Component Fourier Neural Operator for Singularly Perturbed Differential Equations

Ye Li[1], Ting Du[2], Yiwen Pang[1], Zhongyi Huang[2]

[1]Nanjing University of Aeronautics and Astronautics
[2]Tsinghua University
yeli20@nuaa.edu.cn, dt20@mails.tsinghua.edu.cn, yiwenpang@nuaa.edu.cn, zhongyih@tsinghua.edu.cn

Abstract

Solving Singularly Perturbed Differential Equations (SPDEs) poses computational challenges arising from the rapid transitions in their solutions within thin regions. The effectiveness of deep learning in addressing differential equations motivates us to employ these methods for solving SPDEs. In this manuscript, we introduce Component Fourier Neural Operator (ComFNO), an innovative operator learning method that builds upon Fourier Neural Operator (FNO), while simultaneously incorporating valuable prior knowledge obtained from asymptotic analysis. Our approach is not limited to FNO and can be applied to other neural network frameworks, such as Deep Operator Network (DeepONet), leading to potential similar SPDEs solvers. Experimental results across diverse classes of SPDEs demonstrate that ComFNO significantly improves accuracy compared to vanilla FNO. Furthermore, ComFNO exhibits natural adaptability to diverse data distributions and performs well in few-shot scenarios, showcasing its excellent generalization ability in practical situations.

Introduction

Singularly perturbed differential equations (SPDEs) serve as fundamental mathematical models in diverse physical phenomena, including fluid flows and material sciences (Roos, Stynes, and Tobiska 2008). Formally, SPDEs can be regarded as a distinct class of differential equations with a small positive parameter ε appearing before the highest order derivative. These equations yield solutions that undergo rapid changes in thin regions, commonly referred to as boundary layers or inner layers, depending on their respective locations. Consequently, solving SPDEs poses significant challenges, both analytically and numerically.

With the surge of deep learning, efforts have been directed toward employing artificial neural networks for solving partial differential equations (PDEs) (Roos, Stynes, and Tobiska 2008), particularly in the field of physics-informed machine learning (Bar-Sinai et al. 2019; Greenfeld et al. 2019; Karniadakis et al. 2021). Notably, operator learning techniques like FNO (Li et al. 2020) and DeepONet (Lu et al. 2021) have gained attention for their ability to learn operators between infinite-dimensional functional spaces. However, when addressing SPDEs, standard methods like vanilla FNO/DeepONet, and other neural network-based PDE solvers, face challenges. The occurrence of thin layers within SPDE solutions introduces stiffness-related features in neural network models.

In this study, we present the Component Fourier Neural Operator (ComFNO) for SPDEs. ComFNO extends FNO, tailored to boundary or inner layer phenomena. Unlike the explicit inclusion of physical equations, our method integrates prior knowledge from asymptotic analysis directly into the neural network architecture. ComFNO inherently differentiates between smooth and layer parts of SPDE solutions, enabling separate implicit learning. Our demonstrations reveal enhanced predictive accuracy, even in few-shot cases, and adaptability across diverse data distributions. ComFNO's versatility positions it as a promising tool. Our contributions are threefold:

- We propose ComFNO, integrating asymptotic analysis insights into vanilla FNO's architecture.
- Our approach is not limited to FNO, offering a flexible framework applicable to SPDEs and compatible with alternative neural network architectures, like DeepONet.
- Experimental results across various SPDEs, encompassing one-dimensional, two-dimensional, and time-dependent equations, demonstrate significant reductions in mean, infinity norm, and residual error variance. This underscores the improved accuracy of ComFNO in addressing SPDEs, while its robustness is confirmed through empirical investigations involving multiple data distributions and few-shot cases.

Related Work

Physics-Informed Machine Learning

While pure data-driven machine learning has achieved breakthroughs in many fields, researchers have attempted to combine physical knowledge with machine learning to improve performance. Physics-informed machine learning integrating seamlessly data and mathematical physics models (e.g., PDEs), has been one of the hotspots in current researches (Karniadakis et al. 2021; Markidis 2021). This development, stemming from physics-informed neural networks (PINNs) and other neural network-based PDE solvers (Raissi, Perdikaris, and Karniadakis 2019; E and Yu 2018;

Bar-Sinai et al. 2019; Rackauckas et al. 2020; Yadav and Ganesan 2021), signifies a noteworthy evolution.

PINNs employ tailored objective functions for diverse differential equations, utilizing limited datasets and even unsupervised learning. They harness inherent physical insights but may lack thorough data distribution analysis. However, PINN training presents unique challenges compared to conventional supervised learning.(Krishnapriyan et al. 2021; Wang, Yu, and Perdikaris 2022).

Neural Operators

Research into operator learning has recently surged, aiming to approximate mappings between infinite-dimensional functional spaces (Goswami et al. 2022). This innovation eliminates the need for repetitive equation-solving with varying parameters, such as coefficients or source terms, promising significant speed-up compared to vanilla PINNs and traditional solvers. Contributions from the introduction of Deep Operator Network (DeepONet) (Lu et al. 2021) and Fourier Neural Operator (FNO) (Li et al. 2020) are noteworthy. DeepONet harnesses the universal approximation theorem for operators (Chen and Chen 1995), accompanied by rigorous convergence analyses (Deng et al. 2022; Lanthaler, Mishra, and Karniadakis 2022; De Ryck and Mishra 2022) and diverse applications (Lu et al. 2022; Wang, Wang, and Perdikaris 2021; Jin, Meng, and Lu 2022; Haghighat et al. 2021). However, it often demands an extensive dataset to enhance predictive performance. FNO boasts theoretical error estimates (Kovachki, Lanthaler, and Mishra 2021) and the capability to simulate various physical phenomena, including fluid flows (Li et al. 2021; Rosofsky and Huerta 2022), seismic waves (Yang et al. 2021), and material modeling (You et al. 2022). Although DeepONet and FNO perform comparably across several crucial PDEs (Lu et al. 2022), FNO is chosen in this study due to its superior cost-accuracy trade-off (De Hoop et al. 2022).

Numerical and Neural Networks for Solving SPDEs

Addressing SPDEs is a key concern in applied mathematics. Classical numerical techniques, including finite difference and finite element methods, have been extensively studied for such purposes (Roos, Stynes, and Tobiska 2008; Roos 2022). Yet, these methods require predefined grids, with accuracy tied to grid density. The rise of deep learning has led to the use of neural networks in tackling SPDEs (Tawfiq and Al-Abrahemee 2014). Improvements in network architectures and training strategies have enhanced SPDE-solving capabilities (Liu et al. 2020; Greenfeld et al. 2019; Simos and Famelis 2022; Beguinet et al. 2022). Nevertheless, neural network solvers for SPDEs reveal limitations in physics-informed machine learning methods like PINNs. Furthermore, these methods focus on equation solutions rather than solution operators. Developing operator solvers for SPDEs is particularly challenging due to the rapid transitions in solutions within thin regions.

Problem Settings and Preliminaries

Singularly Perturbed Differential Equations

Singularly perturbed differential equations (SPDEs) involve a small positive parameter, usually denoted as ε, which typically appears ahead of the highest-order derivative term. As ε approaches 0, the solution's derivative (or higher-order derivatives) can tend towards infinity within specific regions. Meanwhile, the solution (or its derivative) undergoes significant changes in these regions, while displaying regular behavior away from them. These regions are termed boundary layers or inner layers, depending on their relative positions.

We illustrate the concept using the one-dimensional convection-diffusion equation as an example—a straightforward yet highly significant category of SPDEs—presented in the following form:

$$\begin{cases} -\varepsilon u'' + b(x)u' + c(x)u = f(x), & x \in (0,1)\,, \\ u(0) = u(1) = 0, \end{cases} \tag{1}$$

where ε is a very small positive parameter. The higher-order term u'' delineates diffusion, while the first-order term u' signifies convection. Owing to the existence of this small parameter ε, the solution to such an equation tends to exhibit singular behavior, as demonstrated in Figure 1.

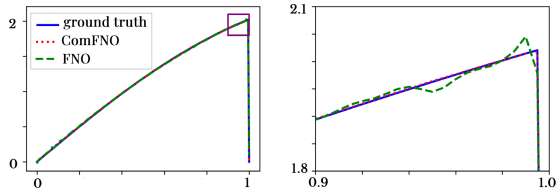


Figure 1: (left) Ground truth and predictions for FNO and ComFNO. Here we have $b(x) = x + 1$, $c(x) = 0$ and $\varepsilon = 0.001$ in Eq. (1). We take 900 distinct $f(x)$ for training and a random $f(x)$ for testing. (right) Zoomed in curves near the boundary layer. We can see that, the true solution of Eq. (1) has a boundary layer at $x = 1$.

Fourier Neural Operator

Fourier neural operator (FNO) is derived from the kernel neural operator, which replaces the operator with the Fourier operator (Li et al. 2020). The Fourier layer, constituting a fundamental building block of FNO, drives the updates $v_t \mapsto v_{t+1}$ as follows:

$$v_{t+1}(x) := \sigma\left(W v_t(x) + \left(\mathcal{K}(a;\theta)v_t\right)(x)\right).$$

Here, W represents a linear transformation, and σ denotes a nonlinear activation function. The parameterized neural network $\mathcal{K}(a;\theta)$ is characterized by the subsequent formulation:

$$\left(\mathcal{K}(\phi)v_t\right)(x) = \mathcal{F}^{-1}\left(R_\theta \cdot (\mathcal{F}v_t)\right)(x).$$

In this context, $\mathcal{F}$ stands for the Fourier transform, while $\mathcal{F}^{-1}$ signifies its inverse. These transformations are detailed

as:

$$(\mathcal{F}f)(k) = \int_D f(x)e^{-2i\pi\langle x,k\rangle}dx,$$
$$\left(\mathcal{F}^{-1}f\right)(x) = \int_D f(k)e^{2i\pi\langle x,k\rangle}dx.$$

Method

Asymptotic Analysis

Our framework builds upon the principles of singular perturbation theory and asymptotic expansion methods. The function u_{as} represents an asymptotic expansion of order m of u if there exists a constant C such that for all $x \in [0,1]$ and sufficiently small ε, the following inequality holds:

$$|u(x) - u_{as}(x)| \leq C\varepsilon^{m+1}.$$

In numerous classes of SPDEs, their solutions or asymptotic expansions often exhibit a decomposition into two distinct components. One part characterizes the solution's behavior within boundary or inner layers referred to as the "layer part"—while the other part pertains to the solution's behavior outside these regions, termed the "smooth part." Within this manuscript, our focus primarily centers on exponential-type layers, which occur frequently and exert substantial influence on the solution. For these exponential-type layers, the layer part shares a significant relationship with exponential functions, thereby furnishing pivotal insights that underpin our innovative framework.

Extensive research has focused on the asymptotic analysis of SPDEs (Roos, Stynes, and Tobiska 2008; Becher and Roos 2015). We will now present some notable contributions in this domain, with illustrative examples further elucidating boundary and inner layer phenomena in SPDEs available in the Appendix.

Ordinary Differential Equations To begin, we assume that all the forthcoming equations involve sufficiently smooth coefficient functions and source terms. The term "reduced solutions" refers to solutions of the reduced problems obtained by setting $\varepsilon = 0$ in the SPDEs.

Let's commence with a simple convection-diffusion equation:

$$\begin{cases} -\varepsilon u'' + b(x)u' + c(x)u = f(x), & x \in (0,1), \\ u(0) = u(1) = 0. \end{cases} \tag{2}$$

The solution of Eq. (2) typically exhibits an exponential boundary layer at $x = 1$ when $b(x) > 0$ on $[0,1]$, and a similar layer at $x = 0$ when $b(x) < 0$. In cases where $b(x)$ has zeros on $[0,1]$, we refer to it as the turning point problem, which will be addressed separately later. The conditions $b < 0$ and $b > 0$ are mutually equivalent, as the change of variable $x \mapsto 1-x$ transforms the problem from one formulation to the other. Focusing on the case when $b(x) \geq \beta > 0$, the solution u has an asymptotic expansion of order m in the following form:

$$u_{as}(x) = \sum_{\nu=0}^{m} \varepsilon^{\nu} u_{\nu}(x) + \sum_{\mu=0}^{m} \varepsilon^{\mu} v_{\mu}\left(\frac{1-x}{\varepsilon}\right). \tag{3}$$

The functions u_ν ($\nu = 0, 1, \ldots, m$) and v_μ ($\mu = 0, 1, \ldots, m$) in Eq. (3) are obtained through matched asymptotic expansion (Eckhaus 2011), a widely-used technique in asymptotic analysis. This method identifies u_0 as the reduced solution, and $v_0\left(\frac{1-x}{\varepsilon}\right) = -u_0(1)\exp\left(-b(1)\frac{1-x}{\varepsilon}\right)$, yielding an asymptotic expansion with the following estimate:

$$|u(x) - (u_0 + v_0)| \leq C\varepsilon, \tag{4}$$

prompting the integration of exponential operations in FNO.

In Eq. (4), u_0 represents the smooth part of the asymptotic expansion, capturing the function's smooth behavior across most regions except the boundary layer, while v_0 serves as the the layer part, acting as a correction within the boundary layer region. Beyond the asymptotic expansion, the solution u to Eq. (2) can be further decomposed into two parts: the smooth part denoted by S, which satisfies

$$|S(x)| \leq C,$$

and the layer part denoted by E, which satisfies

$$|E(x)| \leq C\varepsilon^{-l}\exp\left(-\frac{\beta(1-x)}{\varepsilon}\right).$$

In the context of turning point problems, isolated points where the coefficient of u' vanishes are referred to as turning points. In this study, we focus on the scenario of a single turning point located within the interior of the domain, without loss of generality, where the differential equation is defined on the interval $(-1,1)$ with the turning point situated at $x = 0$. Thus, we investigate the following equation by assuming $b(x) \neq 0$:

$$\begin{cases} -\varepsilon u'' + xb(x)u' + c(x)u = f(x), & x \in (-1,1), \\ u(-1) = u(1) = 0. \end{cases} \tag{5}$$

It is crucial to emphasize that the solution $u(x)$ may demonstrate singular behavior at the turning point $x = 0$ and the boundary points $x = -1$ and $x = 1$ (for further details, refer to the Appendix). For our analysis, we consider the case where $b(x) \geq \beta > 0$ on $[-1,1]$, resulting in the emergence of two distinct boundary layers at $x = -1$ and $x = 1$. For this particular case, the solution's asymptotic expansion is given by

$$u_{as} = u_0 + v_0 + w_0, \text{ with } |u(x) - u_{as}(x)| \leq C\varepsilon,$$

where u_0 is the reduced solution, v_0 and w_0 are defined as follows:

$$v_0(x) = (u(1) - u_0(1))\exp\left(-b(1)\frac{1-x}{\varepsilon}\right),$$
$$w_0(x) = (u(-1) - u_0(-1))\exp\left(b(-1)\frac{1+x}{\varepsilon}\right).$$

Partial Differential Equations In the context of parabolic partial differential equations in the space-time domain $Q = (0,1) \times (0,T]$, the initial-boundary value problem is described by the following equation:

$$\begin{cases} u_t - \varepsilon u_{xx} + b(x,t)u_x + d(x,t)u = f(x,t), & (x,t) \in Q, \\ u(x,0) = s(x), & 0 \leq x \leq 1, \\ u(0,t) = q_0(t), & 0 < t \leq T, \\ u(1,t) = q_1(t), & 0 < t \leq T. \end{cases} \tag{6}$$

In cases where $b > 0$, the solution u displays smooth behavior across most of domain Q. However, near the boundary $x = 1$ of Q, the solution typically manifests a boundary layer. For fixed $t > 0$, this layer's behavior relative to x resembles that in Eq. (2). In this context, the solution $u(x,t)$ can be decomposed as:

$$u(x,t) = u_0(x,t) - \tilde{u}_0(1,t)e^{-b(1,t)(1-x)/\varepsilon} + w(x,t).$$

Here, u_0 is the reduced solution, $|\tilde{u}_0| = \mathcal{O}(1)$, and $|w(x,t)| \leq C\varepsilon^{1/2}$.

For a boundary value problem of an elliptic partial differential equation in the spatial domain $\Omega = (0,1) \times (0,1)$, given by

$$\begin{cases} -\varepsilon\Delta u + \mathbf{b}(x,y) \cdot \nabla u + c(x,y)u = f(x,y), & \text{in } \Omega, \\ u(x,y) = 0, & \text{on } \partial\Omega. \end{cases} \tag{7}$$

Under the assumption of $\mathbf{b} = (b_1, b_2) > 0$ (specifically, with $b_1 > 0$ and $b_2 > 0$), the emergence of two exponential boundary layers is evident, situated at both $x = 1$ and $y = 1$. The interplay of these two boundary layers at the coordinate $(1,1)$ necessitates the introduction of a corner layer correction. The asymptotic expansion of u is formulated as:

$$\begin{aligned} u_{as}(x,y) &:= u^*_{as}(x,y) + v^*_{as}(x,y), \\ u^*_{as}(x,y) &:= u_0(x,y) - u_0(1,y)e^{-b_1(1,y)\frac{1-x}{\varepsilon}} \\ &\quad - u_0(x,1)e^{-b_2(x,1)\frac{1-y}{\varepsilon}}. \\ v^*_{as}(x,y) &:= u_0(1,1)e^{-b_1(1,1)b_2(1,1)\frac{1-x}{\varepsilon}\frac{1-y}{\varepsilon}}. \end{aligned} \tag{8}$$

Here, u_0 is the reduced solution and the following estimation holds:

$$|u(x) - u_{as}(x)| \leq C\varepsilon.$$

Improved Network Structure

In the preceding section, we introduced asymptotic expansions and solution decompositions for diverse classes of SPDEs. These solutions, including their asymptotic expansions, can be partitioned into two components: the smooth part and the layer part. Remarkably, their solutions exhibit exponential boundary layers, with the layer parts linked to exponential functions. Specifically, it is evident that in cases where the spatial dimension is one-dimensional, encompassing ordinary differential equations (ODEs) such as Eq. (2), (5), and partial differential equations (PDEs) like Eq. (6), the layer part demonstrates behavior akin to $\exp(-c(x_0 - x)/\varepsilon)$ as x approaches x_0. Here, x_0 signifies the location of a boundary or inner layer, and c represents a constant for ODEs and a function for PDEs. For instance, in Eq. (5), the solution reveals two exponential boundary layers at $x = -1$ and $x = 1$, with the layer parts exhibiting behavior resembling $\exp(c(1-x)/\varepsilon)$ near $x = 1$ and $\exp(c(-1-x)/\varepsilon)$ near $x = -1$, respectively.

This insight prompts the extension of vanilla FNO to the construction of the Component Fourier Neural Operator (ComFNO) illustrated in Fig. 2. By incorporating exponential operations and the coordinate transformation $x \mapsto \xi = (x_0 - x)/\varepsilon$ to account for scaling in layers, ComFNO integrates prior knowledge of layer locations and types, thereby enhancing the existing operator learning framework. This approach involves two steps: (1) employing the "FNO" block to capture the smooth parts, and (2) employing layer blocks to learn specific layer-related information. For ComFNO with N layer blocks, the model can be succinctly represented by the equation:

$$\text{ComFNO} = \text{FNO}_0 + \sum_{i=1}^{N} \text{NN}_i * \exp(\text{FNO}_i). \tag{9}$$

Here, FNO_i (i=0,1,...,N) signifies the FNO model, while NN_i (i=1,2,...,N) represents shallow neural networks. FNO_0 corresponds to the "FNO" block in Fig. 2, while $\text{NN}_i * \exp(\text{FNO}_i)$ corresponds to the N layer blocks in ComFNO, where NN_i corresponds to the "Dense" block and FNO_i corresponds to the "extra_FNO" block.

For problems with spatial dimensions not less than 2, such as the two-dimensional PDE Eq. (7), the solution exhibits two exponential boundary layers along $x = 1$ and $y = 1$, displaying similar behavior to $\exp(c(1-x)/\varepsilon)$ near $x = 1$ and $\exp(c(1-y)/\varepsilon)$ near $y = 1$. However, their overlapping at the coordinate $(1,1)$ gives rise to a corner layer. In theory, one could incorporate a layer block in ComFNO to rectify inaccuracies near the corner $(1,1)$. Yet, each addition of a layer block results in an increase in network complexity, necessitating a trade-off. Interestingly, we observe that the corner layer part v^*_{as} in Eq. (8) can also be expressed as $\exp(c(1-x)/\varepsilon)$ or $\exp(c(1-y)/\varepsilon)$ with c as a function. Hence, the corner treatment is omitted, as the incorporation of two layer blocks is anticipated to rectify inaccuracies near the corner $(1,1)$. The subsequent section's experiment will demonstrate its effectiveness.

Experiments

In this section, we apply ComFNO to a wide range of SPDEs, including both ordinary and partial differential equations, as well as scenarios involving multiple data distributions and few-shot cases. Furthermore, we conduct a comparative analysis of the experimental results with those obtained using FNO.

Subsequently, we detail the default experimental setup, where, unless explicitly stated, experiment parameters are established as follows. The parameter ε in SPDEs remains set at 1×10^{-3}. Our aim is to learn the mapping $f \mapsto u$, where f represents the source term. The training dataset consists of 900×201 tuples (f, u), with 900 f samples independently drawn from Gaussian random fields and used as inputs. Resolution on $[0,1]$ or $[-1,1]$ is fixed at 201. To derive u, we use high-precision numerical methods. For steady-state problems, which are independent of time, the upwind scheme on the Shishkin mesh is employed. For time-dependent problems, the Crank–Nicolson scheme on the Shishkin mesh is used (Roos, Stynes, and Tobiska 2008). More detailed configurations can be found in the Appendix.

After training the models, we will assess their performance on 100 different f samples outside the training set, maintaining a resolution of 201. In the exposition of experimental findings, we will designate the high-precision numerical results as the ground truth, denoted by u_g, while the

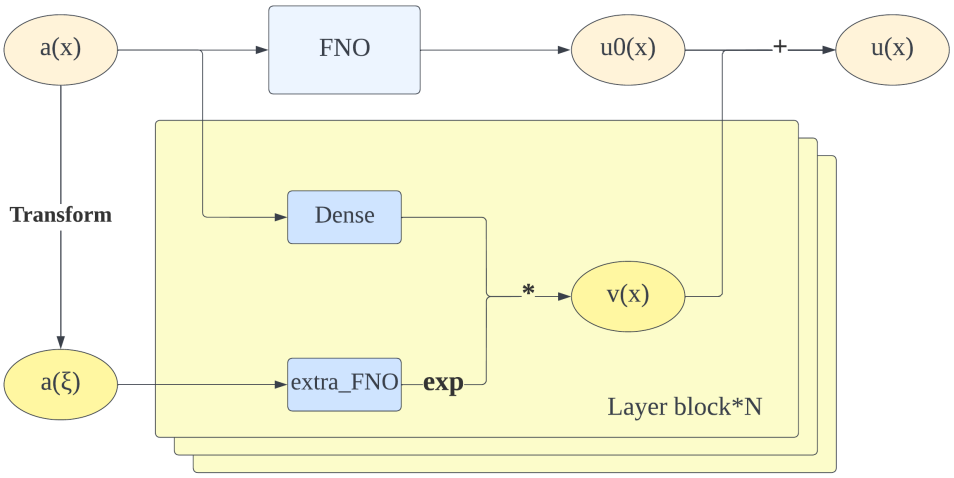


Figure 2: Architecture of ComFNO. $a(x)$ and $u(x)$ represent the input of model and solution of the problem, respectively. Both "FNO" and "extra_FNO" represent Fourier Neural Operators, with the latter being smaller. An "exp" function follows "extra_FNO," indicating an exponential operation applied to its output. "Dense" corresponds to a shallow neural network that learns the coefficients of the exponential function. The layer block's input comprises both $a(x)$ and $a(\xi)$, the latter involving a coordinate transformation to accommodate scaling in layers. For instance, when encountering a boundary or inner layer at $x = x_0$, the use of $\xi = (x_0 - x)/\varepsilon$ is advantageous.

model predictions will be represented as u_p. Throughout this manuscript, we will visualize the prediction residuals of both FNO and ComFNO, namely, $u_p - u_g$. Detailed ground truth and prediction results will be presented in the Appendix.

Ordinary Differential Equations

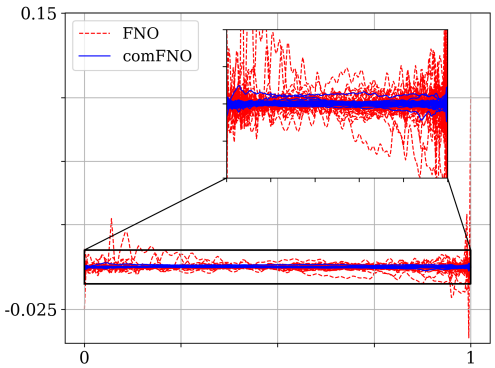


Figure 3: Performance of both FNO and ComFNO on Eq.(10) with $\varepsilon = 0.001$. Both trained models are evaluated on 100 f samples, and their resulting residual curves are depicted. (Subfigure): Zoomed-in view.

For ordinary differential equations, we examine cases both with and without turning points. We begin with the following problem:

$$\begin{cases} -\varepsilon u'' + (x+1)u' = f \quad x \in (0,1), \\ u(0) = u(1) = 0. \end{cases} \tag{10}$$

This problem's solutions feature an exponential boundary layer at $x = 1$, leading us to employ a single layer block and incorporate a coordinate transformation $\xi = (1 - x)/\varepsilon$ to incorporate $f(\xi)$ as one of the inputs to the layer block.

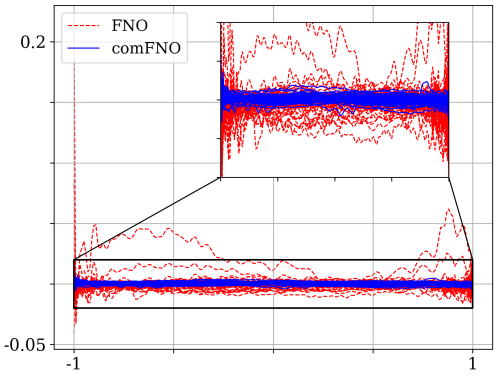


Figure 4: FNO and ComFNO performance on Eq.(11) with $\varepsilon = 0.001$. Both trained models are evaluated on 100 f samples, and their resulting residual curves are depicted. (Subfigure): Zoomed-in view.

Li et al. (2020) emphasizes FNO's resolution insensitivity, yielding improved results even at lower resolutions. Yet, as shown in Figure 4, FNO residuals are notably larger near $x = 1$ compared to other regions. This stems from limited data near the boundary layer and significant solution variations as ε approaches zero, potentially causing underfitting of the boundary layer.

Upon integrating a layer block, residuals decrease within

the boundary layer. Surprisingly, this addition boosts accuracy near the boundary layer and reduces overall errors. Remarkably, for this case, our layer block structure harmonizes with the solution's inherent differential equation configuration.

Next we consider the case with turning point $x = 0$:

$$\begin{cases} -\varepsilon u'' + x(x+2)u' + u = f, & x \in (-1, 1), \\ u(-1) = u(1) = 0. \end{cases} \tag{11}$$

Given the exponential boundary layers at both $x = -1$ and $x = 1$, two layer blocks in ComFNO are essential. These blocks receive inputs $(f(x), f((1-x)/\varepsilon))$ and $(f(x), f((-1-x)/\varepsilon))$, respectively. Fig. 3 displays the prediction residuals of ComFNO and FNO, revealing smaller residuals for ComFNO across all regions. This highlights ComFNO's proficiency in effectively addressing turning point problems.

Partial Differential Equations

Here we consider a parabolic differential equation in space-time domain:

$$\begin{cases} u_t - \varepsilon u_{xx} + u_x + xu = 0 & (x,t) \in (0,1) \times (0,1] \\ u(x,0) = f(x) & x \in [0,1] \\ u(0,t) = u(1,t) = 0, & t \in [0,1]. \end{cases} \tag{12}$$

Our aim is to learn the mapping $f \mapsto u(\cdot, 1)$. Incorporating a layer block with the input $(f(x), f((1-x)/\varepsilon))$ into ComFNO addresses the boundary layer near $x = 1$. Fig. 5 presents prediction residuals for both ComFNO and FNO.

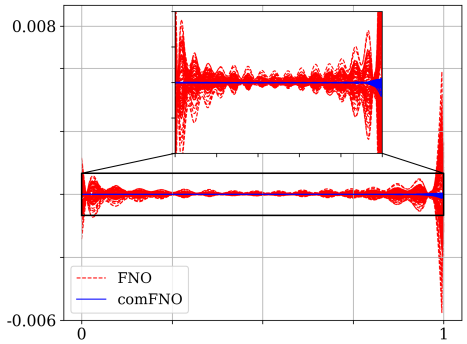
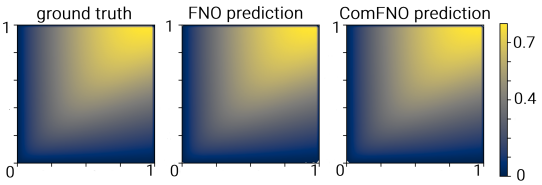


Figure 5: Performance of both FNO and ComFNO on Eq.(12) with $\varepsilon = 0.001$. Both trained models are evaluated on 100 f samples, and their resulting residual curves are depicted. (Subfigure): Zoomed-in view.

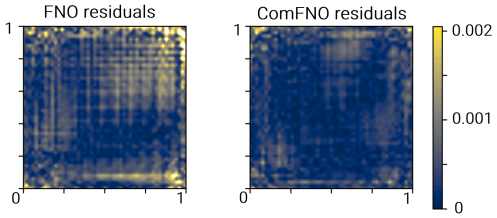
Finally we consider an elliptic differential equation:

$$\begin{cases} -\varepsilon \Delta u + u_x + u_y + u = f(x), & (x,y) \in (0,1)^2, \\ u(0,y) = u(1,y) = 0, & y \in [0,1] \\ u(x,0) = u(x,1) = 0, & x \in [0,1]. \end{cases} \tag{13}$$

The boundary layer for this equation is present at $x = 1$ and $y = 1$. To address this, we utilize two layer blocks with inputs $(f(x,y), f((1-x)/\varepsilon), y)$ and $(f(x,y), f(x,(1-y)/\varepsilon))$ in ComFNO. In Fig. 6(b), our approach's efficacy throughout the entire region, not just the boundary, is evident.



(a) predicts on both two methods



(b) absolute residuals on both two methods

Figure 6: Performance of FNO and ComFNO on Eq.(13) with $\varepsilon = 0.001$. The training set consists of $900 \times 51 \times 51$ tuples (f, u), including 900 independent f samples, each with a resolution of 51×51. Trained models are evaluated using a randomly selected f sample. Absolute residuals for both models are presented, calculated as $|u_p - u_g|$, where u_g is the ground truth and u_p represents the model predictions.

Multiple Distributions

Vanilla FNO faces challenges when handling equations with various distinct ε values concurrently, as differing ε can result in diverse data distributions. To tackle this, an effective strategy is to include the parameter ε as input or a prominent feature within the dataset, and the former method is used here. Our model follows a multi-input design, requiring the additional input ε for effective coordinate transformation.

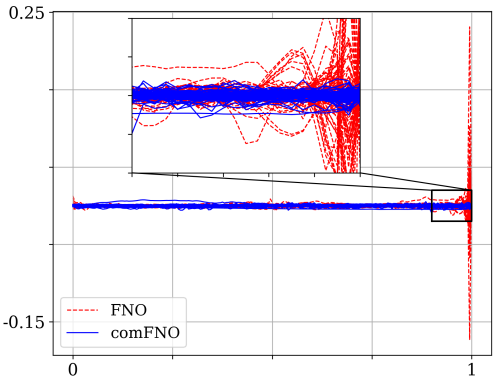


Figure 7: Performance of both FNO and ComFNO on Eq.(10). Both models are evaluated using 100 f samples with varying ε, and the corresponding residual curves are illustrated. (Subfigure): Zoomed-in view.

As an illustration, consider Eq. (10). The training dataset is composed of $100 \times 100 \times 201$ triplets (f, u). These triplets encompass 100 unique random functions f, each paired with

Experiment	FNO			ComFNO		
	mean	$\|\cdot\|_{L^\infty}$	var	mean	$\|\cdot\|_{L^\infty}$	var
1D(no turning point)	5.0e-04	3.8e-03	1.1e-06	1.0e-04	6.0e-04	1.3e-08
1D(turning point)	1.6e-03	8.5e-03	6.1e-06	6.0e-04	2.1e-03	1.9e-07
1D(initial-boundary)	9.0e-05	1.7e-03	4.8e-08	4.2e-06	4.0e-05	3.3e-11
2D	5.0e-4	1.67e-02	8.2e-07	2.0e-04	3.0e-03	1.5e-07
multiple ε	7.0e-04	1.7e-02	7.5e-06	6.0e-04	2.1e-03	2.5e-07
few-shot	1.2e-03	4.5e-03	1.5e-06	3.0e-4	1.3e-03	1.0e-07

Table 1: Mean, infinity norm, and variance of residuals on test set of all experiments.

100 distinct ε values ranging from 0.001 to 0.1 in increments of 0.001. The resolution is set at 201. Our main objective is to evaluate the ability of the two models to effectively fit this particular type of data.

We assess the model performance using 100 f samples that beyond the training set. Each f sample is associated with a distinct ε, ranging from 0.001 to 0.1. As shown in Fig. 7, the outcomes of FNO appear to lack certain data capturing layer structures. Conversely, ComFNO's results are more favorable, with minimal residuals observed in both boundary layers and other regions.

Few-Shot Situation

In this subsection, we primarily showcase the robustness of our method at varying sample sizes, highlighting the effectiveness of layer blocks in preserving sufficient physical information for successful model training, even with reduced data. We illustrate this using Eq. (10) as an example, where the training set is scaled down to 100×101 tuples (f, u). The prediction residuals of both models are displayed in Fig.8.

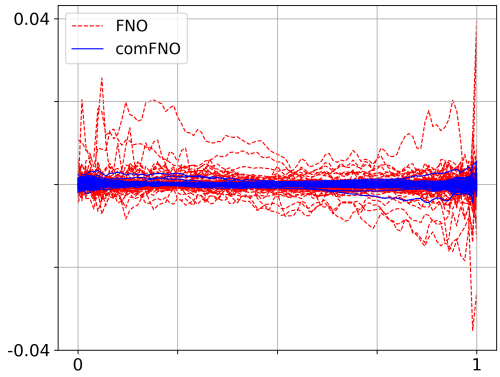


Figure 8: Residuals of FNO and ComFNO on Eq. (10) with $\varepsilon = 0.001$. Trained on 100 samples and tested on 100 different samples.

Metrics

In the previous experiments, we note that vanilla FNO's residuals display oscillations, mirroring the oscillatory nature of its predictions. Despite optimal training and minimal objective function values, these oscillations persist, compromising the model's reliability. For a more thorough comparative analysis of the two models, we introduce the metrics of mean, infinity norm, and variance for the residuals, as follows:

$$mean = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}|\hat{u}_{ij}-u_{ij}|,$$

$$\|\cdot\|_\infty = \frac{1}{N}\sum_{i=1}^{N}\max_{0\geq j\geq M}|\hat{u}_{ij}-u_{ij}|,$$

$$var = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(|\hat{u}_{ij}-u_{ij}| - \frac{1}{M}\sum_{j=1}^{M}|\hat{u}_{ij}-u_{ij}|\right)^2,$$

where N is the number of samples, M is the resolution. We computed mean, infinity norm, and variance of residuals for both FNO and ComFNO in all six experiments (Table 1), where smaller values denote superior model performance. Our innovative architecture demonstrates exceptional accuracy and effectively mitigates oscillations, as indicated by the indicators approaching zero.

Conclusion

This study introduces ComFNO, an innovative neural operator model tailored for addressing singularly perturbed differential equations (SPDEs). We initiate by presenting asymptotic analysis findings for various classes of SPDEs. Subsequently, we propose a unique layer block structure to enhance the training of conventional neural operators. Empirical results underscore the considerable improvement in prediction accuracy across boundary layer and other regions through the integration of layer blocks. Moreover, ComFNO exhibits superiority over vanilla FNO in specific instances, such as few-shot learning. Our model's versatility is demonstrated by testing it across multiple ε values, highlighting its adaptability as a multi-input variant of FNO that can accommodate diverse distributions. Furthermore, our experiments validate ComFNO's capability to mitigate oscillations.

In our experiments, we specifically focused on the problem of exponential boundary layers. However, it is essential to underscore that our approach remains applicable to a broader range of scenarios. Once we possess prior knowledge about the location and type of layers, we can adapt the local variables ξ accordingly and introduce additional operations within the layer blocks (e.g., incorporating exponential operations for exponential layers). This adaptability ensures the validity of our model for addressing inner layer problems or other types of boundary layer problems.

Acknowledgments

Ye Li is supported by the National Natural Science Foundation of China (No.62106103), Fundamental Research Funds for the Central Universities (No.ILA22023) and 173 Program Technical Field Fund (No.2021-JCJQ-JJ- 0018). Zhongyi Huang is supported by the National Natural Science Foundation of China (No.12025104)

References

Bar-Sinai, Y.; Hoyer, S.; Hickey, J.; and Brenner, M. P. 2019. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31): 15344–15349.

Becher, S.; and Roos, H.-G. 2015. Richardson extrapolation for a singularly perturbed turning point problem with exponential boundary layers. *Journal of Computational and Applied Mathematics*, 290: 334–351.

Beguinet, A.; Ehrlacher, V.; Flenghi, R.; Mula, O.; Somacal, A.; et al. 2022. Deep learning-based schemes for singularly perturbed convection-diffusion problems. *arXiv preprint arXiv:2205.04779*.

Chen, T.; and 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.

De Hoop, M.; Huang, D. Z.; Qian, E.; and Stuart, A. M. 2022. The cost-accuracy trade-off in operator learning with neural networks. *arXiv preprint arXiv:2203.13181*.

De Ryck, T.; and Mishra, S. 2022. Generic bounds on the approximation error for physics-informed (and) operator learning. *arXiv preprint arXiv:2205.11393*.

Deng, B.; Shin, Y.; Lu, L.; Zhang, Z.; and Karniadakis, G. E. 2022. Approximation rates of DeepONets for learning operators arising from advection–diffusion equations. *Neural Networks*, 153: 411–426.

E, W.; and Yu, B. 2018. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1): 1–12.

Eckhaus, W. 2011. *Matched asymptotic expansions and singular perturbations*. Elsevier.

Goswami, S.; Bora, A.; Yu, Y.; and Karniadakis, G. E. 2022. Physics-informed neural operators. *arXiv preprint arXiv:2207.05748*.

Greenfeld, D.; Galun, M.; Basri, R.; Yavneh, I.; and Kimmel, R. 2019. Learning to optimize multigrid PDE solvers. In *International Conference on Machine Learning*, 2415–2423. PMLR.

Haghighat, E.; Bekar, A. C.; Madenci, E.; and Juanes, R. 2021. A nonlocal physics-informed deep learning framework using the peridynamic differential operator. *Computer Methods in Applied Mechanics and Engineering*, 385: 114012.

Jin, P.; Meng, S.; and Lu, L. 2022. MIONet: Learning multiple-input operators via tensor product. *arXiv preprint arXiv:2202.06137*.

Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; and Yang, L. 2021. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440.

Kovachki, N.; Lanthaler, S.; and Mishra, S. 2021. On universal approximation and error bounds for Fourier neural operators. *Journal of Machine Learning Research*, 22: Art–No.

Krishnapriyan, A.; Gholami, A.; Zhe, S.; Kirby, R.; and Mahoney, M. W. 2021. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34: 26548–26560.

Lanthaler, S.; Mishra, S.; and Karniadakis, G. E. 2022. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1): tnac001.

Li, Z.; Kovachki, N. B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A.; et al. 2020. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*.

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

Liu, H.; Xing, B.; Wang, Z.; and Li, L. 2020. Legendre neural network method for several classes of singularly perturbed differential equations based on mapping and piecewise optimization technology. *Neural Processing Letters*, 51(3): 2891–2913.

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

Lu, L.; Meng, X.; Cai, S.; Mao, Z.; Goswami, S.; Zhang, Z.; and Karniadakis, G. E. 2022. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393: 114778.

Markidis, S. 2021. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Frontiers in big Data*, 92.

Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A.; and Edelman, A. 2020. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*.

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

Roos, H.-G. 2022. Robust Numerical Methods for Singularly Perturbed Differential Equations–Supplements. *arXiv preprint arXiv:2209.02994*.

Roos, H.-G.; Stynes, M.; and Tobiska, L. 2008. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*, volume 24. Springer Science & Business Media.

Rosofsky, S. G.; and Huerta, E. A. 2022. Applications of physics informed neural operators. *arXiv preprint arXiv:2203.12634*.

Simos, T.; and Famelis, I. T. 2022. A neural network training algorithm for singular perturbation boundary value problems. *Neural Computing and Applications*, 34(1): 607–615.

Tawfiq, L.; and Al-Abrahemee, K. 2014. Design neural network to solve singular pertubation problems. *Journal of Applied & Computational Mathematics*, 3: 1–5.

Wang, S.; Wang, H.; and Perdikaris, P. 2021. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40): eabi8605.

Wang, S.; Yu, X.; and Perdikaris, P. 2022. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449: 110768.

Yadav, S.; and Ganesan, S. 2021. SPDE-Net: Neural network based prediction of stabilization parameter for SUPG technique. In *Asian Conference on Machine Learning*, 268–283. PMLR.

Yang, Y.; Gao, A. F.; Castellanos, J. C.; Ross, Z. E.; Azizzadenesheli, K.; and Clayton, R. W. 2021. Seismic wave propagation and inversion with neural operators. *The Seismic Record*, 1(3): 126–134.

You, H.; Zhang, Q.; Ross, C. J.; Lee, C.-H.; and Yu, Y. 2022. Learning deep implicit Fourier neural operators (IFNOs) with applications to heterogeneous material modeling. *arXiv preprint arXiv:2203.08205*.