,\tau\right].\end{cases}\tag{1}$$



(a) Neural Operator. (b) Bregman Neural Operator.

Figure 1: Illustration of the t-th layer of (Bregman) Neural Operators. On the left, the identity term and the linear term $K_t v_t + b_t$ have been merged into $(\mathbb{1} + K_t)v_t = W_t v_t$.

where F_a is a possibly nonlinear partial differential operator, $f \in L^2(D\times]0,\tau], \mathbb{R}^m)$ denotes a source term, $u_b \in L^2(\partial D\times]0,\tau], \mathbb{R}^m)$ is a boundary condition, $u_0 \in L^2(D, \mathbb{R}^m)$ is an initial condition, and $u\colon D \to \mathbb{R}^m$ is the solution of the PDE.
The problem we will tackle in our numerical section is the *initial value problem*. This involves finding the oracle mapping $\mathcal{G}$ from any initial condition function u_0 to the solution $u(\cdot, \bar{\tau})$ of the PDE at a certain time horizon $\bar{\tau} \in]0,\tau]$.
More generally, the oracle operator $\mathcal{G}$ could be a mapping between two different function spaces $\mathcal{A}$ and $\mathcal{U}$. Without loss of generality, given some bounded open sets $D \subset \mathbb{R}^d$, with $d \in \mathbb{N}_+$, we let $\mathcal{A} = \mathcal{A}(D, \mathbb{R}^m)$ and $\mathcal{U} = \mathcal{U}(D, \mathbb{R}^k)$, with $m, k \in \mathbb{N}_+$, be some (possibly suitable subsets of) separable Banach spaces of functions. For instance, $\mathcal{A}$ can represent the spaces of continuous functions from $D \to \mathbb{R}^m$. Hereafter, $\mathcal{A}$ and $\mathcal{U}$ will be referred to as the spaces of *input functions* and *output functions*, respectively. In a nutshell, operator learning consists in finding the unknown ground-truth correspondence operator $\mathcal{G}\colon \mathcal{A} \to \mathcal{U}$ given $n \in \mathbb{N}_+$ pairs of input and output functions $\{a_i, u_i\}_{i=1}^n$.

2.2 Neural Operators

Among the existing models to parametrize an approximation of $\mathcal{G}$, we focus on neural operators, which are parametric mappings $\mathcal{N}\colon \mathcal{A} \to \mathcal{U}$ of the form

$$(\forall a \in \mathcal{A}), \quad \mathcal{N}(a) = \mathcal{Q} \circ \mathcal{L}_T \circ \ldots \circ \mathcal{L}_1 \circ \mathcal{P}(a), \tag{2}$$

where

- $\mathcal{P}\colon \mathcal{A}(D, \mathbb{R}^m) \to \mathcal{A}(D, \mathbb{R}^{m_0})$ is a local *lifting operator* mapping the input function to its first hidden representation;
- $\mathcal{Q}\colon \mathcal{U}(D, \mathbb{R}^{m_T}) \to \mathcal{U}(D, \mathbb{R}^k)$ is a local *projection operator* mapping the last hidden representation to the output function;
- For every $t \in \{1, \ldots, T\}$, $\mathcal{L}_t\colon \mathcal{V}_{t-1}(D_t, \mathbb{R}^{m_{t-1}}) \to \mathcal{V}_t(D_t, \mathbb{R}^{m_t})$ is an *operator layer* where each $D_t \subset \mathbb{R}^{d_t}$ is an open bounded set, $\mathcal{V}_t = \mathcal{V}_t(D_t, \mathbb{R}^{m_t})$ is a suitable Banach space of functions with $d_0 = d$ and $d_T = s$ such that $\mathcal{V}_0 = \mathcal{A}(D, \mathbb{R}^{m_0})$ and $\mathcal{V}_T = \mathcal{U}(D, \mathbb{R}^{m_T})$, for consistency.

Most methodological developments on neural operators have been focused on tailoring parametric forms of the operator layers $\{\mathcal{L}_1, \ldots, \mathcal{L}_T\}$ suited to the application at hand. Traditionally, the design of neural operator follows closely that of standard neural networks (i.e., a composition of finite-dimensional linear layers followed by non-linear activations) by replacing linear layers with linear operators in function spaces and by interpreting activation functions through their extension to Nemytskii operators. To the best of our knowledge, most operator layers in the literature restrict to the peculiar class of Hilbert space $\mathcal{V}_t = L^2(D, \mathbb{R}^{m_t})$ for every $t \in \{1, \ldots, T\}$. When input space D is the same throughout the layers, a popular class of operator layers, sketched in Figure 1a, is of the form

$$\mathcal{L}_t(v_t) = \sigma_t(W_t v_t + \mathcal{K}_t(v_t) + b_t), \tag{3}$$

where $W_t \in \mathbb{R}^{m_{t+1}\times m_t}$ is a matrix, $b_t \in \mathbb{R}^{m_{t+1}}$ is a bias and σ_t is a *local* nonlinear map acting pointwise from $\mathbb{R}^{m_{t+1}}$ to $\mathbb{R}^{m_{t+1}}$. Departing from feed-forward neural networks, neural operators additionally possess a *non-local* linear operator $\mathcal{K}_t\colon L^2(D, \mathbb{R}^{m_t}) \to L^2(D, \mathbb{R}^{m_{t+1}})$. In its

simplest version, $\mathcal{K}_t$ is an integral kernel operator of the form $(\mathcal{K}_t(v))(x) = \int_D \kappa_t(x,y)v(y)dy$, for all $x \in D$, with κ being a kernel to be specified [15]. Specific examples include those based upon a convolution performed in the Fourier space [17, 14], a graph kernel network [1] or its multipole variant [16] to name a few.
Hereafter, we follow a different path and propose to interpret operator layers from the viewpoint of a proximal optimization by seeing the parametric form of (3) as the minimizer of a Bregman regularized optimization problem. This novel perspective allows us to propose a novel architecture, displayed in Figure 1b, of the form

$$\mathcal{L}_t(v_t) = \sigma_t(\sigma_t^{-1}(\mathcal{M}_t v_t) + W_t v_t + \mathcal{K}_t(v_t) + b_t), \tag{4}$$

involving a nonlinear skip-like connection term $\sigma_t^{-1}(\mathcal{M}_t v_t)$. In this formulation, when all the weights are zero and $\mathcal{M}_t$ is the identity, then $\mathcal{L}_t$ is the identity operator. This architecture was originally proposed in [9] in the finite dimensional setting.

In the next section, we define the notion of *Bregman Proximity Operator* which will serve to introduce or novel viewpoint on neural operators.

2.3 Bregman Proximity Operator

The definition of Bregman proximity operator relies on the choice of a Bregman divergence, loosely called distance, which itself is built upon a Legendre function.

Definition 1. *A function $\phi\colon \mathbb{R}^m \to]-\infty,+\infty]$ is called* Legendre *if it is proper convex lower semicontinuous and satisfies the following properties: i)* $\operatorname{int}(\operatorname{dom}\phi) = \operatorname{dom}\partial\phi$ *and $\partial\phi$ is single-valued on its domain; ii) ϕ is strictly convex on* $\operatorname{int}(\operatorname{dom}\phi)$.

Remark 1. *One can prove that ϕ is Legendre if and only if ϕ^* is Legendre. Moreover, if ϕ is Legendre, then ϕ and ϕ^* are differentiable on* $\operatorname{int}(\operatorname{dom}\phi)$ *and* $\operatorname{int}(\operatorname{dom}\phi^*)$ *respectively and*

$$\nabla\phi\colon \operatorname{int}(\operatorname{dom}\phi) \to \operatorname{int}(\operatorname{dom}\phi^*) \quad \textit{and} \quad \nabla\phi^*\colon \operatorname{int}(\operatorname{dom}\phi^*) \to \operatorname{int}(\operatorname{dom}\phi)$$

are bijective and inverse of each other.

In the finite dimensional setting, Legendre functions ϕ are typically built from an elementary Legendre function $\varphi\colon \mathbb{R} \to]-\infty,+\infty]$ as $\phi\colon x \in \mathbb{R}^m \to \sum_{i=1}^m \varphi(x_i)$. Since here we stand in an infinite dimensional setting, i.e., Lebesgue function space, the counterpart of the previous finite sum is a convex integral functional defined below.

Fact 1 (Convex integral functionals on Lebesgue spaces based on Legendre function)**.** *Let $D \subset \mathbb{R}^d$ be an open bounded set. Let $p,q \in [1,+\infty]$ be conjugate exponents, that is such that $1/p + 1/q = 1$, and set $\mathcal{V} := L^p(D,\mathbb{R}^m)$ and $\mathcal{V}^* = L^q(D,\mathbb{R}^m)$. The spaces $\mathcal{V}$ and $\mathcal{V}^*$ can put in duality via the pairing $\mathcal{V} \times \mathcal{V}^* \to \mathbb{R}$, $(v,u) \mapsto \langle v,u\rangle = \int_D \langle v(x),u(x)\rangle dx$. Let $\phi \in \Gamma_0(\mathbb{R}^m)$ be a Legendre function and let $\Phi\colon \mathcal{V} \to]-\infty,+\infty]$ be such that*

$$\Phi(v) = \int_D \phi(v(x))dx. \tag{5}$$

Then $\Phi \in \Gamma_0(\mathcal{V})$, $\operatorname{dom}\partial\Phi = \{v \in \mathcal{V} \,|\, \text{for a.e. } x \in D, v(x) \in \operatorname{int}(\operatorname{dom}\phi)$ and $(\nabla\phi)\circ v \in \mathcal{V}^*\}$, $\partial\Phi$ is single valued on $\operatorname{dom}\partial\Phi$, and, for every $v \in \operatorname{dom}\partial\Phi$, $\partial\Phi(v) = \{\nabla\phi \circ v\}$. The unique element $\nabla\phi \circ v$ of $\partial\Phi(v)$ will be denoted by $\tilde{\nabla}\Phi(v)$, suggesting it will serve as a kind of gradient of Φ at v[1]

The Legendre function Φ in (5) inherits certain properties of ϕ, such as p-uniform convexity — an extension of strong convexity when $p = 2$. This characteristic, proved in Proposition 4 of the appendix, will play a pivotal role in Remark 3. Additionally, we have the following.

Remark 2. *In Fact 1, suppose that $p = 1$ and $\operatorname{dom}\phi^* = \mathbb{R}^m$. Then $\operatorname{ran}\partial\Phi = \mathcal{V}^*$. Indeed, we note that $\nabla\phi\colon \operatorname{int}(\operatorname{dom}\phi) \to \mathbb{R}^m$ is a continuous bijection with inverse $\nabla\phi^*$, which is also continuous. Therefore if we let $u \in \mathcal{V}^* = L^\infty(D,\mathbb{R}^m)$ and set $v = (\nabla\phi^*)\circ u$, since u is essentially bounded, we have that v is essentially bounded too, and hence integrable. In the end $v \in L^1(D,\mathbb{R}^m)$ and $u = (\nabla\phi)\circ v \in \partial\Phi(v)$.*

[1] Note that in general the domain of the function Φ has empty interior, so Gâteaux and/or Frechet differential cannot be properly defined.

We are now equipped to define Bregman distances in Lebesgue spaces.

Definition 2 (Bregman distance in Lebesgue spaces)**.** *Under the notations of Fact 1, the* Bregman distance *with respect to* Φ *is defined as*

$$D_\Phi : \mathcal{V} \times \mathcal{V} \to [0, +\infty], \quad D_\Phi(u, v) = \begin{cases} \Phi(u) - \Phi(v) - \langle u - v, \tilde{\nabla}\Phi(v) \rangle & \text{if } v \in \operatorname{dom} \partial\Phi \\ +\infty & \text{otherwise.} \end{cases}$$

Finally, we can define the Bregman proximity operator [22].

Definition 3 (Bregman proximity operator)**.** *Let* $\mathcal{V} = L^p(D, \mathbb{R}^m)$ *with* $p \in [1, +\infty[$*. Let* $g \in \Gamma_0(\mathcal{V})$ *and let* $\Phi \in \Gamma_0(\mathcal{V})$ *be defined as in Fact 1, with* $\phi \in \Gamma_0(\mathbb{R}^m)$ *be Legendre and such that* $\operatorname{ran} \partial(\Phi + g) = \mathcal{V}^*$*. Then the* Bregman proximity operator *of* g *relative to* Φ *is defined as*

$$\operatorname{prox}_g^\Phi : \mathcal{V}^* \to \mathcal{V}, \quad v^* \mapsto \operatorname{argmin} \Big\{ \langle \cdot, -v^* \rangle + \Phi + g \Big\}.$$

Note that $\operatorname{prox}_g^\Phi$ is well-defined since $\Phi + g$ is strictly convex and lower semicontinuous and $\operatorname{ran} \partial(\Phi + g) = \mathcal{V}^*$, and it holds $\operatorname{prox}_g^\Phi = [\partial(\Phi + g)]^{-1}$.

Remark 3.

(i) *If instead of* $\operatorname{ran} \partial(\Phi + g) = \mathcal{V}^*$*, one asks the stronger condition* $\operatorname{ran}(\partial\Phi + \partial g) = \mathcal{V}^*$*, then we have* $\partial(\Phi + g) = \partial\Phi + \partial g$ *and the Bregman proximity operator writes down as* $\operatorname{prox}_g^\Phi = (\partial\Phi + \partial g)^{-1}$ *and* $\operatorname{ran}(\operatorname{prox}_g^\Phi) \subset \operatorname{dom} \partial\Phi$*.*

(ii) *By Proposition 4, if* $\mathcal{V} = L^p(D, \mathbb{R}^m)$ *with* $p \in]1, +\infty[$*, the condition* $\operatorname{ran} \partial(\Phi + g) = \mathcal{V}^*$ *is satisfied if* ϕ *is* p*-uniformly convex. Moreover, by Remark 2, if* $p = 1$ *and* $\operatorname{dom} \phi^* = \mathbb{R}^m$*, then* $\operatorname{ran} \partial\Phi = \mathcal{V}^*$*.*

3 Revisiting Neural Operators

In Section 3.1, we propose a novel Bregman proximal viewpoint on operator layers. Then, we establish several connections. First, we show in Section 3.2 that the proposed framework is general enough to recover most classical operator layers when the Legendre function ϕ is the Euclidean distance. Second, we showcase in Section 3.3 how it yields a new variant when ϕ is a general Bregman divergence. Finally, we apply our framework to Fourier neural operators in Section 3.4.

3.1 Bregman Proximal Viewpoint on Operator Layers

Departing from usual kernel-based points of view [14], we suggest defining operator layers as the solution of a functional optimization problem. For every $t = 1, \cdots, T$, $\mathcal{L}_t : \mathcal{V}_{t-1} \to \mathcal{V}_t$,

$$\mathcal{L}_t(v) = \operatorname*{argmin}_{w \in \mathcal{V}_t} \Big\{ -\langle w, \mathcal{K}_t(v) + b_t \rangle + g_t(w) + D_{\Phi_t}(w, \mathcal{M}_t v) \Big\} = \operatorname{prox}_{g_t}^{\Phi_t} \Big(\tilde{\nabla}\Phi_t(\mathcal{M}_t v) + \mathcal{K}_t(v) + b_t \Big), \quad (6)$$

where

- $\Phi_t : \mathcal{V}_t \to]-\infty, +\infty]$ is a convex integral functional on an appropriate Lebesgue space based on some Legendre function $\phi_t \in \Gamma_0(\mathbb{R}^{m_t})$, as defined in Fact 1. $D_{\Phi_t} : \mathcal{V}_t \times \mathcal{V}_t \to [0, +\infty]$ is the corresponding Bregman distance as detailed in Definition 2
- $\mathcal{M}_t : \mathcal{V}_{t-1} \to \mathcal{V}_t$ is a bounded linear operator which maps $\operatorname{dom} \partial\Phi_{t-1}$ into $\operatorname{dom} \partial\Phi_t$,
- $b_t \in \mathcal{V}_t^*$ and $\mathcal{K}_t : \mathcal{V}_{t-1} \to \mathcal{V}_t^*$ is a bounded linear operator of the form

$$\mathcal{K}_t(v)(x) = \int_{D_{t-1}} \kappa^{(t)}(x, dy) v(y),$$

with $\kappa^{(t)} : D_t \times \mathfrak{B}(D_{t-1}) \to \mathbb{R}^{m_t \times m_{t-1}}$ a *(transition) kernel* from D_{t-1} to D_t, meaning a function which is measurable with respect to the first variable and a finite measure with respect to the second variable.

- $g_t \in \Gamma_0(\mathcal{V}_t)$ and $\operatorname{ran}(\partial\Phi_t + \partial g_t) = \mathcal{V}_t^*$.

Remark 4.

(i) *In* (6)*, the proximity operator plays the role of an activation function operator, which in general will have the form of a nonlinear Nemytskii operator. Moreover, differently from [14], in* (6) *there is an additional nonlinear term, which is* $\tilde{\nabla}\Phi_t(\cdot)$.

(ii) *In view of Remark 3*(i)*, the condition* $\operatorname{ran}(\partial\Phi_t+\partial g_t) = \mathcal{V}_t^*$ *implies that* $\operatorname{prox}_{g_t}^{\Phi_t} = (\partial\Phi_t+\partial g_t)^{-1}$ *and hence* $\operatorname{ran}(\operatorname{prox}_{g_t}^{\Phi_t}) \subset \operatorname{dom}\partial\Phi_t$*. In this way* $\operatorname{dom}\mathcal{L}_t = \mathcal{M}_t^{-1}(\operatorname{dom}\partial\Phi_t)$ *and* $\operatorname{ran}(\mathcal{L}_t) \subset \operatorname{dom}\partial\Phi_{t+1}$ *and the composition* (2) *is well-defined provided that for the lifting operator* $\mathcal{P}$ *it holds* $\operatorname{ran}(\mathcal{P}) \subset \operatorname{dom}\partial\Phi_1$ *(e.g., if* $\mathcal{P}(v)(x) = \nabla\phi_1^*(Pv(x))$*).*

(iii) *When* $\mathcal{V}_{t-1} = \mathcal{V}_t$ *and* $\mathcal{M}_t$ *is the identity, the operator layer* (6) *takes the form* $\operatorname{prox}_{g_t}^{\Phi_t}(\tilde{\nabla}\Phi_t(v) - \mathcal{B}_t v) = (\partial\Phi_t + \partial g_t)^{-1}(\tilde{\nabla}\Phi_t - \mathcal{B}_t)(v)$*, where* $\mathcal{B}_t\colon \mathcal{V}_t \to \mathcal{V}_t^*$*. This is a* Bregman forward-backward operator*, which is well-known in the context of operator splitting methods in optimization [22, 6].*

Remark 5. *Often in applications the kernel of the linear operator* $\mathcal{K}_t$ *is split into two terms: one absolutely continuous part and a single pure point part, i.e.,* $\kappa^{(t)} = \kappa_{ac}^{(t)} + \kappa_p^{(t)}$*, where, for every* $x \in D_t$*, and measurable set* $A \subset D_{t-1}$*,*

$$\kappa_{\mathrm{ac}}^{(t)}(x, A) = \int_A k_t(x, y)dy \quad \text{and} \quad \kappa_p^{(t)}(A) = K_t\delta_{\varphi_t(x)}(A),$$

with $k_t\colon D_t \times D_{t-1} \to \mathbb{R}^{m_t \times m_{t-1}}$*,* $K_t \in \mathbb{R}^{m_t\times m_{t-1}}$*,* $\varphi_t\colon D_t \to D_{t-1}$ *measurable, and* $\delta_{\varphi_t(x)}$ *the delta Dirac at* $\varphi_t(x) \in D_{t-1}$*. Thus, we have*

$$\mathcal{K}_t(v)(x) = \mathcal{K}_t^{(\mathrm{ac})}(v)(x) + \mathcal{K}_t^{(\mathrm{p})}(v)(x) = \int_{D_{t-1}} k_t(x, y)v(y)dy + K_t v(\varphi_t(x)).$$

In the next section, we showcase how the mapping in (6) matches classical neural operator layers and how it can be leveraged to devise a novel variant.

3.2 Classical Neural Operators

Our main result is stated in the proposition below.

Proposition 1. *Let* $\phi_t = (1/2)|\cdot|^2$*,* $p = 2$ *and* $g_t(v) = \Psi_t(v) - (1/2)\|v\|^2$ *with* $\Psi_t(v) = \int_{D_t}\psi_t(v(x))dx$ *where* ψ_t *is strongly convex Legendre function. Then,* $g_t \in \Gamma_0(\mathcal{V}_t)$ *and*

$$\mathcal{L}_t(v) = \operatorname{prox}^{\frac{1}{2}|\cdot|^2}_{\Psi_t - \frac{1}{2}|\cdot|^2}\Big(\mathcal{M}_t v + \mathcal{K}_t(v) + b_t\Big) = \nabla\Psi^*(\mathcal{M}_t v + \mathcal{K}_t(v) + b_t), \tag{7}$$

where $\nabla\Psi^*$ *matches a variety of monotone activation operators. In addition, when the domains are all the same, say* $D_t = D$*,* $\mathcal{M}_t = \mathbb{1}$*, and the linear operator* $\mathcal{K}_t = \mathcal{K}_t^{(\mathrm{ac})} + \mathcal{K}_t^{(\mathrm{p})}$ *is as given in Remark 5, then* $\mathcal{L}_t(v) = \nabla\Psi^*((\mathbb{1} + K_t)v + \mathcal{K}_t^{(ac)}(v) + b_t)$*, where* $(\mathbb{1} + K_t)$ *is can be written as* W_t*.*

In essence, Proposition 1 shows that the parametric structure of operator layers can be interpreted via the Bregman proximal operator, when the Bregman distance reduces to the Euclidean distance. The crucial aspect in establishing this connection is the observation that the Euclidean proximity operator of $g_t = \Psi - (1/2)\|\cdot\|^2$ simplifies to $\nabla\Psi^* = (\psi^{*\prime} \circ \cdot)$, aligning with a broad spectrum of activation operators given an appropriate selection of ψ. We report in Table 1 the corresponding ψ to retrieve several well-known activation operators. A proof concerning the characterization of the SoftPlus function is included in the appendix. To the best of our knowledge, $\nabla\Psi_t^*$ can only match monotonic activation operators, which notably discards GeLu and swish. While this connection has been previously noted in the neural network literature [7, 9], our work extends this analysis to Banach function spaces. In addition, embracing the characterization of the proximity operator via its minimization viewpoint shed another light on the action of operator layers, as discussed below.

Table 1: Relationship between Legendre function ψ and activation function $\psi^{*\prime}$.

$\mathrm{dom}\psi$	ψ	ψ'	ψ^*	$\psi^{*\prime}$
$[-1,1]$	$t \mapsto -\sqrt{1-t^2}$	$t \mapsto t/\sqrt{1-t^2}$	$t \mapsto \sqrt{1+t^2}$	ISRU
$[0,1]$	$t \mapsto t\log t + (1-t)\log(1-t)$	$t \mapsto \log(\frac{t}{1-t})$	$t \mapsto \log(1+\mathrm{e}^t)$	Sigmoid
$[-1,1]$	$t \mapsto \log(1-t^2) + t\,\mathrm{arctanh}(t)$	arctanh	$\log\cosh$	tanh
$[-1,1]$	$t \mapsto \sqrt{1-t^2} + t\arcsin(t)$	arcsin	$-\cos$	sin
$\mathbb{R}_{>0}$	$t \mapsto \frac{1}{\beta^2}\mathrm{Li}_2(\mathrm{e}^{-\beta t}) + \frac{1}{2}t^2$	$t \mapsto \frac{1}{\beta}\log(\mathrm{e}^{\beta t}-1)$	$t \mapsto -\frac{1}{\beta^2}\mathrm{Li}_2(-\mathrm{e}^{\beta t})$	$\mathrm{SoftPlus}_\beta$

Remark 6. *The operator in* (7) *can be rewritten as a prediction operator regularized by* Ψ*, that is the solution of an optimization problem which balances an affinity term* $\langle \cdot, \mathcal{M}_t v + \mathcal{K}_t(v) + b_t \rangle$ *, and a confidence term* Ψ*, i.e.,*

$$\mathcal{L}_t(v) = \mathrm{argmin}\Big\{ -\langle \cdot, \mathcal{M}_t v + \mathcal{K}_t(v) + b_t \rangle + \Psi \Big\}. \tag{8}$$

The choice of the regularization Ψ *and its domain* $\mathrm{dom}\,\Psi$ *governs certain properties of* $\mathcal{L}_t$*. We emphasize that, in our case, the regularization is w.r.t. the output of each layer and not w.r.t. the output of the overall neural operator, as is the case in the literature [4, 3]. We expect that this perspective could give rise to novel activation operators, more relevant for the task at hand, by exploring other regularizers* Ψ*.*

3.3 Bregman Neural Operators

We now provide the counterpart of Proposition 1 for general Bregman distance.

Proposition 2. *Let* $\phi_t = \psi_t \neq (1/2)|\cdot|^2$*, with* ψ_t *being a strongly convex Legendre function, and* $g_t = 0$*. Then,* $\mathcal{L}_t$ *acts between* L^2 *spaces as follows*

$$\mathcal{L}_t(v) = \mathrm{prox}_0^{\Psi_t}\Big(\tilde{\nabla}\Psi_t(\mathcal{M}_t v) + \mathcal{K}_t(v) + b_t \Big) = \nabla\Psi_t^*(\tilde{\nabla}\Psi_t(\mathcal{M}_t v) + \mathcal{K}_t(v) + b_t), \tag{9}$$

where $\nabla\Psi_t^*$ *matches a variety of monotone activation operators. In addition, when the domains are all the same, say* $D_t = D$ *and the linear operator* $\mathcal{K}_t$ *is of the form given in Remark 5, then* $\mathcal{M}_t = \mathbb{1}$ *and*

$$\mathcal{L}_t(v) = \nabla\Psi_t^*(\tilde{\nabla}\Psi_t(v) + K_t v + \mathcal{K}_t^{(ac)}(v) + b_t). \tag{10}$$

Let us remark that when $\nabla\Psi^*$ corresponds to a strictly monotonic activation operator $\nabla\Psi^* = (\sigma \circ \cdot)$, then $\tilde{\nabla}\Psi = (\sigma^{-1} \circ \cdot)$ corresponds to the action of the inverse activation operator. The schematic representation of (10) is reported in Figure 1b. This novel variant, called *Bregman Neural Operator* simply differs from classical neural operators by the additional skip-like term involving the inverse activation operator. Finally, we note that the form of (10) corresponds to a mirror descent step [21, 2] with mirror map $\tilde{\nabla}\Psi_t$.

Remark 7.

(i) *When* $K_t v$*,* $\mathcal{K}_t^{(ac)}$ *and* b_t *are zeros, then* $\mathcal{L}_t$ *reduces to the identity.*
(ii) *One limitation is that it requires range*$(\mathcal{L}_{t-1}) \subset dom\partial\Psi_t$ *as mentioned in Remark 4* (ii)*. This condition is satisfied with the composition form of* (2)*, given that range*$(\mathcal{P}) \subset dom\partial\Psi_1$*.*
(iii) *The counterpart of Remark 6 for Bregman can be devised by adding the extra* $\tilde{\nabla}\Psi_t$ *term.*

3.4 Case of Fourier Neural Operators

We study the implications of the proposed viewpoint in the peculiar case of Hilbert function spaces with equal input and output spaces, i.e., $\mathcal{V}_t = \mathcal{V}_t^* = L^2(D, \mathbb{R}^m)$ for every $t \in \{1, \ldots, T\}$. A popularly encountered scenario in practice is that where $D = \mathbb{T}^d$ is the unit torus and the kernel associated to the absolutely continuous part of $\mathcal{K}_t$ is translation invariant, i.e., $k_t(x, y) = k_t(x - y)$, thus indicating a convolution structure. Fourier operator layers [17] are then devised by leveraging the convolution theorem, stating that the action of $\mathcal{K}_t^{(\mathrm{ac})}$ can be

written as a linear operator in the Fourier domain:

$$K_t^{(\mathrm{ac})}(v)(x) = \int_D k_t(x-y)v(y)dy = \mathcal{F}^{-1}(R_t \cdot \mathcal{F}(v))(x), \tag{11}$$

with $\mathcal{F}\colon L^2(\mathbb{T}^d, \mathbb{R}^m) \to \ell^2(\mathbb{Z}^d, \mathbb{R}^m)$ being the Fourier transform, $\mathcal{F}^{-1}$ its inverse, and $R_t \in \ell^2(\mathbb{Z}^2, \mathbb{R}^{m\times m})$. Often, R_t does not range in the entire $\ell^2(\mathbb{Z}^2, \mathbb{R}^{m\times m})$ space but is parametrized by a finite parameter [15]. It follows that the Bregman variant of Fourier operator layer reads $\mathcal{L}_t(v) = \sigma_t(\sigma_t^{-1}(v) + W_t v + \mathcal{F}^{-1}(R_t \cdot \mathcal{F}(v)) + b_t)$. The classical Fourier neural operator layer is retrieved by omitting the $\sigma_t^{-1}(v)$ term.

Remark 8. *The Bregman FNO mapping in can also be seen from the perspective of a regularized prediction operator [4, 3], where the affinity term is split in two:*

$$\mathcal{L}_t(v) = \operatorname{argmin}\Big\{ -\langle \cdot, +W_t(v) + b_t\rangle - \langle \mathcal{F}\cdot, \mathcal{F}R_t(v)\rangle_{\ell^2(\mathbb{Z}^2, \mathbb{R}^{m\times m})} + \Psi \Big\}. \tag{12}$$

4 Expressivity of Bregman neural operator networks

In this section we give a preliminary positive result concerning the universal approximation properties of Bregman neural operators.

In the following the activation function $\sigma\colon \mathbb{R} \to I$ is required to be a homeomorphism between $\mathbb{R}$ and an open interval I of $\mathbb{R}$ and of sigmoidal type, meaning that $\lim_{t\to-\infty} \sigma(t) = 0$ and $\lim_{t\to+\infty} \sigma(t) = 1$.

Theorem 3. *Let σ be as above. Suppose that $\mathcal{A}$ and $\mathcal{U}$ are Lebesgue spaces with exponents less than $+\infty$. Let $\mathcal{G}\colon \mathcal{A} \to \mathcal{U}$ be a continuous operator. Then for any compact set $K \subset \mathcal{A}$ and $\varepsilon > 0$ there exists a Bregman neural operator network $\mathcal{N}\colon \mathcal{A} \to \mathcal{U}$ of the type* (2) *such that*

$$\sup_{u\in K} \|\mathcal{G}(u) - \mathcal{N}(u)\|_{\mathcal{U}} \le \varepsilon.$$

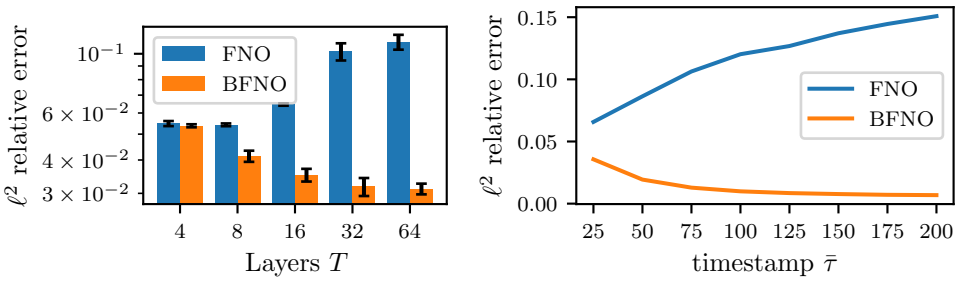
This theorem is based on the fact that we were able to prove this same result for Bregman neural networks in finite dimensional spaces.

5 Numerical Experiments

The primary objective of our numerical experiments is to evaluate and assess the added benefits of the Bregman variant of the simplest neural operator, namely Fourier Neural Operator (FNO) as it often serves as the building block for more sophisticated models.

5.1 Experimental Setting

Models. We consider the FNO [17] and its Bregman variant (BFNO), described in Section 3.4. The lifting and projection layers, namely $\mathcal{P}$ and $\mathcal{Q}$ in (2), are convolutional layers with kernel size 1 and width 128. Note that, for BFNO, we add an activation operator after $\mathcal{P}$ to ensure that the conditions of Remark 4 (ii) are met. Following the code of [17], we use the ReLU activation for FNO while, for BFNO, we resort to an invertible approximation: SoftPlus with parameter $\beta = 10^3$ to make it almost indistinguishible from ReLU. Hereafter, we consider models made of $T \in \{4, 8, 16\}$ Fourier layers with a width 64 (resp. 32) and 16 (resp. 12) maximum number of Fourier modes for 1D (resp. 2D) problems.
Datasets. We have selected a range of benchmark datasets resulting from the resolution of PDEs used both in the original FNO paper [17] and in the PDEBench suite [26]. They represent various dynamics and complexities pertinent to physical modeling tasks. Hereafter, we consider initial value problems where the goal is to learn the mapping between the initial condition a_i and the solution at some future time u_i from $n = 10^4$ pairs $\{a_i, u_i\}_{i=1}^n$. A detailed description of the experimental setting for each PDE and the learning procedure is provided in Section C of the appendix.



(a) Performance comparison (b) Performance in autoregressive mode

Figure 2: Results on 1D Burgers ($\nu = 10^{-3}$)

5.2 Results and Analysis

Illustration and impact of the number of layers T. First, we illustrate the behavior of the prediction error as the number of operator layers T increases. To this end, we conducted an experiment using the Burgers' dataset with viscosity $\nu = 10^{-3}$, with results presented in Figure 2a. First, we observe that BFNO systematically yields lower prediction error, irrespectively of T. Second, the performance of FNO degrades starting from $T = 16$, while BFNO demonstrates better performance as T increases until it reaches a plateau at $T = 64$. We believe that this interesting property is due to the added skip-like term of BFNO which helps in stabilizing the learning since BFNO layers reduce to the identity when all the weights are zero. In Figure 4, we report one instance of an input-output pair and the best predicted output by FNO and BFNO, showing that BFNO better predicts the sharp edges. An analysis of the layer-wise behavior of the weights is discussed in Section D.3

Learning the solution map. As previously mentioned, we consider the problem of learning the mapping between the initial condition and the solution of a PDE at some future time. In Table 2, we compare the prediction performance, in terms of ℓ^2 relative error, between FNO and BFNO for $T = \{4, 8, 16\}$ layers across different PDEs of varying complexities. Results indicate that BFNO consistently yields better or comparable prediction performance. Additionally, the behavior observed with the Burgers' PDE, where the performance improves or stabilizes without degrading as T increases, also holds for other PDEs. In contrast, FNO may suffer from a degradation of performance. An extended version of Table 2 is provided in the appendix, where the prediction performance is also analyzed both in frequency bands and on the boundary of the domain, leading to similar conclusions. This behavior highlights BFNO's capacity to avoid issues that arise from deep models, such as overfitting.

Learning the time-step evolution map. We now consider the problem of learning the mapping between the solution at some time t and the solution at $t + 25$. Then we pose our model in an autoregressive mode, where the output is fed again to the input of the model, repeating it 8 times. Results provided in Figure 2b show that BFNO actually benefits from better prediction at long horizons. Moreover, it seems that the BFNO is capable of correcting the error generated in the firsts horizons.

6 Conclusion

In summary, our contributions are twofold: we have provided a novel theoretical framework that broadens the understanding of neural operators through the lens of a Bregman regularized optimization problem, and we have introduced Bregman neural operators that achieve enhanced performance as their depth increases.

References

[1] A. Anandkumar, K. Azizzadenesheli, K. Bhattacharya, N. Kovachki, Z. Li, B. Liu, and A. Stuart. Neural operator: Graph kernel network for partial differential equations. In

Table 2: Relative error of FNO and BFNO models on benchmark PDEs.

	4 layers		8 layers		16 layers	
	FNO	BFNO	FNO	BFNO	FNO	BFNO
1D Advection	$1.0 \pm 0.0\%$	$\mathbf{0.7} \pm 0.0\%$	$1.4 \pm 0.1\%$	$\mathbf{0.6} \pm 0.1\%$	$1.8 \pm 0.1\%$	$\mathbf{0.6} \pm 0.1\%$
1D Burgers ($\nu = 10^{-1}$)	$0.5 \pm 0.0\%$	$\mathbf{0.3} \pm 0.0\%$	$0.7 \pm 0.0\%$	$\mathbf{0.3} \pm 0.0\%$	$0.9 \pm 0.0\%$	$\mathbf{0.4} \pm 0.0\%$
1D Burgers ($\nu = 10^{-3}$)	$5.5 \pm 0.1\%$	$\mathbf{5.4} \pm 0.1\%$	$5.4 \pm 0.1\%$	$\mathbf{4.1} \pm 0.2\%$	$6.5 \pm 0.1\%$	$\mathbf{3.5} \pm 0.2\%$
2D NS ($\nu = 10^{-3}$)	$4.6 \pm 0.1\%$	$\mathbf{4.3} \pm 0.1\%$	$4.1 \pm 0.1\%$	$\mathbf{4.0} \pm 0.0\%$	$\mathbf{3.9} \pm 0.1\%$	$4.0 \pm 0.1\%$
2D NS ($\nu = 10^{-4}$)	$\mathbf{13.5} \pm 0.1\%$	$13.7 \pm 0.1\%$	$13.0 \pm 0.2\%$	$\mathbf{12.6} \pm 0.1\%$	$12.6 \pm 0.1\%$	$\mathbf{12.2} \pm 0.1\%$
1D NS	$58.2 \pm 0.6\%$	$\mathbf{57.0} \pm 0.6\%$	$58.2 \pm 0.6\%$	$\mathbf{56.8} \pm 0.8\%$	$59.7 \pm 0.6\%$	$\mathbf{56.5} \pm 0.6\%$
2D Darcy	$34.6 \pm 0.0\%$	$\mathbf{33.4} \pm 0.2\%$	$32.8 \pm 0.2\%$	$\mathbf{31.5} \pm 0.4\%$	$32.9 \pm 0.2\%$	$\mathbf{30.0} \pm 0.5\%$

ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, 2020.

[2] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[3] M. Blondel, F. Llinares-López, R. Dadashi, L. Hussenot, and M. Geist. Learning energy networks with generalized fenchel-young losses. *Advances in Neural Information Processing Systems*, 35:12516–12528, 2022.

[4] M. Blondel, A. F. Martins, and V. Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

[5] H. Brezis. *Functional Analysis, Sobolev Spaces, and Partial Differential Equations*. Springer, New York, 2011.

[6] M. N. Bùi and P. L. Combettes. Bregman forward.backward operator splitting. *Set-Valued and Variational Analysis*, 29:583–603, 2021.

[7] P. L. Combettes and J.-C. Pesquet. Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, 28(3):491–518, Feb. 2020.

[8] G. Cybenko. Approximation by superposition of sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[9] J. Frecon, G. Gasso, M. Pontil, and S. Salzo. Bregman neural networks. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6779–6792. PMLR, 17–23 Jul 2022.

[10] S. Goswami, M. Yin, Y. Yu, and G. E. Karniadakis. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Computer Methods in Applied Mechanics and Engineering*, 391:114587, mar 2022.

[11] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu. GNOT: A general neural operator transformer for operator learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12556–12569. PMLR, 23–29 Jul 2023.

[12] J. Helwig, X. Zhang, C. Fu, J. Kurtin, S. Wojtowytsch, and S. Ji. Group equivariant fourier neural operators for partial differential equations. In *International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[13] G. Kissas, J. H. Seidman, L. F. Guilhoto, V. M. Preciado, G. J. Pappas, and P. Perdikaris. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

[14] N. Kovachki, S. Lanthaler, and S. Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

[15] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[16] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, K. Bhattacharya, and A. Anandkumar. Multipole graph neural operator for parametric partial differential equations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766. Curran Associates, Inc., 2020.

[17] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[18] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021.

[19] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, mar 2021.

[20] R. Molinaro, Y. Yang, B. Engquist, and S. Mishra. Neural inverse operators for solving pde inverse problems. In *International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[21] A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[22] Q. Nguyen. Forward-backward splitting with bregman distances. *Vietnam J. Math*,, 45:519–539, 2017.

[23] M. A. Rahman, Z. E. Ross, and K. Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023.

[24] B. Raonic, R. Molinaro, T. De Ryck, T. Rohner, F. Bartolucci, R. Alaifari, S. Mishra, and E. de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 77187–77200. Curran Associates, Inc., 2023.

[25] L. Serrano, L. Le Boudec, A. Kassaï Koupaï, T. X. Wang, Y. Yin, J.-N. Vittaut, and P. Gallinari. Operator learning with neural fields: Tackling pdes on general geometries. In *Advances in Neural Information Processing Systems*, volume 36, pages 70581–70611, 2023.

[26] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert. Pdebench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[27] T. Tripura and S. Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, feb 2023.

[28] S. Wang and P. Perdikaris. Long-time integration of parametric evolution equations with physics-informed DeepONets. *Journal of Computational Physics*, 475:111855, feb 2023.

[29] S. Wang, H. Wang, and P. Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40), oct 2021.

[30] S. Wang, H. Wang, and P. Perdikaris. Improved architectures and training algorithms for deep operator networks. *Journal of Scientific Computing*, 92(2), jun 2022.

[31] C. Zalinescu. *Convex Analysis in General Vector Spaces*. World Scientific, Singapore, 2002.

A Supplementary Mathematical Preliminaries

A.1 Additional Considerations for Bregman Proximal Operators

Proposition 4. *Let $\phi \in \Gamma_0(\mathbb{R}^m)$ be a Legendre function, let $p \in [1, +\infty[$, and suppose that ϕ is p-uniformly convex with constant $c > 0$, meaning that*

$$\forall\, y, y' \in \mathbb{R}^m, \forall\, \lambda \in]0,1[:\ \phi((1-\lambda)y + \lambda y') + \lambda(1-\lambda)\frac{c}{p}|y - y'|^p \leq (1-\lambda)\phi(y) + \lambda\phi(y'). \quad (13)$$

Let $\mathcal{V} = L^p(D, \mathbb{R}^m)$. Then the integral functional $\Phi\colon \mathcal{V} \to]-\infty, +\infty]$ defined as in Fact 1 is p-uniformly convex with respect to the norm $\|\cdot\|_p$. Moreover, for every $g \in \Gamma_0(\mathcal{V})$ such that $\operatorname{dom}\Phi \cap \operatorname{dom} g \neq \varnothing$, we have $\operatorname{dom}(\Phi + g)^* = \mathcal{V}^*$ and $(\Phi + g)^*$ is Fréchet differentiable on $\mathcal{V}^*$.

Proof. It follows by integrating (13). The second part follows by [31, Theorem 3.5.10], considering that $\Phi + g$ is also p-uniformly continuous. □

A.2 Link Between Activation Function and Proximity Operator

As demonstrated in the work of [7], many activation functions ρ can be expressed as proximity operators $\operatorname{prox}_g = \operatorname{argmin}_{t\in\mathbb{R}} g(t) + \frac{1}{2}(\cdot - t)^2$ for some appropriate convex function g. The simplest case is that of the ReLu activation function, recalled below.

Example 1 (ReLu)**.** *The rectified linear unit function $\rho\colon t \in \mathbb{R} \mapsto \max(t, 0) \in \mathbb{R}$ can be expressed as the proximity operator prox_g of $g = \iota_{[0,+\infty[}$. Henceforth, prox_g reduces to the projection onto the positive orthant.*

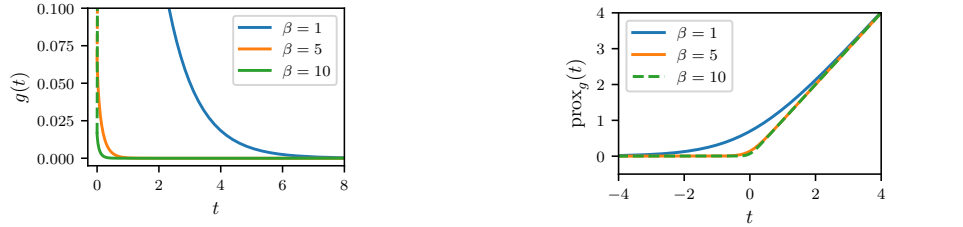
We also provide a novel characterization of SoftPlus.

Example 2 (SoftPlus)**.** *Given $\beta > 0$, the SoftPlus activation function, i.e., $\rho\colon t \mapsto \operatorname{SoftPlus}_\beta(t) \triangleq (1/\beta)\log(\exp(\beta t) + 1)$, is the proximity operator of*

$$g\colon t \in \mathbb{R}_{>0} \mapsto \frac{1}{\beta^2}\operatorname{Li}_2(\mathrm{e}^{-\beta t}) \in \mathbb{R}_{>0}, \quad (14)$$

where Li_2 is the dilogarithm function defined as $\operatorname{Li}_2\colon t \mapsto -\int_0^t \frac{\log(1-u)}{u}\mathrm{d}u$.

Proof. For every $s \in \mathbb{R}$, $\operatorname{prox}_g(s) = \operatorname{argmin}_{t\in\mathbb{R}}\{h(t) \triangleq g(t) + (1/2)(s-t)^2\}$ with $h(t) = (1/\beta^2)\operatorname{Li}_2(\mathrm{e}^{-\beta t}) + (1/2)(s-t)^2 = \psi(t) - st + (1/2)s^2$ where we introduced $\psi(t) = (1/\beta^2)\left(\operatorname{Li}_2(\mathrm{e}^{-\beta t}) + (1/2)\log(\mathrm{e}^{-\beta t})^2\right) = (1/\beta^2)\int^{\mathrm{e}^{-\beta t}} \log(r/(1-r))/r\mathrm{d}r$. The latter can be written as $\psi(t) = (1/\beta)\int^t \log(\mathrm{e}^{\beta r} - 1)\mathrm{d}r$ up to a constant. Finally, since h is strongly convex, the minimum is attained for t such that $h'(t) = 0$, which yields $\log(\mathrm{e}^{\beta t} - 1) = \beta s \Leftrightarrow t = \rho(s)$, thus ending the proof. □



(a) Representation of $t \mapsto g(t) = \frac{1}{\beta^2}\operatorname{Li}_2(\mathrm{e}^{-\beta t})$

(b) Representation of $\operatorname{prox}_g(t) = \operatorname{SoftPlus}_\beta(t)$

Figure 3: **Illustration of SoftPlus as a proximity operator.**

We present an illustration of the convex function g defined in Eq. 14 in Figure 3a. Intuitively, it serves as a smooth surrogate for the indicator function of the positive orthant $\iota_{[0,+\infty[}$. A

larger value of $\beta > 0$ leads to a closer approximation. This aligns with the representation of SoftPlus as the proximity operator of g from Eq. 14, depicted in Fig. 3b where a larger β makes SoftPlus closer to ReLU.

B Approximation results

We consider first shallow Bregman neural networks for finite dimensional spaces. Let $\sigma\colon \mathbb{R} \to I$ be a homeomorphism where I is an open interval in $\mathbb{R}$. We set

$$\mathsf{BN}_2(\sigma; I^d) = \operatorname{span}\Big\{\sigma(\sigma^{-1}(m^\top x) + w^\top x + b) \,\Big|\, m \in \Delta^{d-1}, w \in \mathbb{R}^d, b \in \mathbb{R}\Big\}. \tag{15}$$

Remark 9. *Since m belongs in the standard simplex Δ^{d-1}, $m^\top x$ is a convex combination of elements of I and so it is an element of I. Thus, since $\sigma^{-1}\colon I \to \mathbb{R}$, the functions in $\mathsf{BN}_2(\sigma; I^d)$ are well-defined.*

The following result follows from an adaptation of the argument in [8] to our different architecture (15).

Theorem 5. *Suppose that σ is* sigmoidal, *meaning that $\lim_{t\to-\infty}\sigma(t) = 0$ and $\lim_{t\to+\infty}\sigma(t) = 1$. Then, the space $\mathsf{BN}_2(\sigma; I^d)$ is dense in $\mathcal{C}(I^d, \mathbb{R})$ with respect to the topology of uniform convergence on compacta.*

Proof. Let $K \subset I^d$ be a compact set we prove that the trace space $\mathsf{BN}_2(\sigma; I^d)_{|K}$ is dense in $\mathcal{C}(K, \mathbb{R})$. To that purpose we rely on the following general fact concerning dense sets in Banach space (see, e.g., [5]). Let $\mathcal{B}$ be a Banach space, let $\mathcal{A} \subset \mathcal{B}$. Then the following propositions are equivalent.

- $\operatorname{span}\mathcal{A}$ is dense in $\mathcal{B}$
- $\mathcal{A}^\perp = \{u^* \in \mathcal{B}^* \,|\, \forall\, u \in \mathcal{A}\colon \langle u, u^*\rangle = 0\} = \{0\}$.
- $\forall\, u^* \in \mathcal{B}^*$, $(\forall\, u \in \mathcal{A}\colon\ \langle u, u^*\rangle = 0) \ \Rightarrow\ u^* = 0$.

This implies that for our purpose we can prove that

$$\forall\, \mu \in \mathcal{M}(K),\ \Big(\forall\, \varphi \in \mathsf{BN}_2(\sigma; I^d)\colon \int_K \varphi\mu = 0\Big) \ \Rightarrow\ \mu = 0,$$

where $\mathcal{M}(K)$ is the space of signed finite Radon measures on K (the dual of $\mathcal{C}(K)$). Thus, let μ be a signed measure on K and suppose that

$$\forall\, \varphi \in \mathsf{BN}_2(\sigma; I^d)\colon \quad \int_K \varphi\, d\mu = 0. \tag{16}$$

Fix $w \in \mathbb{R}^d, m \in \Delta^{d-1}$, and $b \in \mathbb{R}$. Define, for every $\lambda > 0$ and $c \in \mathbb{R}$

$$\sigma_{\lambda,c}\colon I \to \mathbb{R}, \quad x \mapsto \sigma(\sigma^{-1}(m^\top x) + \lambda(w^\top x + b) + c).$$

It is clear that $\sigma_{\lambda,c} \in \mathsf{BN}_2(\sigma; I^d)$. Moreover,

$$\lim_{\lambda\to+\infty} \sigma_{\lambda,c}(x) = \left\{ \begin{array}{ll} 1 & \text{if } w^\top x + b > 0 \\ 0 & \text{if } w^\top x + b < 0 \\ \sigma(\sigma^{-1}(m^\top x) + c) & \text{if } w^\top x + b = 0. \end{array} \right\} := \gamma(x).$$

Define the sets

$$\Pi^+_{w,b} = \Big\{x \in K \,\Big|\, w^\top x + b > 0\Big\}, \quad \Pi^-_{w,b} = \Big\{x \in K \,\Big|\, w^\top x + b < 0\Big\}, \quad \Pi_{w,b} = \Big\{x \in K \,\Big|\, w^\top x + b = 0\Big\}.$$

They are intersections of half-spaces and hyperplanes with K. So,

$$\gamma(x) = \chi_{\Pi^+_{w,b}}(x) + \sigma(\sigma^{-1}(m^\top x) + c)\chi_{\Pi_{w,b}}(x),$$

where χ_A is the characteristic functions of the set $A \subset I^d$. Since σ is bounded we can apply the Lebesgue's dominated convergence theorem and get

$$\lim_{\lambda \to +\infty} \underbrace{\int_K \sigma_{\lambda,c} d\mu}_{=0} = \int_K \gamma \, d\mu = \mu(\Pi^+_{w,b}) + \int_{\Pi_{w,b}} \sigma(\sigma^{-1}(m^\top x) + c) d\mu(x).$$

Note that the integral on the left is zero by the hypothesis (16). In this way we proved that

$$\forall\, m \in \Delta^{d-1}, \forall\, w \in \mathbb{R}^d, \forall\, b, \forall\, c \in \mathbb{R}: \quad \mu(\Pi^+_{w,b}) + \int_{\Pi_{w,b}} \sigma(\sigma^{-1}(m^\top x) + c) d\mu(x) = 0. \tag{17}$$

Now observe that (17) implies

$$\left|\mu(\Pi^+_{w,b})\right| = \left| \int_{\Pi_{w,b}} \sigma(\sigma^{-1}(m^\top x) + c) d\mu(x) \right| \leq \int_{\Pi_{w,b}} |\sigma(\sigma^{-1}(m^\top x) + c)| d|\mu|(x) \to 0 \text{ as } c \to -\infty,$$

since $|\sigma(\sigma^{-1}(m^\top x) + c)| \to 0$ as $c \to -\infty$ (pointwise). Therefore, $\mu(\Pi^+_{w,b}) = 0$. Then (17) yields

$$\forall\, c \in \mathbb{R}: \quad \int_{\Pi_{w,b}} \sigma(\sigma^{-1}(m^\top x) + c) d\mu(x) = 0.$$

Moreover, by assumption $\sigma(\sigma^{-1}(m^\top x) + c) \to 1$ as $c \to +\infty$ (pointwise) and hence, again by Lebesgue's dominated convergence theorem,

$$\lim_{c \to +\infty} \underbrace{\int_{\Pi_{w,b}} \sigma(\sigma^{-1}(m^\top x) + c) d\mu(x)}_{=0} = \int_{\Pi_{w,b}} 1 \, d\mu = \mu(\Pi_{w,b}),$$

which yields $\mu(\Pi_{w,b})$. In the end we proved that the measure μ is zero on all the sets of type

$$\Pi_{w,b} \quad \text{and} \quad \Pi^+_{w,b}.$$

Now the proof continues as in [8, Lemma 1], and we can conclude that $\mu = 0$. □

Now we address the vectorial case. We denote by $\mathsf{BN}_2(\sigma; I^d, \mathbb{R}^k)$ the space

$$\left\{ Q\sigma(\sigma^{-1}(Mx) + Wx + b) \,\middle|\, r \in \mathbb{N}_+, Q \in \mathbb{R}^{k \times r}, W, M \in \mathbb{R}^{r \times d}, \text{with } M \text{ right stochastic, and } b \in \mathbb{R}^r \right\},$$

where σ and σ^{-1} are applied component-wise.

Corollary 6. *We have that*

$$\mathsf{BN}_2(\sigma; I^d, \mathbb{R}^k) = (\mathsf{BN}_2(\sigma; I^d))^k := \underbrace{\mathsf{BN}_2(\sigma; I^d) \times \cdots \times \mathsf{BN}_2(\sigma; I^d)}_{k \text{ times}} \tag{18}$$

and it is dense in $\mathcal{C}(I^d, \mathbb{R}^k)$, in the topology of uniform convergence on compact sets.

Proof. In view of Theorem 5, it is clear that $(\mathsf{BN}_2(\sigma; I^d))^k$ is dense in $\mathcal{C}(I^d, \mathbb{R})^k \cong \mathcal{C}(I^d, \mathbb{R}^k)$ in the topology of uniform convergence on compact sets. Let's prove equality (18). The inclusion $\mathsf{BN}_2(\sigma; I^d, \mathbb{R}^k) \subset (\mathsf{BN}_2(\sigma; I^d))^k$ is immediate. Let $\varphi \colon I^d \to \mathbb{R}^k$ with components $\varphi_j \in \mathsf{BN}_2(\sigma; I^d)$, $j = 1, \ldots, k$. Then, there exists $r \in \mathbb{N}_+$, and for each $j \in \{1, \ldots, k\}$, $q_j \in \mathbb{R}^r$, $W_j \in \mathbb{R}^{r \times d}$, $b_j \in \mathbb{R}^r$, and $M_j \in \mathbb{R}^{r \times d}$ right stochastic matrix (the rows are positive and sum one), such that

$$\varphi_j(x) = q_j^\top \sigma(\sigma^{-1}(M_j^\top x) + W_j^\top x + b_j).$$

Then considering the block matrices

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_k \end{bmatrix} \in \mathbb{R}^{kr \times d}, \quad W = \begin{bmatrix} W_1 \\ \vdots \\ W_k \end{bmatrix} \in \mathbb{R}^{kr \times d}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix} \in \mathbb{R}^{kr}, \quad Q = \begin{bmatrix} q_1^\top & 0 & \cdots & 0 \\ 0 & q_2^\top & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_k^\top \end{bmatrix} \in \mathbb{R}^{k \times kr},$$

we have

$$\varphi(x) = Q\sigma(\sigma^{-1}(Mx) + Wx + b),$$

and hence $\varphi \in \mathsf{BN}_2(\sigma; I^d, \mathbb{R}^k)$. The statement follows. □

A general deep Bregman neural network with T layers is defined as follow

$$\mathsf{BN}_T(\sigma; I^d, \mathbb{R}^k) = \Big\{ W_T \circ L_{T-1} \circ \cdots \circ L_1 \Big\},$$

where, for every $t = 1, \ldots, T-1$,

$$L_t \colon I^{d_{t-1}} \to I^{d_t}, \quad x \mapsto \sigma(\sigma^{-1}(M_t x) + W_t x + b_t),$$

with $W_t \in \mathbb{R}^{d_t \times d_{t-1}}$, $b_t \in \mathbb{R}^{d_t}$ and $M_t \in \mathbb{R}^{d_t \times d_{t-1}}$ right stochastic, for $t = 1, \ldots, T-1$, with $d_0 = d$ and $W_T \in \mathbb{R}^{k \times d_{T-1}}$. Note that also the dimensions $d_1, \ldots, d_{T-1}$ can be chosen freely. Clearly for a deep network with $T > 2$, if we take, for every $t = 2, \ldots, T-1$, $d_t = d_1$, $W_t = 0$, $b_t = 0$, and M_t equals to the identity, then the layers L_t with $t = 2, \ldots, T-1$ act as the identity operator and hence

$$\mathsf{BN}_2(\sigma; I^d, \mathbb{R}^k) \subset \mathsf{BN}_T(\sigma; I^d, \mathbb{R}^k).$$

Therefore, $\mathsf{BN}_T(\sigma; I^d, \mathbb{R}^k)$ is dense in $\mathcal{C}(I^d, \mathbb{R}^k)$ for the topology of uniform convergence on compact sets.

Remark 10. *Often in applications it is desirable to have functions defined on the entire space $\mathbb{R}^d$. In this case one can simply precompose the functions in $\mathsf{BN}_T(\sigma; I^d, \mathbb{R}^k)$ by the homeomorphism*

$$x \in \mathbb{R}^d \to \sigma(x) \in I^d$$

and obtaining a dense set in $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)$. Such space is then denoted by $\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)$.

Let $D \subset \mathbb{R}^d$ be any nonempty bounded open set. If $\mathcal{F}(\mathbb{R}^d)$ is any class of real functions from $\mathbb{R}^d$ to $\mathbb{R}$ we denote by $\mathcal{F}_{|\overline{D}}$ the set of restrictions to $\overline{D}$ of the functions in $\mathcal{F}(\mathbb{R}^d)$. In the following according to Remark 10 we put

$$\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k) = \Big\{ W_T \circ L_{T-1} \circ \cdots \circ L_1 \circ \sigma \Big\},$$

which is a dense space in $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)$ with respect to the topology of uniform convergence on compact sets.

Lemma 7. *Suppose that σ is a sigmoidal activation function as in Theorem 5. Let $p \in [1, +\infty[$. Then $\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}}$ is dense in $L^p(D, \mathbb{R}^k)$ (in the norm $\|\cdot\|_p$).*

Proof. It is well known that $\mathcal{C}_c(D, \mathbb{R}^k)$ is dense in $L^p(D, \mathbb{R}^k)$ and hence $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}}$ is dense in $L^p(D, \mathbb{R}^k)$ (in the norm $\|\cdot\|_p$). Moreover, $\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}}$ is dense in $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}}$ (in the norm $\|\cdot\|_\infty$). On the other hand

$$\forall\, f \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}} \colon \quad \|f\|_p = \Big(\int_D |f|^p \, dx \Big)^{1/p} \leq \|f\|_\infty |D|^{1/p}.$$

Thus, if $f \in L^p(D, \mathbb{R}^k)$ and $\varepsilon > 0$,

$$\exists\, g \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)_{\overline{D}} \text{ s.t. } \|f - g\|_p \leq \frac{\varepsilon}{2}$$

$$\exists\, h \in \mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)_{|\overline{D}} \text{ s.t. } \|g - h\|_\infty \leq \frac{\varepsilon}{2|D|^{1/p}} \;\Rightarrow\; \|g - h\|_p \leq \frac{\varepsilon}{2}$$

and hence $\|f - h\|_p \leq \varepsilon$. □

Proof of Theorem 3. Since $\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)$ is dense in $\mathcal{C}(\mathbb{R}^d, \mathbb{R}^k)$ in the topology of uniform convergence on compact set, we can follows the flow of arguments in [15], by simply replacing the standard neural network $\mathsf{N}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)$ by $\mathsf{BN}_T(\sigma; \mathbb{R}^d, \mathbb{R}^k)$ □

C Experimental Settings

We adopt the same experimental setting as in the PDEBench repository [26]. For the sake of information, we recall the considered problems and PDEs and the specific settings we consider when appropriate. The learning procedure used is presented at the end of this section.

C.1 1D Advection Equation

The advection equation is a linear Partial Differential Equation (PDE) modeling the transport of a fluid quantity u, namely its velocity field, defined by the following equation:

$$\partial_t u(x,t) + \beta \partial_x u(x,t) = 0, \quad x \in (0,1), t \in (0,2], \tag{19}$$

$$u(x,0) = u_0(x), \ x \in (0,1), \tag{20}$$

with β a constant advection speed. Note that this system admits an exact solution: $u(t,x) = u_0(x - \beta t)$.

For this dataset, we follow the setting given in [26], Section D.1 by taking $\beta = 0.4$. We learn the mapping between the value of the field at $t = 0$ ($u(x,0)$) and the value at time $t = 2$ ($u(x,2)$), *i.e.* we learn the mapping between the first and the last temporal value of each sample.

C.2 1D Burgers Equation

The Burgers' equation is a PDE describing the nonlinear advection and diffusion of a velocity field, defined as follows:

$$\partial_t u(x,t) + \partial_x (u^2(x,t)/2) = \nu/\pi \partial_{xx} u(x,t), \quad x \in (0,1), t \in (0,2], \tag{21}$$

$$u(x,0) = u_0(x), \ x \in (0,1), \tag{22}$$

where ν is the diffusion coefficient, which is assumed to be constant in this dataset.

We follow again the setup presented in [26], section D.2, with $\nu = 0.001$. As in the previous dataset, we learn the mapping from the field at $t = 0$ as input to the field at $t = 2$ as target.

C.3 1D Compressible Navier-Stokes Equations (1D NS)

The compressible Navier-Stokes equations describe the motion of viscous fluids that can change in density due to compression or expansion. This can be described through the following partial differential equations:

$$\partial_t \rho + \partial_x \cdot (\rho \mathbf{u}) = 0, \tag{23}$$

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \partial_x \mathbf{u}) = -\partial_x p + \eta \triangle \mathbf{u} + (\zeta + \eta/3)\partial_{xx} \mathbf{u}), \tag{24}$$

$$\partial_t(\epsilon + \rho v^2/2) + \partial_x \cdot [(p + \epsilon + \rho v^2/2)\mathbf{u} - \mathbf{u} \cdot \sigma'] = \mathbf{0}, \tag{25}$$

where ρ is the mass density, $\mathbf{u} = \mathbf{u}(\mathbf{x}, \mathbf{t})$ is the fluid velocity, p is the gas pressure, ϵ is an internal energy described by the equation of state, σ' is the viscous stress tensor, and η and ζ are shear and bulk viscosity, respectively.

In our experiments, we consider the setup introduced in [26], Section D.5, fixing $\eta = 10^{-8}$, $\zeta = 10^{-8}$ and out-going boundary conditions. We learn the mapping of the velocity $\mathbf{v}$ from time $t = 10$ as input to time $t = 15$ as target. For this dataset, we added a symmetrical padding preprocessing to replicate periodic boundary conditions (as prescribed in the original FNO code [17]).

C.4 2D Incompressible Navier-Stokes Equations (2D NS)

We also consider a dataset from the 2D Navier-Stokes equation for a viscous, incompressible fluid in vorticity form on the unit torus [17] defined as follows:

$$\begin{aligned} \partial_t w(x,t) + u(x,t)\cdot\nabla w(x,t) &= \nu\Delta w(x,t) + f(x), && x\in(0,1)^2, t\in(0,T_{final}] \\ \nabla\cdot u(x,t) &= 0, && x\in(0,1)^2, t\in(0,T_{final}] \\ w(x,0) &= w_0(x), && x\in(0,1)^2 \end{aligned} \tag{26}$$

with u is the 2D velocity field, $w = \nabla\times u$ is the vorticity, $w_0 : (0,1)^2;\to\mathbb{R}$ is the initial vorticity function, $\nu\in\mathbb{R}_+$ is the viscosity coefficient, and $f : (0,1)^2\to\mathbb{R}$ is the forcing function.

We follow the setup introduced in [17], Section A.3.3, with $\nu = 10^{-3}$ and $\nu = 10^{-4}$. We learn the mapping of the velocity field $\mathbf{v}$ from sample time $t = 10$ to $t = 50$ for $\nu = 10^{-3}$ and from $t = 10$ to $t = 20$ for $\nu = 10^{-4}$.

C.5 Darcy Flow

We consider a dataset based on the steady state of the 2D Darcy Flow equation on the unit square, representing the flow through porous media and defined as follows:

$$\begin{aligned} -\nabla(a(x)\nabla u(x)) &= f(x), && x\in(0,1)^2, \\ u(x) &= 0, && x\in\partial(0,1)^2. \end{aligned} \tag{27}$$

We follow the setup described in [26], Section D.4, with $f(x)$ fixed to the constant $\beta = 0.1$.

C.6 Learning procedure

Models are trained using the Adam optimizer with a constant learning rate, a batch size of 128 for 1D problems (resp. 32 for 2D problems), a maximum of 2000 epochs and an early stopping strategy with patience of 100 epochs and $\delta = 10^{-3}$. The learning rate is validated on a grid of multiple values equally spaced in logarithmic scale. If not mentioned otherwise, we use 8000 (resp. 1000) training samples for 1D (resp. 2D) problems, and 1000 samples each for validation and testing. All results are averaged over four random splittings.

Experiments have been made on an internal clusters of GPUs with memory from 10Go to 45Go. All the experiments can be achieved with GPUs with a memory of 10Go, except for models with 32 or 64 layers which require at least a memory of 24Go.

D Additional Results

D.1 Comparison of Predictions

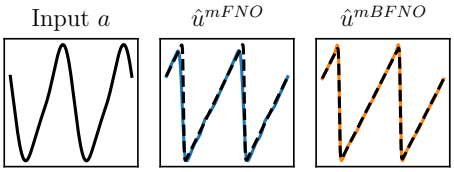


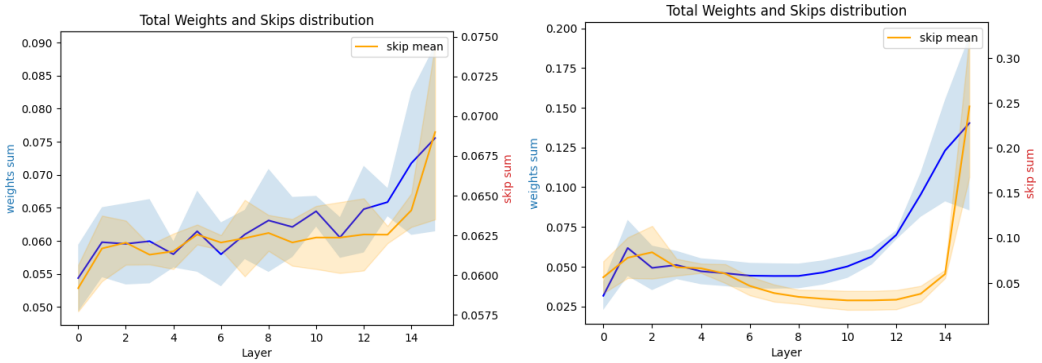
Figure 4: Prediction comparison. The ground truth output is displayed in dashed black.

D.2 Detailed Analysis of the Prediction Performance

Table 3: Additional comparison of the performance in terms of relative ℓ^2 error (rL2), relative mean squared error on the boundary (rMSE) as well as in the low, mid and high frequency bands (fRMSE low, fRMSE mid, fRMSE high). Note that here 2D NS corresponds to $\nu = 10^{-3}$.

		$T = 4$		$T = 8$		$T = 16$	
PDE	Metric	BFNO	FNO	BFNO	FNO	BFNO	FNO
1D advection	rL2	$1.03 \cdot 10^{-2}$	$6.82 \cdot 10^{-3}$	$6.43 \cdot 10^{-3}$	$1.36 \cdot 10^{-2}$	$6.43 \cdot 10^{-3}$	$1.81 \cdot 10^{-2}$
	bRMSE	$1.14 \cdot 10^{-1}$	$1.62 \cdot 10^{0}$	$1.16 \cdot 10^{-1}$	$3.87 \cdot 10^{0}$	$1.12 \cdot 10^{-1}$	$1.58 \cdot 10^{1}$
	fRMSE low	$7.10 \cdot 10^{-6}$	$7.37 \cdot 10^{-5}$	$7.59 \cdot 10^{-6}$	$2.33 \cdot 10^{-4}$	$7.62 \cdot 10^{-6}$	$1.36 \cdot 10^{-3}$
	fRMSE mid	$5.41 \cdot 10^{-6}$	$1.77 \cdot 10^{-5}$	$5.07 \cdot 10^{-6}$	$3.78 \cdot 10^{-5}$	$4.65 \cdot 10^{-6}$	$1.08 \cdot 10^{-4}$
	fRMSE high	$4.20 \cdot 10^{-7}$	$2.03 \cdot 10^{-6}$	$3.60 \cdot 10^{-7}$	$3.20 \cdot 10^{-6}$	$3.60 \cdot 10^{-7}$	$5.18 \cdot 10^{-6}$
1D Burgers	rL2	$5.37 \cdot 10^{-2}$	$5.48 \cdot 10^{-2}$	$4.14 \cdot 10^{-2}$	$5.42 \cdot 10^{-2}$	$3.51 \cdot 10^{-2}$	$6.45 \cdot 10^{-2}$
	bRMSE	$4.31 \cdot 10^{-1}$	$4.38 \cdot 10^{-1}$	$2.97 \cdot 10^{-1}$	$3.80 \cdot 10^{-1}$	$2.39 \cdot 10^{-1}$	$4.79 \cdot 10^{-1}$
	fRMSE low	$5.67 \cdot 10^{-5}$	$5.70 \cdot 10^{-5}$	$3.63 \cdot 10^{-5}$	$5.08 \cdot 10^{-5}$	$3.08 \cdot 10^{-5}$	$5.31 \cdot 10^{-5}$
	fRMSE mid	$3.49 \cdot 10^{-5}$	$3.44 \cdot 10^{-5}$	$2.70 \cdot 10^{-5}$	$3.58 \cdot 10^{-5}$	$2.38 \cdot 10^{-5}$	$3.79 \cdot 10^{-5}$
	fRMSE high	$1.17 \cdot 10^{-6}$	$1.20 \cdot 10^{-6}$	$1.07 \cdot 10^{-6}$	$1.24 \cdot 10^{-6}$	$9.80 \cdot 10^{-7}$	$1.23 \cdot 10^{-6}$
2D NS	rL2	$4.27 \cdot 10^{-2}$	$4.61 \cdot 10^{-2}$	$4.01 \cdot 10^{-2}$	$4.14 \cdot 10^{-2}$	$3.98 \cdot 10^{-2}$	$3.90 \cdot 10^{-2}$
	bRMSE	$3.87 \cdot 10^{-2}$	$4.16 \cdot 10^{-2}$	$3.63 \cdot 10^{-2}$	$3.76 \cdot 10^{-2}$	$3.61 \cdot 10^{-2}$	$3.54 \cdot 10^{-2}$
	fRMSE low	$4.05 \cdot 10^{-4}$	$4.33 \cdot 10^{-4}$	$3.72 \cdot 10^{-4}$	$3.82 \cdot 10^{-4}$	$3.80 \cdot 10^{-4}$	$3.63 \cdot 10^{-4}$
	fRMSE mid	$9.59 \cdot 10^{-5}$	$9.03 \cdot 10^{-5}$	$6.53 \cdot 10^{-5}$	$7.73 \cdot 10^{-5}$	$6.22 \cdot 10^{-5}$	$6.15 \cdot 10^{-5}$
	fRMSE high	$9.85 \cdot 10^{-6}$	$6.95 \cdot 10^{-6}$	$5.95 \cdot 10^{-6}$	$5.98 \cdot 10^{-6}$	$5.40 \cdot 10^{-6}$	$9.18 \cdot 10^{-6}$

D.3 Weight distribution



(a) FNO (b) BFNO

Figure 5: Illustration of the normalized weights distributions accross the layers for a $T = 16$ layers model trained on Burgers.