

Article

A Physics-Guided Bi-Fidelity Fourier-Featured Operator Learning Framework for Predicting Time Evolution of Drag and Lift Coefficients

Amirhossein Mollaali ¹, Izzet Sahin ¹, Iqar Raza ², Christian Moya ², Guillermo Paniagua ¹
and Guang Lin ^{1,2,*}

¹ School of Mechanical Engineering, Purdue University, West Lafayette, IN 47906, USA; amollaal@purdue.edu (A.M.); isahin@purdue.edu (I.S.); gpaniagua@purdue.edu (G.P.)

² Department of Mathematics, Purdue University, West Lafayette, IN 47906, USA; iraza@purdue.edu (I.R.); cmoyacal@purdue.edu (C.M.)

* Correspondence: guanglin@purdue.edu

Abstract: In the pursuit of accurate experimental and computational data while minimizing effort, there is a constant need for high-fidelity results. However, achieving such results often requires significant computational resources. To address this challenge, this paper proposes a deep operator learning-based framework that requires a limited high-fidelity dataset for training. We introduce a novel physics-guided, bi-fidelity, Fourier-featured deep operator network (DeepONet) framework that effectively combines low- and high-fidelity datasets, leveraging the strengths of each. In our methodology, we begin by designing a physics-guided Fourier-featured DeepONet, drawing inspiration from the intrinsic physical behavior of the target solution. Subsequently, we train this network to primarily learn the low-fidelity solution, utilizing an extensive dataset. This process ensures a comprehensive grasp of the foundational solution patterns. Following this foundational learning, the low-fidelity deep operator network's output is enhanced using a physics-guided Fourier-featured residual deep operator network. This network refines the initial low-fidelity output, achieving the high-fidelity solution by employing a small high-fidelity dataset for training. Notably, in our framework, we employ the Fourier feature network as the trunk network for the DeepONets, given its proficiency in capturing and learning the oscillatory nature of the target solution with high precision. We validate our approach using a well-known 2D benchmark cylinder problem, which aims to predict the time trajectories of lift and drag coefficients. The results highlight that the physics-guided Fourier-featured deep operator network, serving as a foundational building block of our framework, possesses superior predictive capability for the lift and drag coefficients compared to its data-driven counterparts. The bi-fidelity learning framework, built upon the physics-guided Fourier-featured deep operator, accurately forecasts the time trajectories of lift and drag coefficients. A thorough evaluation of the proposed bi-fidelity framework confirms that our approach closely matches the high-fidelity solution, with an error rate under 2%. This confirms the effectiveness and reliability of our framework, particularly given the limited high-fidelity dataset used during training.

Keywords: bi-fidelity learning; DeepONet; physics-guided; Fourier features network; cylinder benchmark; drag and lift coefficients



Citation: Mollaali, A.; Sahin, I.; Raza, I.; Moya, C.; Paniagua, G.; Lin, G. A Physics-Guided Bi-Fidelity Fourier-Featured Operator Learning Framework for Predicting Time Evolution of Drag and Lift Coefficients. *Fluids* **2023**, *8*, 323. <https://doi.org/10.3390/fluids8120323>

Academic Editors: Fang-Bao Tian and D. Andrew S. Rees

Received: 21 October 2023

Revised: 2 December 2023

Accepted: 13 December 2023

Published: 18 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Deep neural networks have emerged as a fast and reliable alternative for solving many scientific problems [1,2]. Recent advancements in hardware and software have made it possible to train extensive neural networks, e.g., physics-informed neural networks [2–4], deep operator networks [5–9], and neural operators [10–13], that accurately estimate and predict complex engineering and physical systems. However, constructing these models requires a large number of data samples for training. Due to the costs associated with experiments and

computations, most studies only include a limited number of high-fidelity data samples to describe a physical problem [14,15]. As a result, training neural networks with only high-fidelity data becomes prohibitively expensive and may lead to inaccurate results.

One possible solution for reducing the high cost of producing high-fidelity data is to incorporate data from models or experiments with varying accuracy into the neural network training procedure using multi-fidelity approaches. Instead of immediately resorting to expensive experimental setups or high-dimensional, high-resolution meshes to obtain accurate high-fidelity data, less accurate datasets can be acquired using prediction methods or coarse meshes to construct an initial low-fidelity surrogate. As expected, the accuracy of these initial results is impractical for subsequent steps and needs to be improved. To mitigate this issue and ensure accuracy, multi-fidelity techniques can be employed to bridge the gap between low-fidelity and high-fidelity spaces. This approach can offer cost-effective and precise results for various problems, such as parametric optimization, which relies on accurate data to provide optimal solutions.

In the context of multi-fidelity model optimization, various methods exist for obtaining a low-fidelity model, including simplified physics models, coarser meshes in high-fidelity models, or relaxed solver convergence criteria. However, it is crucial for surrogate models to represent the high-fidelity model accurately. To achieve this, corrections are applied to the low-fidelity models. For instance, techniques like space mapping, shape preserving, and adaptive response correction are utilized in aerodynamic models. Compared to function approximation surrogates, the main advantage of multi-fidelity model optimization is that it requires fewer high-fidelity data points to construct a physics-based model with the desired accuracy level. This reduction in data dependency improves the efficiency of the optimization algorithm, making the process faster and more cost-effective while maintaining acceptable accuracy levels [16]. For example, in [16], the low-fidelity surrogate, constructed from the datasets obtained from the transonic small-disturbance equation, was corrected using the shape-preserving response correction methodology. They were able to reduce computational and optimization costs by reducing the needed high-fidelity dataset by about 90%.

Multi-fidelity surrogates have shown promising results for several challenging problems and test conditions. For example, in [17], drag and lift coefficients were optimized for an airfoil under transonic conditions using a multi-fidelity model. The surrogate was constructed using coarse and fine meshes to collect low- and high-fidelity datasets. The authors suggested a multi-level method as a computationally effective solution by performing mapping algorithms such as multi-level optimization, space mapping, and shape-preserving response prediction techniques. Similarly, in [18], a multi-fidelity method was considered as an alternative cheap closure model for evaluating drag in shock–particle interactions, rather than using expensive fine-resolution computational solutions. The authors applied space mapping, radial basis functions, and modified Bayesian kriging as correcting techniques on low-fidelity surrogates to determine the best and most inexpensive closure model.

In [19], an accelerated optimization process with the multi-fidelity method was outlined for aerodynamic inverse design by comparing the manifold mapping-based design, which needs less than 20 high-fidelity and 1000 to 2000 low-fidelity evaluations, rather than direct aerodynamic inverse design subject to pattern search with 700 to 1200 high-fidelity model evaluations. Furthermore, in [20], by comparing single- and multi-fidelity methods to decrease the drag coefficient of a missile, a significant reduction in optimization time was observed. The authors used a faster semi-empirical missile tool to obtain low-fidelity data.

Recognizing the potential of multi-fidelity surrogates, efforts have been made to reduce the need for high-fidelity data and achieve cost savings. One approach is to enhance mapping methods. For example, in [21], various mapping approaches were explored to determine surrogate-based optimization schemes that can effectively handle constraints in variable-complexity design problems. They observed a 53% reduction in calling high-fidelity functions in the case of the wing design problem, resulting in reduced computational costs. In [22], the co-kriging method was used to create a genetic-based surrogate model for

a similar purpose. The genetic-based surrogate model was found to have better accuracy than the low-fidelity model-based surrogate models. Moreover, using the genetic-based model reduced computational costs due to fewer high-fidelity data requirements. To generalize the model, [23] focused on the residual while constructing the surrogate model for robust optimization. They created a surrogate model to eliminate errors due to the choice of turbulence model on the accuracy and cost of a numerical solution and applied the model to optimize diffuser geometry. In [24], the accuracy of the available multi-fidelity model was improved by combining gradient-enhanced kriging and a generalized hybrid bridge function while constructing the multi-fidelity model, and promising results of the robustness were reported.

To improve the accuracy of multi-fidelity surrogates and reduce the cost of high-fidelity data requirements, deep learning methods have been applied to create low-fidelity surrogates and map low- to high-fidelity spaces. The concept of an intelligent teacher–student interaction in machine learning, where student learning is accelerated with privileged information and corrected by transferring knowledge from teacher to student, is described in [25]. In [26], fidelity-weighted learning was introduced as a new student–teacher structure for deep neural networks. This approach was evaluated in natural language and information retrieval processing and outlined fast and reliable mapping for weakly and strongly labeled data. Inductive transfer and bi-fidelity-weighted learning methods were utilized in [27] for uncertainty propagation by constructing neural network surrogates from low- and high-fidelity datasets. Two approaches, partial adaption and shallow network, were used to map between low- and high-fidelity data. The bi-fidelity-weighted method showed promising performance on validation errors for three multidisciplinary examples.

Many works [28–34] have proposed scientific machine-learning-based solutions for the multi-fidelity problem. For instance, in [28], a novel approach called the deep operator network (DeepONet) integrated a multi-fidelity neural network model to reduce the desired high-fidelity data and attained an error one order of magnitude smaller. This approach was implemented to compute the Boltzmann transport equation (BTE) and proposed a fast solver for the inverse design of BTE problems. To reduce computational costs for complex physical problems involving parametric uncertainty and partial unknowns, a bi-fidelity modeling approach utilizing a deep operator network was introduced in [30]. This approach was applied to three problems: a nonlinear oscillator, heat transfer, and wind farm system. The evaluation demonstrates that the proposed method significantly improves the validation error by increasing its efficiency. In [31], the authors improved their proposed non-autonomous DeepONet-based framework [35] by incorporating a residual learning approach. This approach merges information from pre-existing mathematical models, enabling precise and efficient predictions in challenging power engineering problems. Finally, in [32], the authors proposed a deep operator learning framework for computing fine-scale solutions of multiscale partial differential equations (PDEs). Specifically, they trained multi-fidelity homogenization maps using mathematically motivated neural operators.

The objective of this paper is to significantly lower the computational expenses involved in simulating the time-dependent evolution of lift and drag coefficients without compromising on result resolution. To achieve this, we design a novel physics-guided, bi-fidelity Fourier-featured deep operator learning-based framework, which is constructed using coarse and fine datasets obtained through numerical simulation. To attain the aforementioned objective, we make the following contributions.

- We develop a physics-guided, bi-fidelity Fourier-featured deep operator learning-based framework (see Section 2.2). This framework takes an arbitrary undisturbed free-stream velocity as input and produces continuous, oscillatory time trajectories of the lift and drag coefficients for a cylinder within a channel. Our approach begins with the design and training of a physics-guided low-fidelity deep operator network using an extensive dataset that captures the foundational solution patterns. Subsequently, the low-fidelity deep operator network's predictions are enhanced through a physics-guided residual deep operator network. This elevation process transitions the low-

fidelity solution to a high-fidelity solution utilizing a small high-fidelity dataset. The use of the developed framework enables a comprehensive analysis of the target solution's time evolution for any given undisturbed free-stream velocity, eliminating the need for time-consuming numerical computations and simulations.

- The aforementioned deep operator learning-based framework is constructed using a novel physics-guided approach. This approach utilizes the oscillatory nature of the time trajectory of drag and lift coefficients to transform the problem into a functional inverse problem, thereby reducing the solution space for training the deep operator networks. In this paper, we compare the physics-guided approach with the traditional data-driven approach.
- Within this framework, we incorporate the Fourier-featured network as the trunk network of the DeepONets, thereby leading to the development of the Fourier-featured DeepONet. This incorporation harnesses the intrinsic capabilities of the Fourier-featured network, particularly its proficiency for capturing the fluctuations in lift and drag time trajectories. Consequently, the Fourier-featured DeepONet demonstrates superior performance compared to the vanilla DeepONet, which often struggles with understanding and precisely modeling these oscillatory patterns.

The rest of this paper is organized as follows. Section 2 outlines the numerical method used to generate data on the lift and drag coefficient trajectories, including the specifics of the simulation setup and parameters. Subsequently, we describe the physics-guided, bi-fidelity Fourier-featured deep operator learning methodology for developing a framework to learn target solutions. Section 3 presents numerical experiments that evaluate both the low-fidelity and the proposed bi-fidelity deep operator learning framework using their respective datasets. Section 4 discusses the results and their implications in detail. Finally, in the concluding Section 5, we encapsulate our key findings and conclusions.

2. Methodology and Description of the Tools

This paper proposes a novel approach called physics-guided bi-fidelity Fourier-featured deep operator learning to address the challenge of minimizing the cost associated with acquiring high-fidelity data. The approach is applied specifically to accurately estimate the fluctuations in the drag and lift coefficients over time for a cylinder in a channel operating under low-Reynolds-number conditions.

In order to obtain the necessary data, both low-fidelity and high-fidelity data are collected by simulating the flow around the cylinder using a commercial solver, Ansys Fluent. The data collection process involves varying the Reynolds number within an interval centered around 100. Coarse (low-fidelity) and fine (high-fidelity) meshes are used to capture the flow characteristics. Low-fidelity data are employed to establish an initial deep operator learning network that accepts velocity and time as inputs. To refine the predictions of the low-fidelity deep operator network, a secondary deep operator learning network utilizing high-fidelity data has been integrated. This integration leads to a more precise prediction of time trajectories for drag and lift coefficients. Please note that this study employs velocity as an input parameter. Recognizing the impact of inlet velocity on the pressure field, our future research will leverage detailed pressure data surrounding the cylinder as an input parameter to enhance further understanding.

In the following sections of this paper, we provide detailed descriptions of the numerical method used for data collection. This includes the simulation setup and parameters. Additionally, we delve into the bi-fidelity Fourier-featured deep operator learning framework, explaining its architectural components and training process. These advancements together contribute to a more efficient and cost-effective approach for estimating fluctuations in the drag and lift coefficients over time of the cylinder in the channel under low-Reynolds-number conditions.

2.1. Numerical Approach

We consider a cylinder with a diameter of $D = 0.1$ m placed inside a rectangular channel of length $L = 2.2$ m and height $H = 0.41$ m, as illustrated in Figure 1 (replicated from [36]). This benchmark simulation investigates two- and three-dimensional laminar flows around a cylinder and has been studied by several research groups using various numerical approaches. The simulation is detailed in terms of drag, lift, and Strouhal number. The x -axis (streamwise direction) and y -axis are aligned along the length and height of the channel, respectively. The center of the cylinder is positioned at $(0.2, 0.2)$, slightly off-center along the y -axis from the center line of the channel to induce vortex shedding. The left and right boundaries of the channel are taken as the inlet and outlet, respectively. The involved quantities are normalized with the cylinder's diameter, D . The flow geometry is illustrated in Figure 1. The top and bottom walls of the channel and cylinder solid surface follow the no-slip boundary condition, whereas the outlet boundary condition is set for the exit. A parabolic velocity condition, defined in Equation (1), is applied to the inlet boundary.

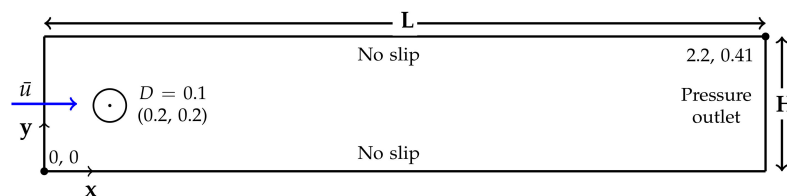


Figure 1. Two-dimensional Benchmark cylinder configuration adapted from [36].

$$u_{in}(y) = u_{max} \left[1 - \left(2 \frac{y - H/2}{H} \right)^2 \right]. \quad (1)$$

Here, $u_{in}(y)$ represents the velocity vector, and u_{max} denotes the maximum velocity at the centerline of the channel. The mean velocity of the flow can be calculated using Equation (2).

$$\bar{u} = \frac{1}{H} \int_0^H u_{in}(y) dy. \quad (2)$$

When u_{max} is set to 1.5 m/s, the mean velocity around the cylinder is $\bar{u} = 1$ m/s. This results in a Reynolds number of 100, based on Equation (3), where $\nu = 0.001$ m²/s is the kinematic viscosity of the fluid.

$$Re = \frac{\bar{u}D}{\nu}. \quad (3)$$

To create the low- and high-fidelity datasets, the Reynolds number was adjusted to values between 90 and 110 by varying the maximum velocity. For each inlet condition, the lift F_L and drag F_D were instantaneously calculated on the cylinder surface using Equation (4). Here, S denotes the cylinder surface, and n_x and n_y represent the x and y components of the normal vector on S , respectively. The tangential velocity on the cylinder surface, u_{tg} , is obtained using a tangent vector of $tg = (n_x, -n_y)$.

$$F_D = \int_S \left(\rho \nu \frac{\partial u_{tg}}{\partial n} n_y - P n_x \right) dS, \quad F_L = - \int_S \left(\rho \nu \frac{\partial u_{tg}}{\partial n} n_x + P n_y \right) dS \quad (4)$$

The lift C_L and drag C_D coefficients can be estimated using the corresponding forces on the cylinder surface, as shown in Equation (5).

$$C = \frac{2F}{\rho \bar{u}^2 D}. \quad (5)$$

The Fluent laminar solver is used for numerical simulations with transient settings. Unstructured grids are used for the simulations, with refinement near the cylinder wall.

A coarse mesh containing around 3000 cells is used to obtain low-fidelity data, while high-fidelity data are achieved with a refined mesh of around 87,700 cells. The Courant (CFL) number is kept below 0.9 for each mesh and inlet velocity condition. Additionally, the convergence criteria for the inner iterations are set to 10^{-8} .

Figure 2 compares the lift coefficient (C_L) and drag coefficient (C_D) of the reference [36] with the fine and coarse mesh results. It can be observed that, by using the fine mesh results, the maximum values of C_L and C_D overlap with the reference data, thus validating the results.

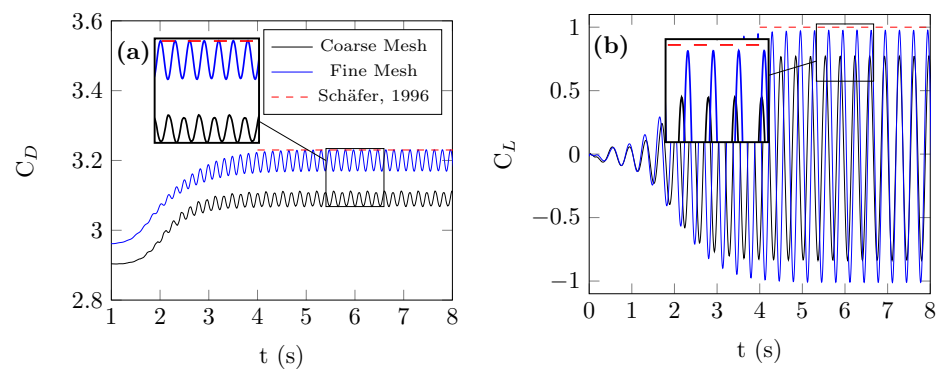


Figure 2. Comparison of low- (coarse mesh) and high-fidelity (fine mesh) C_D and C_L with the reference [36] at $Re = 100$: (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

2.2. Bi-Fidelity Fourier-Featured Deep Operator Learning

This section describes the proposed bi-fidelity Fourier-featured deep operator learning framework for approximating the operators that predict the drag and lift coefficients over time.

2.2.1. Deep Operator Learning

This paper proposes a bi-fidelity Fourier-featured deep operator learning framework to approximate the operator mapping $\mathcal{G}$ between the undisturbed free-stream velocity in the channel, denoted as $\bar{u} \in \mathcal{U}$, and the time-dependent trajectories of the drag and lift coefficients of the cylinder, denoted as C_D and C_L , respectively, for $t \in [t_0, t_f]$, i.e.,

$$\mathcal{G} : \bar{u} \mapsto v \quad (6)$$

where $v \in \{C_D, C_L\}$. In the above, the input domain for the undisturbed free-stream velocity $\bar{u}$ is given by $\mathcal{U} = [0.9, 1.1]$ m/s, and the time domain for the coefficients is $[t_0, t_f] = [0, 8]$ s.

To approximate the operator $\mathcal{G}$, we will design a bi-fidelity Fourier-featured deep operator learning framework, denoted as $\mathcal{G}_\theta^f$, where θ is the vector of trainable parameters. This framework consists of two deep operator networks, which we will introduce in the next sections: a low-fidelity deep operator network, denoted as $\mathcal{G}_{\theta^c}^c$, and a residual deep operator network, denoted as $\mathcal{G}_{\theta^e}^e$. The proposed framework then satisfies

$$\mathcal{G}_\theta^f = \mathcal{G}_{\theta^c}^c + \mathcal{G}_{\theta^e}^e \approx \mathcal{G} \quad (7)$$

2.2.2. Low-Fidelity Deep Operator Network

Using a coarse mesh in simulations of complex systems is an effective approach to reduce the computational demands of the simulation process. This methodology simplifies the system's complexity while generating abundant data that will significantly enhance the training process for deep operator networks. However, the adoption of a coarse mesh also introduces its own set of challenges. Notably, it omits fine-grained details in the system's dynamics, leading to approximation errors. These errors may initially be minor but can

compound over time, potentially resulting in significant discrepancies and impacting the accuracy and reliability of the deep operator network's predictions.

The proposed bi-fidelity framework capitalizes on the maximum advantages of the low-fidelity deep operator network while deploying a residual deep operator network to bridge the gap between the coarse and fine operators. This strategy enables the framework to benefit from the computational cost-effectiveness of the low-fidelity deep operator network without compromising the high-accuracy characteristic of the fine operator.

The proposed bi-fidelity learning approach first designs a low-fidelity deep operator network with trainable parameters θ^c , denoted as $\mathcal{G}_{\theta^c}^c : \bar{u} \mapsto v^c$. This deep operator network maps the undisturbed free-stream velocity $\bar{u} \in \mathcal{U}$ onto the low-fidelity solution v^c obtained using coarse mesh simulation. The target solution for the low-fidelity deep operator network, $v^c \in \{C_D, C_L\}$, corresponds to temporal trajectories of either the drag or lift coefficients within the specified time domain $[t_0, t_f]$. We will design this low-fidelity deep operator network to be either a physics-guided deep operator network (see Section 2.2.5) or a data-driven deep operator network with Fourier-featured layers (see Section 2.2.4).

Training the low-fidelity deep operator network. To train the coarse deep operator network, we minimize the following loss function:

$$\mathcal{L}(\theta^c; \mathcal{D}^c) = \frac{1}{N} \sum_{j=1}^N |v_j^c(\bar{u}_j)(t_j) - \mathcal{G}_{\theta^c}^c(\bar{u}_j)(t_j)|^2 \quad (8)$$

using the dataset of N triplets: $\mathcal{D}^c = \{u_i, t_i, v^c(\bar{u}_i)(t_i)\}_{i=1}^N$, where $v \in \{C_D, C_L\}$ for a given undisturbed free-stream velocity $\bar{u}_i \in \mathcal{U}$. In practice, for each velocity $\bar{u}_i$, we can obtain q samples of the target solution by evaluating it at q different times $\{t_i^j\}_{j=1}^q$.

Since generating the low-fidelity dataset is computationally less expensive, we assume that this dataset can be large enough. This large dataset is particularly advantageous for training our low-fidelity deep operator network as it provides a diverse range of examples to learn from, thereby enhancing the efficacy and generalization ability of the model.

Furthermore, we note that, in our framework, the trainable low-fidelity deep operator network could alternatively be replaced with either an efficient simulator, a physics-informed neural network (PINN) [2,4], or traditional machine learning approaches like spectral proper orthogonal decomposition (POD) [37]. An efficient simulator could enhance the computational process by providing faster approximations. In contrast, a PINN offers the potential to increase modeling efficiency by incorporating the simulator's equations directly into the loss function. Additionally, employing POD might provide valuable insights due to its ability to decompose complex datasets into a set of orthogonal modes, thereby simplifying the analysis and interpretation of fluid dynamics phenomena and potentially enhancing the model's predictive accuracy and interpretability.

2.2.3. Residual Deep Operator Network

To approximate the fine or high-fidelity operator, we propose a residual deep operator learning strategy that uses the output from the trained low-fidelity deep operator network $\mathcal{G}_{\theta^{c*}}^c$. Figure 3 provides a visual representation of the proposed framework. The framework involves defining the residual operator:

$$\mathcal{G}^e := \mathcal{G} - \mathcal{G}_{\theta^{c*}}^c \quad (9)$$

which estimates the difference between the true fine operator and the output of the trained low-fidelity deep operator network.

Bi-fidelity Learning Framework:

Training:

Coarse Dataset: $D^c = \{(\bar{u}_i^c, t_i, v^c(\bar{u}_i)(t_i))\}_{i=1}^{N^c}$

Fine Dataset: $D^f = \{(\bar{u}_i^f, t_i, v^f(\bar{u}_i)(t_i))\}_{i=1}^{N^f}$

1. Training low-fidelity DeepONet ($G_{\theta^c}^c$):

$(\bar{u}^c, t) \rightarrow \text{Low-fidelity DeepONet } (G_{\theta^c}^c) \rightarrow v^c(\bar{u}^c)(t)$

2. Training Residual DeepONet ($G_{\theta^e}^e$):

2.1: Predict low-fidelity solution for each $\bar{u}^f$ using trained Low-fidelity DeepONet ($G_{\theta^c}^c$):

$(\bar{u}^f, t) \rightarrow \text{Trained Low-fidelity DeepONet } (G_{\theta^c}^c) \rightarrow \bar{v}^c(\bar{u}^f)(t)$

2.2: Provide residual dataset:

$D^e = \{(\bar{u}_i^f, t_i, \epsilon(\bar{u}_i)(t_i))\}_{i=1}^{N^f}$, Where $\epsilon(\bar{u}^f)(t) = v^f(\bar{u}^f)(t) - \bar{v}^c(\bar{u}^f)(t)$

2.3: Train residual DeepONet on residual dataset:

$(\bar{u}^f, t) \rightarrow \text{Residual DeepONet } (G_{\theta^e}^e) \rightarrow \epsilon(\bar{u}^f)(t)$

Inference:

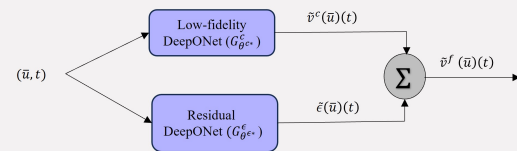


Figure 3. Training and inference phases of the developed bi-fidelity learning framework. In this figure, $\bar{u}$ represents the undisturbed free-stream velocity and t denotes time, both serving as inputs to the model. The variable $v \in \{C_D, C_L\}$ indicates the target solution. The superscripts c and f correspond to coarse and fine grades of the target solution, respectively, while $(\sim)$ symbolizes the approximate solution. The term $\epsilon = v^f - v^c$ is defined as residuals, which is the difference between the high-fidelity (v^f) and the low-fidelity (v^c) solutions. Notably, within this framework, both the low and residual DeepONets can be data-driven or physics-guided deep operators. Their specific types are not depicted in the figure to highlight this flexibility.

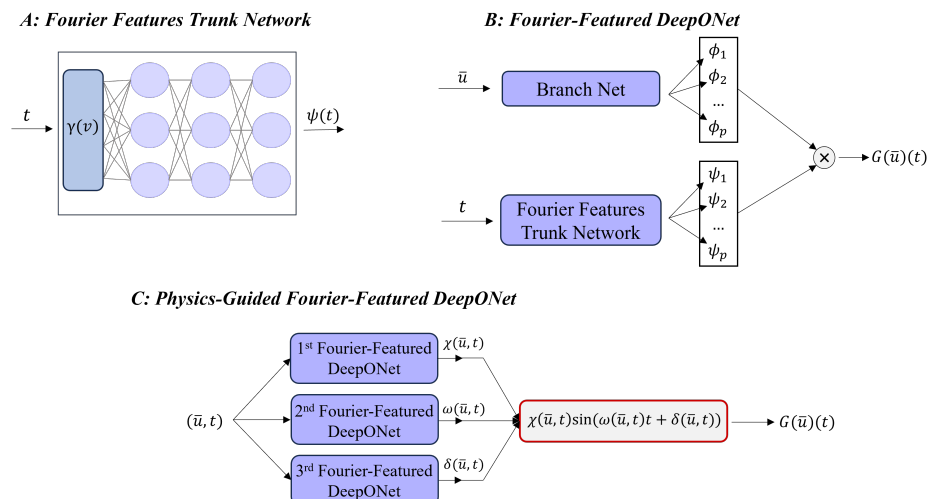


Figure 4. Hierarchy of the implemented models. (A) Fourier-featured trunk network. (B) Fourier-featured DeepONet structure. (C) Physics-guided Fourier-featured DeepONet model: integrating multiple DeepONets through physical equations to achieve the target solution. In this figure, $\bar{u}$ represents the undisturbed free-stream velocity and t denotes time, both serving as inputs to the model. Additionally, ϕ and ψ are the basis for the branch and trunk networks, respectively.

We approximate the residual operator mentioned above using a second deep operator network. Specifically, the proposed residual deep operator network, denoted by $G_{\theta^e}^e$, maps the undisturbed free-stream velocity $\bar{u} \in \mathcal{U}$ to the residual solution $\epsilon = v^f - v^c$, where

v^f and v^c are the fine solution and the predicted coarse solution, respectively. Note that this second network fine-tunes the predictions generated by the low-fidelity deep operator network, facilitating precise adjustments to the high-fidelity solution.

Training the residual deep operator network. To train the residual deep operator network, we minimize the following loss function:

$$\mathcal{L}(\theta; \mathcal{D}^\epsilon) = \frac{1}{N} \sum_{j=1}^N |\epsilon(\bar{u}_j)(t_j) - \mathcal{G}_{\theta^\epsilon}^\epsilon(\bar{u}_j)(t_j)|^2, \quad (10)$$

using the dataset of N triplets $\mathcal{D}^\epsilon = \{\bar{u}_i, t_i, \epsilon_j(\bar{u}_j)(t_j)\}_{i=1}^N$, where $\epsilon_j(\bar{u}_j)(t_j) = v^f(\bar{u}_j)(t_j) - \mathcal{G}_{\theta^{c*}}^c(\bar{u}_j)(t_j)$ denotes the residual error, i.e., the difference between the high-fidelity solution, obtained via fine-mesh simulation, and the corresponding prediction generated by the trained low-fidelity deep operator network.

Obtaining high-fidelity solutions for the dataset $\mathcal{D}^\epsilon$ can be expensive, resulting in a smaller dataset size compared to the coarse dataset $\mathcal{D}^c$. This expense is due to the substantial computational resources required for high-resolution CFD simulations. While we used a uniform random sampling method to collect data, it may not be ideal for costly high-fidelity data. Adopting a more strategic sampling approach, aimed at specific areas of complexity or interest, could potentially enhance the proposed framework's performance. We will explore this strategic sampling approach in future work.

In constructing the residual deep operator neural network, especially for achieving a balance between efficiency and accuracy through high-fidelity evaluations, determining the number of high-fidelity training points is vital but lacks a universal guideline due to its dependence on specific problem details and data nature. This paper addresses this by considering factors like problem complexity, low-fidelity data quality, and desired error thresholds, in conjunction with the neural network's size and structure. Through extensive hyper-parameter optimization, we have identified an optimal size for the high-fidelity dataset. This process is inherently iterative, necessitating adjustments based on the model's validation stage performance to finely tune the balance between computational efficiency and accuracy.

Inference. After training the low-fidelity and residual deep operator networks, we predict the time evolution of lift and drag coefficients for an arbitrary undisturbed free-stream velocity in a channel using a two-step process. First, we obtain a low-fidelity approximation of the solution using the trained low-fidelity deep operator network. This serves as our initial prediction/guess. Then, we use the residual deep operator network to predict the error between the true solution over time and our initial guess. Finally, we compute the approximation of the true solution by adding the outputs from both the trained low-fidelity and residual deep operator networks. Algorithm 1 summarizes the details of this two-step process for predicting the lift and drag coefficients for a given time partition $\mathcal{P} := \{t_0, \dots, t_f\}$.

Algorithm 1: A Bi-Fidelity Fourier-Featured Deep Operator Framework for Predicting the Lift and Drag Coefficients.

- 1 **Require:** The trained *low-fidelity* deep operator network $\mathcal{G}_{\theta^{c*}}^c$, the trained *residual* deep operator network $\mathcal{G}_{\theta^{\epsilon*}}^\epsilon$, the undisturbed free-stream velocity of the fluid in the channel $\bar{u} \in \mathcal{U}$, a given time partition $\mathcal{P}$.
 - 2 **Step 1:** Use the trained low-fidelity deep operator network $\mathcal{G}_{\theta^{c*}}^c$ to predict the low-fidelity solution on $\mathcal{P}$ for the given velocity $\bar{u}$, i.e., $\{v^c(\bar{u})(t) \equiv \mathcal{G}_{\theta^{c*}}^c(\bar{u})(t) : t \in \mathcal{P}\}$, where $v^c \in \{C_D, C_L\}$.
 - 3 **Step 2.** Use the trained residual deep operator network $\mathcal{G}_{\theta^{\epsilon*}}^\epsilon$ to predict the errors $\{\epsilon(\bar{u})(t) \equiv \mathcal{G}_{\theta^{\epsilon*}}^\epsilon(\bar{u})(t) : t \in \mathcal{P}\}$ between the high-fidelity solution and the predicted low-fidelity solution on $\mathcal{P}$ for the given velocity $\bar{u}$.
 - 4 **Return.** The predicted high-fidelity solution on $\mathcal{P}$, i.e., $\{v^f(\bar{u})(t) : t \in \mathcal{P}\}$, where $v^f(\bar{u})(t) = v^c(\bar{u})(t) + \epsilon(\bar{u})(t)$ and $v^f \in \{C_D, C_L\}$.
-

2.2.4. Deep Operator Learning

In this paper, we adopt the deep operator network (DeepONet) proposed in [5] as the foundational model for constructing our novel, physics-guided, bi-fidelity Fourier-featured deep operator learning-based framework. DeepONet is based on the universal approximation theorem of nonlinear operators [38]. It can approximate any nonlinear continuous operator, which are mappings between infinite dimensional spaces.

Figure 4B illustrates the DeepONet architecture, which is designed to approximate the target operator using a trainable linear representation. To enable this representation, it is crucial to construct and train two interconnected but distinct sub-neural networks: the branch network and the trunk network. The branch network is primarily responsible for handling the input, denoted as $\bar{u} \in \mathcal{U}$, and produces a vector of trainable basis coefficients, denoted as ϕ . The trunk network, on the other hand, decodes the output by processing the output location $t \in T$. Its output is another vector of trainable basis functions, denoted as ψ . The desired linear representation is achieved by taking the dot product of the outputs from the branch and trunk networks. This effectively combines these outputs in a meaningful way to approximate the target operator as follows:

$$v(\bar{u}, t) \approx \mathcal{G}_\theta(\bar{u})(t) = \sum_{k=1}^p \phi_k(\bar{u}) \cdot \psi_k(t) \quad (11)$$

2.2.5. Physics-Guided Deep Operator Learning

The oscillatory behavior of the time trajectory of lift and drag coefficients suggests that a physics-guided approach could effectively decipher these patterns. Physics-guided neural networks have a unique advantage in incorporating physical intuition into deep learning, resulting in accurate and robust predictions. This advantage is especially valuable when dealing with complex phenomena, such as predicting drag and lift coefficients across a wide range of input velocities. By incorporating physical intuition into deep learning, the precision, reliability, and generalizability of the predictions can be improved compared to fully data-driven models. Although data-driven models are powerful, they may fail to capture complex dependencies if they are not adequately represented in the dataset. Physics-guided models address this potential limitation by utilizing known relationships and behaviors.

In this paper, we adopt a physics-guided approach along with deep operator neural networks to transform the problem into an operator/functional inverse problem as illustrated in Figure 4C. In particular, the oscillatory nature of the lift and drag coefficients over time, as shown in Figure 2, suggests the suitability of sinusoidal family functions for modeling this behavior. Based on this observation, we propose integrating this physics-guided approach into our deep operator learning approach. More specifically, our proposed physics-guided deep operator network is defined as follows:

$$\mathcal{G}_P^v(\bar{u}, t) = \chi(\bar{u}, t) \sin(\omega(\bar{u}, t)t + \delta(\bar{u}, t)). \quad (12)$$

Here, $v \in \{C_D, C_L\}$ represents either the lift or drag coefficients, while χ , ω , and δ are the outputs of three distinct DeepONets. These networks take the undisturbed free-stream velocity and time as inputs. In this configuration, χ learns the amplitude variations over time of the target solution, ω understands the frequency of the solution's trajectory, and δ captures the phase angle. This division of tasks allows our model to capture both the scale and periodicity of the system dynamics. By embedding the physics of the problem into our model, we anticipate enhancing the model's robustness and reliability.

Training the Physics-Guided Deep Operator Network. To train the low-fidelity physics-guided deep operator network, we minimize the subsequent loss function:

$$\mathcal{L}(\theta^c; \mathcal{D}^c) = \frac{1}{N} \sum_{j=1}^N |v_j^c(\bar{u}_j)(t_j) - \mathcal{G}_{P, \theta^c}^c(\bar{u}_j)(t_j)|^2 \quad (13)$$

utilizing a dataset comprising N triplets, denoted as $\mathcal{D}^c = \{u_i, t_i, v^c(\bar{u}_i)(t_i)\}_{i=1}^N$, where $v \in \{C_D, C_L\}$ corresponds to a specific channel undisturbed free-stream velocity $\bar{u}_i$ from the set $\mathcal{U}$. Furthermore, to train the residual physics-guided deep operator network, we optimize the following loss function:

$$\mathcal{L}(\theta; \mathcal{D}^c) = \frac{1}{N} \sum_{j=1}^N |\epsilon(\bar{u}_j)(t_j) - \mathcal{G}_{P, \theta^c}^c(\bar{u}_j)(t_j)|^2, \quad (14)$$

using the dataset of N triplets $\mathcal{D}^c = \{\bar{u}_i, t_i, \epsilon_j(\bar{u}_j)(t_j)\}_{i=1}^N$, where the term $\epsilon_j(\bar{u}_j)(t_j) = v^f(\bar{u}_j)(t_j) - \mathcal{G}_{P, \theta^{c*}}^c(\bar{u}_j)(t_j)$ represents the residual error.

Inference. For a comprehensive understanding, Algorithm 2 provides a thorough depiction of the inference phase for the bi-fidelity framework, built upon the physics-guided deep operator components.

Algorithm 2: A Physics-Guided Bi-Fidelity Fourier-Featured Deep Operator Framework for Predicting the Lift and Drag Coefficients.

- 1 **Require:** The trained *physics-guided low-fidelity* deep operator network $\mathcal{G}_{P, \theta^{c*}}^c$, the trained *physics-guided residual* deep operator network $\mathcal{G}_{P, \theta^{c*}}^{\epsilon}$, the undisturbed free-stream velocity of the fluid in the channel $\bar{u} \in \mathcal{U}$, a given time partition $\mathcal{P}$.
 - 2 **Step 1:** Use the trained physics-guided low-fidelity deep operator network $\mathcal{G}_{P, \theta^{c*}}^c$ to predict the low-fidelity solution on $\mathcal{P}$ for the given velocity $\bar{u}$, i.e., $\{v^c(\bar{u})(t) \equiv \mathcal{G}_{P, \theta^{c*}}^c(\bar{u})(t) : t \in \mathcal{P}\}$, where $v^c \in \{C_D, C_L\}$.
 - 3 **Step 2.** Use the trained physics-guided residual deep operator network $\mathcal{G}_{P, \theta^{c*}}^{\epsilon}$ to predict the errors $\{\epsilon(\bar{u})(t) \equiv \mathcal{G}_{P, \theta^{c*}}^{\epsilon}(\bar{u})(t) : t \in \mathcal{P}\}$ between the high-fidelity solution and the predicted low-fidelity solution on $\mathcal{P}$ for the given velocity $\bar{u}$.
 - 4 **Return.** The predicted high-fidelity solution on $\mathcal{P}$, i.e., $\{v^f(\bar{u})(t) : t \in \mathcal{P}\}$, where $v^f(\bar{u})(t) = v^c(\bar{u})(t) + \epsilon(\bar{u})(t)$ and $v^f \in \{C_D, C_L\}$.
-

2.2.6. Fourier Features Network

To design each DeepONet, we employ a feed-forward neural network for the branch network. However, traditional feed-forward networks have limitations in capturing high-frequency oscillatory patterns, making it difficult to encode the target output solutions, specifically the oscillatory time trajectory of the drag and lift coefficients. To overcome this limitation, we develop the Fourier-featured DeepONet by adopting the Fourier features network as the trunk network within the DeepONets' architecture. This network is responsible for encoding the output and can effectively encapsulate the target solution. It is selected for its exceptional ability to capture periodic patterns and oscillatory behaviors, transcending the capabilities of conventional feed-forward layers.

The Fourier features network, as shown in Figure 4A, relies on a random Fourier mapping expressed as $\gamma(v) = [\sin(Bv), \cos(Bv)]^T$ [39]. The matrix B in $R^{m \times d}$ contains values sampled from a Gaussian distribution $N(0, \sigma^2)$. By integrating a random Fourier mapping with a traditional neural network, the Fourier features network effectively boosts learning capabilities by simplifying high-dimensional data. This method is distinctive in its ability to mitigate spectral bias, a common issue that hinders the learning of high-frequency data components. As a result, it significantly enhances performance across various tasks, making it an ideal choice for the trunk network in the DeepONet model used for predicting oscillatory time trajectories of lift and drag coefficients.

3. Numerical Model Results

This section presents the training and testing procedures used in the proposed physics-guided bi-fidelity Fourier-featured deep operator learning framework to predict the drag and lift coefficients of a cylinder.

3.1. Training and Testing Datasets

To train and evaluate our physics-guided bi-fidelity Fourier-featured deep operator learning framework, two separate datasets are needed: one for low-fidelity data and another for high-fidelity data. The low-fidelity dataset is obtained by simulating with a coarse mesh, while the high-fidelity dataset is generated using a fine mesh. The fine mesh of the high-fidelity dataset produces more accurate results but comes with a significantly higher computational cost. Within each dataset, two-dimensional numerical simulations were conducted for each $u_j \in \mathcal{U}$, where $j = 1, \dots, N$, specifically for the undisturbed free-stream velocity within the channel. To obtain the training targets, we monitored and recorded the lift and drag coefficients of the cylinder over time. The values of these coefficients were collected within the specified time domain. Another key difference between these datasets, apart from the mesh resolution, is their size. The low-fidelity dataset is much larger than its high-fidelity counterpart, primarily due to cost-effectiveness and affordability.

In this paper, we utilized 150 sets of simulation data to train the low-fidelity DeepONet model and 50 sets to train the residual DeepONet. For testing purposes, we allocated 10% of the data, while the remaining 90% were used for training. To ensure independence, we employed a simple random selection method to transform the input functions into separate samples. This approach fosters the development of a more generalized model, enhancing its applicability under diverse operating conditions.

3.2. Neural Networks and Training Protocols

To ensure the model's effectiveness and establish an efficient training process, we conducted routine hyper-parameter optimization. This optimization aimed to identify the optimal architecture and suitable settings for the training process. The results of this optimization are presented in Table 1. For the training of all DeepONets discussed in this paper, physics-guided and data-driven, we employed the settings detailed along with the Adam optimizer, coupled with a “reduce-on-plateau” learning rate scheduler.

Table 1. Hyper-parameter optimization results.

Hyper-Parameter	Optimum Value
Initial learning rate (η)	1×10^{-4}
Number of layers in each net	3
Number of neurons in each layer	100
Activation function	ReLU
Loss function	MSE

3.3. Low-Fidelity Deep Operator Learning Results

For low-fidelity learning, we designed both physics-guided and data-driven low-fidelity deep operator networks and assessed their performance. After training each operator on the low-fidelity training dataset, we evaluated their effectiveness using a low-fidelity test dataset. This test dataset constitutes 10 percent of the total low-fidelity data, which were excluded from the training phase. It includes trajectories spanning $q = 300$ time steps within the output domain $[t_0, t_f]$.

To visualize the performance of the proposed low-fidelity models, we randomly selected an undisturbed free-stream velocity $u_j \in \mathcal{U}$ from the low-fidelity test dataset and predicted the drag and lift coefficients using both physics-guided and data-driven deep operator networks. Figures 5 and 6 depict these coefficients for the chosen velocity. The displayed results underscore the exceptional predictive prowess of the low-fidelity DeepONet models; their predictions align closely with the true values derived from CFD simulations using a coarse mesh.

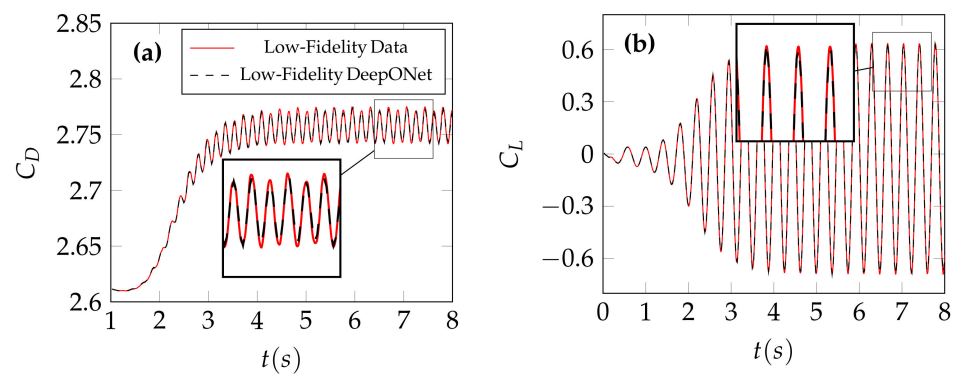


Figure 5. Comparison between the low-fidelity solution (data) time trajectory and the corresponding low-fidelity physics-guided Fourier-featured DeepONet prediction for a random undisturbed free-stream velocity ($\bar{u} = 0.94$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

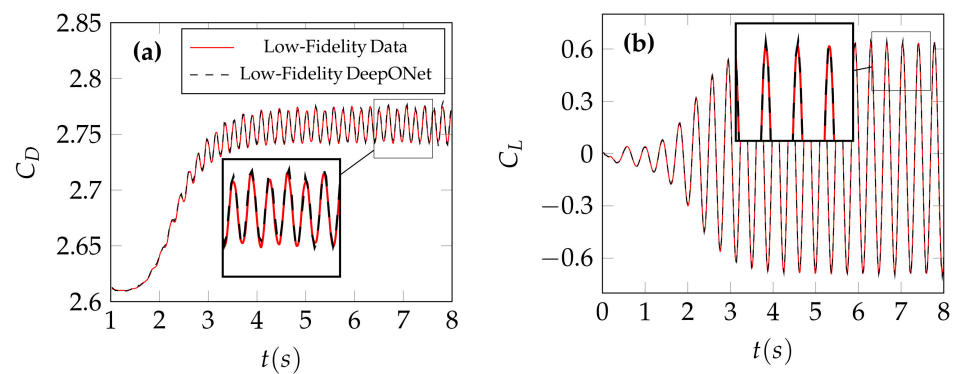


Figure 6. Comparison between the low-fidelity solution (data) time trajectory and the corresponding low-fidelity data-driven Fourier-featured DeepONet prediction for a random undisturbed free-stream velocity ($\bar{u} = 0.94$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

Comparative Performance Analysis of Low-Fidelity Deep Operator Networks

In order to evaluate and compare the generalization capabilities of the proposed physic-guided and data-driven models, we computed the mean and standard deviation of the L_1 - and L_2 -relative errors for each model using 15 test trajectories. These were calculated between the actual and forecasted values of the drag and lift coefficients' time trajectories using the low-fidelity testing dataset. As shown in Tables 2 and 3, the physics-guided low-fidelity DeepONet consistently outperformed its data-driven counterpart in terms of accuracy and reduced errors. A key contributing factor to this enhanced performance is that the physics-guided approach inherently narrows down the solution space, ensuring more precise predictions by leveraging established physical principles and constraints. However, both the physics-guided and data-driven low-fidelity DeepONets achieved errors below 2% for both coefficients. This demonstrates the effectiveness of the low-fidelity deep operator network in providing accurate predictions within the context of the problem domain, showcasing its potential to serve as the basis for the bi-fidelity framework.

Table 2. Mean and standard deviation of L_1 - and L_2 -relative errors between low-fidelity solution (data) and low-fidelity Fourier-featured deep operator networks' predictions for the drag coefficient, C_D .

Low-Fidelity Model	mean L_1	st.dev. L_1	mean L_2	st.dev. L_2
Physics-guided	0.26%	0.13%	0.34%	0.16%
Data-driven	0.47%	0.34%	0.59%	0.39%

Table 3. Mean and standard deviation of L_1 - and L_2 -relative errors between low-fidelity solution (data) and low-fidelity Fourier-featured deep operator networks' predictions for lift coefficient, C_L .

Low-Fidelity Model	mean L_1	st.dev. L_1	mean L_2	st.dev. L_2
Physics-guided	0.71%	0.81%	0.90%	0.93%
Data-driven	0.88%	1.04%	1.06%	1.14%

3.4. Bi-Fidelity Fourier-Featured Deep Operator Learning Results

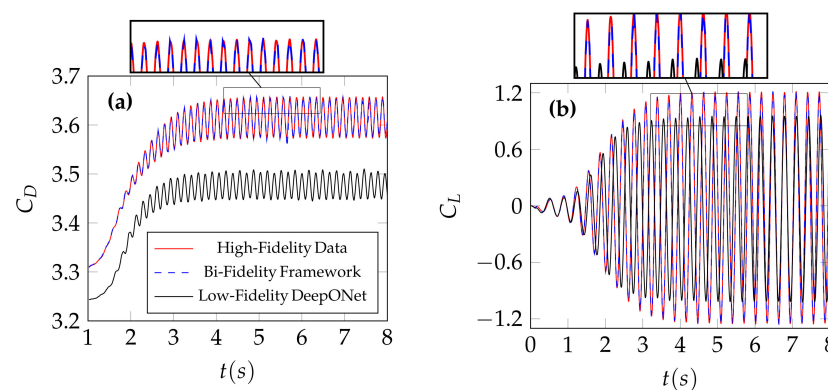
In our proposed bi-fidelity learning framework, both the low-fidelity and residual models can be either a physics-guided or data-driven Fourier-featured DeepONet. To rigorously assess the effectiveness of this bi-fidelity structure and the performance of the introduced physics-guided Fourier-featured DeepONet, we constructed three distinct bi-fidelity configurations:

- First Configuration: Both the low-fidelity and the residual models are physics-guided Fourier-featured DeepONets.
- Second Configuration: The low-fidelity model is a data-driven Fourier-featured DeepONet, while the residual model is a physics-guided Fourier-featured DeepONet.
- Third Configuration: Both the low-fidelity and residual models are data-driven Fourier-featured DeepONets.

Through these diverse configurations, we aim to provide a comprehensive analysis of the data-driven and physics-guided approaches within our bi-fidelity learning framework.

This section focuses on the assessment of these frameworks through their predictions on the high-fidelity test dataset. For this purpose, after training both the low-fidelity and residual DeepONets, we employ the methodology outlined in Algorithms 1 and 2 to approximate the high-fidelity solution for each framework. These solutions correspond with the respective bi-fidelity frameworks' predictions on the high-fidelity test dataset.

To demonstrate the performance of our proposed bi-fidelity learning framework, we randomly selected an undisturbed free-stream velocity $u_j \in \mathcal{U}$ from the high-fidelity test dataset. Utilizing this velocity, we predicted the time trajectory of both drag and lift coefficients by employing each of the three previously outlined bi-fidelity learning framework configurations. Figures 7–9 present the drag and lift coefficients' time trajectory for the chosen velocity. Correspondingly, these figures represent predictions obtained from the first, second, and third configurations, respectively. As observed in these figures, the fine solution for both drag and lift coefficients aligns closely with that of the bi-fidelity Fourier-featured deep operator learning frameworks' predictions, indicative of a strong match. Such observations imply not only the superior capability of the residual deep operator network in discerning the residuals between coarse predictions and fine solutions but also underscore the outstanding performance of our proposed bi-fidelity learning framework.

**Figure 7.** Comparison between the high-fidelity solution (data) time trajectory and the prediction from the first configuration of the bi-fidelity learning framework: (**low-fidelity model**: physics-guided Fourier-featured DeepONet; **high-fidelity model**: physics-guided Fourier-featured DeepONet). Displayed for a random undisturbed free-stream velocity ($\bar{u} = 1.06$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

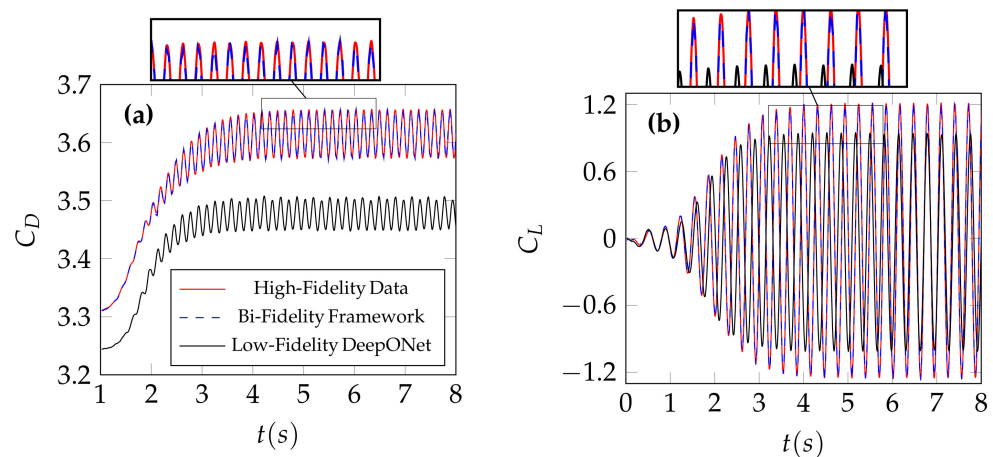


Figure 8. Comparison between the high-fidelity solution (data) time trajectory and the prediction from the second configuration of the bi-fidelity learning framework: (**low-fidelity model**: data-driven Fourier-featured DeepONet; **high-fidelity model**: physics-guided Fourier-featured DeepONet). Displayed for a random undisturbed free-stream velocity ($\bar{u} = 1.06$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

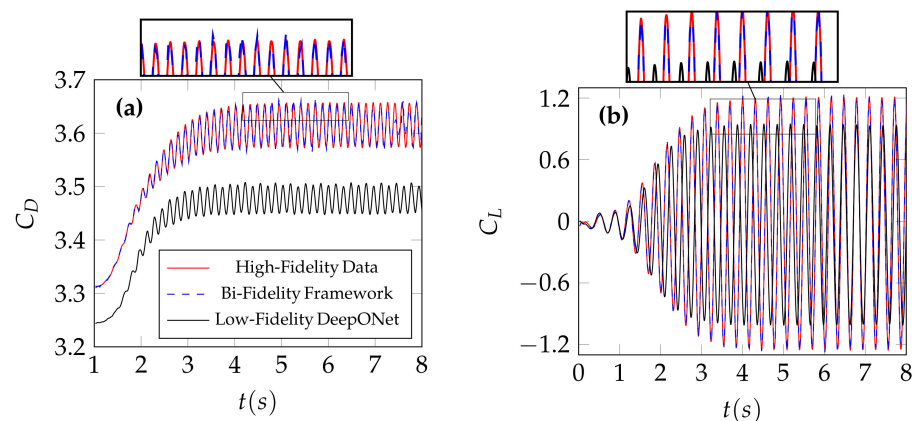


Figure 9. Comparison between the high-fidelity solution (data) time trajectory and the prediction from the third configuration of the bi-fidelity learning framework: (**low-fidelity model**: data-driven Fourier-featured DeepONet; **high-fidelity model**: data-driven Fourier-featured DeepONet). Displayed for a random undisturbed free-stream velocity ($\bar{u} = 1.06$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

3.4.1. Comparative Performance Analysis of Bi-Fidelity Learning Frameworks

To comprehensively evaluate the generalization performance of the proposed bi-fidelity learning frameworks, Tables 4 and 5 show the L_1 - and L_2 -relative errors corresponding to each framework's predictions of the fine solution. These statistics underscore that all three bi-fidelity configurations achieve errors below 3% when predicting the fine solution for the target coefficients. Interestingly, the results indicate a superior predictive capability from the physics-guided approach, attributable to its leveraging of the system's physical equations. Among the configurations, the first, which employs a physics-guided Fourier-featured DeepONet for both low-fidelity and residual models, yields the lowest error. Furthermore, the second configuration, utilizing the physics-guided Fourier-featured DeepONet for the residual model and data-driven DeepONet for the low-fidelity model, exhibits lower errors than the third configuration that relies on the data-driven Fourier-featured DeepONet for both models. This clearly showcases the efficacy of the proposed physics-guided deep operator for both low-fidelity and residual models in the bi-fidelity framework.

Table 4. Mean and standard deviation of L_1 - and L_2 -relative errors between high-fidelity solution (data) and the bi-fidelity Fourier-featured deep operators learning frameworks' predictions for the drag coefficient, C_D .

Low-Fidelity Model	Residual Model	mean L_1	st.dev. L_1	mean L_2	st.dev. L_2
Physics-guided	Physics-guided	0.83%	0.49%	1.19%	0.65%
Data-driven	Physics-guided	0.84%	0.46%	1.22%	0.63%
Data-driven	Data-driven	1.24%	0.65%	1.73%	0.81%

Table 5. Mean and standard deviation of L_1 and L_2 -relative errors between high-fidelity solution (data) and the bi-fidelity Fourier-featured deep operators learning frameworks' predictions for lift coefficient, C_L .

Low-Fidelity Model	Residual Model	mean L_1	st.dev. L_1	mean L_2	st.dev. L_2
Physics-guided	Physics-guided	1.60%	0.88%	2.02%	1.10%
Data-driven	Physics-guided	1.70%	0.75%	2.10%	0.92%
Data-driven	Data-driven	2.03%	0.98%	2.55%	1.35%

3.4.2. Comparison of Physics-Guided Bi-Fidelity Deep Operator Learning Framework: With vs. without Fourier Feature Trunk Network

As detailed in Section 2.2.6, we employed Fourier feature networks as the trunk net for both the data-driven and physics-guided DeepONets in our low-fidelity and residual models. Referring to Figure 10, it becomes evident that the bi-fidelity deep operator learning framework constructed using the vanilla DeepONet struggles to capture the oscillatory characteristics of the lift and drag time trajectories. In contrast, the proposed bi-fidelity Fourier-featured deep operator learning framework adeptly captures the fluctuations in these trajectories. This underlines the effectiveness and appropriateness of integrating the Fourier-featured trunk net with the DeepONet, especially for lift and drag time evolution prediction.

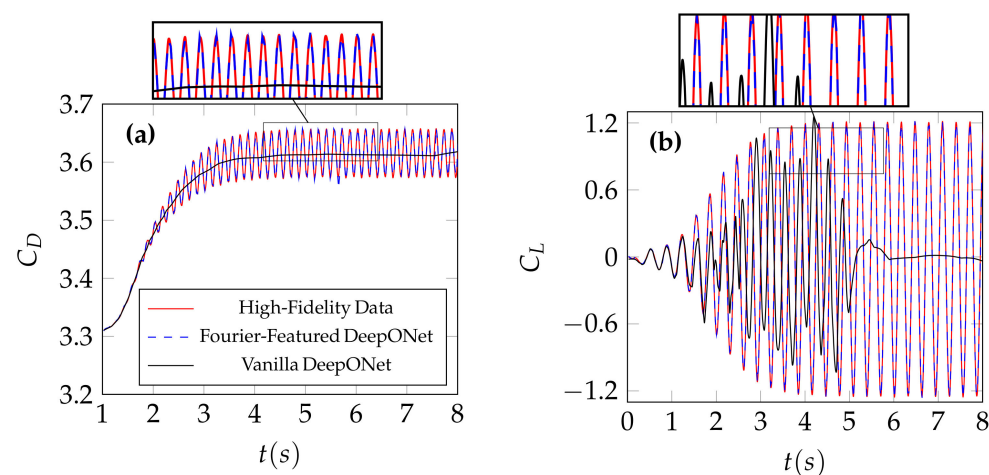


Figure 10. Comparison of physics-guided bi-fidelity deep operator learning frameworks utilizing: vanilla DeepONet vs. Fourier-featured DeepONet. Displayed for a random undisturbed free-stream velocity ($\bar{u} = 1.06$ m/s): (a) drag coefficient, C_D , and (b) lift coefficient, C_L .

4. Discussion

The developed physics-guided bi-fidelity Fourier-featured deep operator learning framework has demonstrated outstanding performance, achieving error rates below 2% for both coarse and fine operator approximation. This indicates a highly accurate prediction of time trajectories for the drag and lift coefficients. This achievement is particularly

noteworthy considering that only 45 high-fidelity input trajectories ($N = 45$) were used in the training process. It highlights the high efficiency of the proposed framework, which effectively utilizes available information, such as inferred physics, especially in scenarios where obtaining high-fidelity data is expensive. These results clearly emphasize the potential of our framework in tackling complex computational challenges while optimizing resource utilization.

Based on our comprehensive comparative analysis, it is evident that the proposed physics-guided Fourier-featured DeepONet outperforms its data-driven counterpart, primarily due to its enhanced predictive power and efficiency in narrowing the solution space. This makes it an optimal tool within the bi-fidelity learning framework, particularly for the low-fidelity and residual models when predicting lift and drag coefficients. However, it is crucial to acknowledge several key distinctions between these two methodologies.

As is observed in Tables 4 and 5, incorporating the physics-guided DeepONet in the residual model has a more noticeable impact on enhancing the bi-fidelity learning framework's performance compared to its use as a low-fidelity model. The primary reason for this is the differing complexities in the relationship between the input and the solution in low-fidelity and residual modeling. In the context of low-fidelity modeling, the relationship is relatively straightforward, with changes primarily in amplitude and without significant alterations in phase shift and frequency. Conversely, the residual model, which is tasked with predicting the residual solution, presents a more complex scenario. This model exhibits behavior that varies across all parameters: amplitude, frequency, and phase shift. Thus, employing a physics-guided DeepONet, which concentrates on narrowing the solution space in these dimensions, enhances the framework's ability to accurately map the input to the output.

Physics-guided models are notable for their ability to incorporate underlying physics-based insights into their framework. This incorporation has the potential to generate more robust and accurate predictions. This feature is particularly advantageous when dealing with complex phenomena, such as predicting drag and lift coefficients across a wide range of input values. By integrating physics-based insights into the model, we can potentially improve the precision and reliability of predictions, even when dealing with a diverse and complex parameter space.

On the other hand, while data-driven models are undoubtedly powerful, they may overlook intricate dependencies if these subtleties are not adequately captured within the dataset. Physics-guided models address this potential limitation by leveraging known relationships and behaviors, thereby reducing the risk of overlooking such details.

Future Work

While recognizing the unique strengths and limitations of both data-driven and physics-guided approaches, our research establishes a foundation for future advancements in the bi-fidelity framework of lift and drag coefficients. This paves the way for exploring other bi-fidelity learning methods, such as input augmentation [28]. This approach has the potential to improve the accuracy of predicting fine solutions, offering a promising avenue for more precise and reliable predictions in bi-fidelity learning.

Looking toward future research, a promising direction is to explore the integration of physics-informed neural networks (PINNs) and traditional low-order methods like proper orthogonal decomposition (POD) within our bi-fidelity learning framework. Applying these methods for both low-fidelity and residual learning, we plan to conduct a comparative analysis against our DeepONet-based bi-fidelity framework. This comparison is expected to yield a deeper understanding of the strengths and limitations inherent in both DeepONet-based configurations and traditional model-based approaches.

Additionally, an exciting future direction involves the implementation of the recently introduced POD-DeepONet model that employs the POD modes of the training data as the trunk net [40]. Employing this model in both the low-fidelity and residual aspects of our

framework could enhance the predictive accuracy and efficiency, leading to a more robust and versatile framework for accurately predicting complex fluid dynamics phenomena.

DeepONets are proficient in mapping inputs to outputs, yet they encounter limitations due to their fixed output domain. This restriction affects their performance when dealing with data outside their trained output domain range. In response to this challenge, recent advancements have proposed the use of a non-autonomous DeepONet, designed to operate beyond this fixed domain and improve prediction capabilities [35]. Accordingly, another significant direction for future research involves replacing the standard DeepONet with the non-autonomous variant in both the low-fidelity and residual models of our framework. Implementing this change is expected to enhance the framework's capacity to navigate various output domains, potentially increasing the predictive accuracy for lift and drag coefficients. The adoption of a non-autonomous DeepONet could lead to the development of a more adaptable and robust bi-fidelity learning framework.

While neural networks, such as DeepONets, are adept at interpolation tasks, they encounter difficulties in extrapolation scenarios, as highlighted in [41]. Our current study does not explore situations where inputs extend beyond the established range of the training set. Nevertheless, [41] proposes several methods to address these extrapolation challenges. One approach is extrapolation via fine-tuning with sparse new observations, which involves fine-tuning the network using sparsely acquired new data. Another technique is extrapolation via multi-fidelity learning with sparse new observations, incorporating multi-fidelity learning strategies to integrate data of varying accuracy and detail. These methods hold the potential to significantly improve the performance of DeepONets in extrapolation tasks. Integrating them into our current research would substantially change and expand our paper's scope. Therefore, we plan to apply these techniques in future work, building upon the foundational research that we have conducted thus far.

Furthermore, we aim to expand our research from bi-fidelity to multi-fidelity and Bayesian operator learning [28,42–44]. This expansion aims to optimize the use of data while mitigating the computational costs associated with predicting computationally expensive data. The ultimate goal is to enhance the model's prediction by maximizing generalization and accuracy. As a result, our research not only contributes to the current understanding but also lays the foundation for future innovations in predicting lift and drag coefficients' time trajectories.

Finally, we plan to use novel Bayesian multi-fidelity operator learning frameworks to optimize complex dynamical systems [45] or predict network dynamical systems [46] in a distributed and federated [47] manner.

5. Conclusions

This paper presents a framework based on physics-guided bi-fidelity Fourier-featured deep operator learning. It aims to predict the time trajectories of lift and drag coefficients for a cylinder inside a channel. The proposed approach uses deep operator neural networks (DeepONets) to design a physics-guided strategy that approximates the operator mapping between undisturbed free-stream velocity and the corresponding drag and lift coefficients over time. The framework approximates both coarse and fine operators, which map undisturbed free-stream velocity to the target time trajectories. The latter is achieved by integrating physics-guided low-fidelity and residual deep operator networks. To accurately capture the oscillatory nature inherent in these coefficients, our physics-guided strategy employs sinusoidal family functions, converting the prediction into an operator/functional inverse problem. Simultaneously, we incorporate the Fourier features network as the DeepONet's trunk, further enhancing the capture of these fluctuations through random Fourier mapping. To assess the predictive capabilities of the physics-guided approach against a data-driven strategy for lift and drag coefficients, we evaluated both methodologies using test datasets. We established three different configurations within our bi-fidelity framework and examined their performance on these test datasets. The results underscore that the physics-guided Fourier-featured DeepONet possesses an enhanced capacity for predicting

the time trajectories of lift and drag, outperforming its data-driven counterparts. This proficiency directly enhances the accuracy of the bi-fidelity framework. Both the low-fidelity and proposed bi-fidelity framework effectively predict the drag and lift coefficients' time trajectories. As a result, our physics-guided bi-fidelity Fourier-featured framework offers improved performance with fewer high-fidelity data.

Author Contributions: Conceptualization, A.M., I.S., C.M., G.P. and G.L.; Methodology, I.S., C.M. and G.L.; Software, A.M., I.S. and C.M.; Validation, A.M., I.S., I.R. and C.M.; Formal analysis, A.M. and I.S.; Investigation, A.M., I.S., I.R., C.M. and G.L.; Resources, G.P. and G.L.; Data curation, I.S.; Writing—original draft, A.M. and I.S.; Writing—review & editing, I.S., I.R., C.M., G.P. and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: I.S. gratefully acknowledges the support of Lillian Gilbreth Postdoctoral Fellowships from Purdue University's College of Engineering. I.R. gratefully acknowledges the support of International Research Support Initiative Program (IRSIP), Higher Education Commission (HEC) of Pakistan. G.L. gratefully acknowledges the support of the National Science Foundation (DMS-2053746, DMS-2134209, ECCS-2328241, and OAC-2311848), and U.S. Department of Energy (DOE) Office of Science Advanced Scientific Computing Research program DE-SC0021142 and DE-SC0023161.

Data Availability Statement: The data presented in this paper are openly available in <https://github.com/Mollaali/Bifidelity-Learning-Physics-DeepONet-CFD>.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 14 December 2023).
- Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [[CrossRef](#)]
- Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [[CrossRef](#)]
- Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
- Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [[CrossRef](#)]
- Goswami, S.; Yin, M.; Yu, Y.; Karniadakis, G.E. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114587. [[CrossRef](#)]
- Wang, S.; Wang, H.; Perdikaris, P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Sci. Adv.* **2021**, *7*, eabi8605. [[CrossRef](#)] [[PubMed](#)]
- Cai, S.; Wang, Z.; Lu, L.; Zaki, T.A.; Karniadakis, G.E. DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *J. Comput. Phys.* **2021**, *436*, 110296.
- Zhang, Z.; Wing Tat, L.; Schaeffer, H. Belnet: Basis enhanced learning, a mesh-free neural operator. *Proc. R. Soc.* **2023**, *479*, 20230043. [[CrossRef](#)]
- Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv* **2020**, arXiv:2010.08895.
- Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Stuart, A.; Bhattacharya, K.; Anandkumar, A. Multipole graph neural operator for parametric partial differential equations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6755–6766.
- Wen, G.; Li, Z.; Azizzadenesheli, K.; Anandkumar, A.; Benson, S.M. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Adv. Water Resour.* **2022**, *163*, 104180. [[CrossRef](#)]
- Kovachki, N.; Li, Z.; Liu, B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Neural operator: Learning maps between function spaces. *arXiv* **2021**, arXiv:2108.08481.
- Han, J.; Ou, S.; Park, J.; Lei, C. Augmented heat transfer in rectangular channels of narrow aspect ratios with rib turbulators. *Int. J. Heat Mass Transf.* **1989**, *32*, 1619–1630. [[CrossRef](#)]
- Paniagua, G.; De nos, R.; Almeida, S. Effect of the Hub Endwall Cavity Flow on the Flow-Field of a Transonic High-Pressure Turbine. *J. Turbomach.* **2004**, *126*, 578–586. [[CrossRef](#)]
- Leifsson, L.; Koziel, S. Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction. *J. Comput. Sci.* **2010**, *1*, 98–106. [[CrossRef](#)]
- Leifsson, L.; Koziel, S. Aerodynamic shape optimization by variable-fidelity computational fluid dynamics models: A review of recent progress. *J. Comput. Sci.* **2015**, *10*, 45–54. [[CrossRef](#)]

18. Sen, O.; Gaul, N.J.; Choi, K.; Jacobs, G.; Udaykumar, H. Evaluation of multifidelity surrogate modeling techniques to construct closure laws for drag in shock–particle interactions. *J. Comput. Phys.* **2018**, *371*, 434–451. [\[CrossRef\]](#)
19. Du, X.; Ren, J.; Leifsson, L. Aerodynamic inverse design using multifidelity models and manifold mapping. *Aerosp. Sci. Technol.* **2019**, *85*, 371–385. [\[CrossRef\]](#)
20. Demiral, E.; Sahin, C.; Arslan, K. Aerodynamic Shape Optimization Using Multi-fidelity Surrogate-based Approach for Computationally Expensive Problems. In Proceedings of the AIAA AVIATION 2022 Forum, Chicago, IL, USA, 27 June–1 July 2022. [\[CrossRef\]](#)
21. Robinson, T.; Willcox, K.; Eldred, M.; Haimes, R. Multifidelity Optimization for Variable-Complexity Design. In Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, USA, 6–8 September 2006. [\[CrossRef\]](#)
22. Huang, L.; Gao, Z.; Zhang, D. Research on multi-fidelity aerodynamic optimization methods. *Chin. J. Aeronaut.* **2013**, *26*, 279–286. [\[CrossRef\]](#)
23. Jaeggi, D.; Parks, G.; Dawes, W.; Clarkson, J. Robust Multi-Fidelity Aerodynamic Design Optimization Using Surrogate Models. In Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada, 10–12 September 2008. [\[CrossRef\]](#)
24. Han, Z.H.; Görtz, S.; Zimmermann, R. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerosp. Sci. Technol.* **2013**, *25*, 177–189. [\[CrossRef\]](#)
25. Vapnik, V.N.; Izmailov, R. Learning using privileged information: Similarity control and knowledge transfer. *J. Mach. Learn. Res.* **2015**, *16*, 2023–2049.
26. Dehghani, M.; Mehrjou, A.; Gouws, S.; Kamps, J.; Schölkopf, B. Fidelity-Weighted Learning. *arXiv* **2017**, arXiv:1711.02799.
27. De, S.; Britton, J.; Reynolds, M.; Skinner, R.; Jansen, K.; Doostan, A. On Transfer Learning of Neural Networks Using Bi-fidelity Data For Uncertainty Propagation. *Int. J. Uncertain. Quantif.* **2020**, *10*, 543–573. [\[CrossRef\]](#)
28. Lu, L.; Pestourie, R.; Johnson, S.G.; Romano, G. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Phys. Rev. Res.* **2022**, *4*, 023210. [\[CrossRef\]](#)
29. Meng, X.; Karniadakis, G.E. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *J. Comput. Phys.* **2020**, *401*, 109020. [\[CrossRef\]](#)
30. De, S.; Reynolds, M.; Hassanaly, M.; King, R.N.; Doostan, A. Bi-fidelity modeling of uncertain and partially unknown systems using DeepONets. *Comput. Mech.* **2023**, *71*, 1217–1251. [\[CrossRef\]](#)
31. Moya, C.; Lin, G.; Zhao, T.; Yue, M. On Approximating the Dynamic Response of Synchronous Generators via Operator Learning: A Step Towards Building Deep Operator-based Power Grid Simulators. *arXiv* **2023**, arXiv:2301.12538.
32. Zhang, Z.; Moya, C.; Leung, W.T.; Lin, G.; Schaeffer, H. Bayesian deep operator learning for homogenized to fine-scale maps for multiscale PDE. *arXiv* **2023**, arXiv:2308.14188.
33. Howard, A.A.; Perego, M.; Karniadakis, G.E.; Stinis, P. Multifidelity deep operator networks for data-driven and physics-informed problems. *J. Comput. Phys.* **2023**, *493*, 112462. [\[CrossRef\]](#)
34. Howard, A.A.; Murphy, S.H.; Ahmed, S.E.; Stinis, P. Stacked networks improve physics-informed training: applications to neural networks and deep operator networks. *arXiv* **2023**, arXiv:2311.06483.
35. Lin, G.; Moya, C.; Zhang, Z. Learning the dynamical response of nonlinear non-autonomous dynamical systems with deep operator neural networks. *Eng. Appl. Artif. Intell.* **2023**, *125*, 106689. [\[CrossRef\]](#)
36. Schäfer, M.; Turek, S.; Durst, F.; Krause, E.; Rannacher, R. Benchmark Computations of Laminar Flow Around a Cylinder. In *Flow Simulation with High-Performance Computers II*; Hirschel, E.H., Ed.; Vieweg and Teubner Verlag: Wiesbaden, Germany, 1996; pp. 547–566.
37. Sieber, M.; Paschereit, C.O.; Oberleithner, K. Spectral proper orthogonal decomposition. *J. Fluid Mech.* **2016**, *792*, 798–828. [\[CrossRef\]](#)
38. Chen, T.; Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* **1995**, *6*, 911–917. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7537–7547.
40. Lu, L.; Meng, X.; Cai, S.; Mao, Z.; Goswami, S.; Zhang, Z.; Karniadakis, G.E. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Comput. Methods Appl. Mech. Eng.* **2022**, *393*, 114778. [\[CrossRef\]](#)
41. Zhu, M.; Zhang, H.; Jiao, A.; Karniadakis, G.E.; Lu, L. Reliable extrapolation of deep neural operators informed by physics or sparse observations. *Comput. Methods Appl. Mech. Eng.* **2023**, *412*, 116064. [\[CrossRef\]](#)
42. Howard, A.A.; Perego, M.; Karniadakis, G.E.; Stinis, P. Multifidelity deep operator networks. *arXiv* **2022**, arXiv:2204.09157.
43. Lin, G.; Moya, C.; Zhang, Z. B-DeepONet: An enhanced Bayesian DeepONet for solving noisy parametric PDEs using accelerated replica exchange SGLD. *J. Comput. Phys.* **2023**, *473*, 111713. [\[CrossRef\]](#)
44. Moya, C.; Zhang, S.; Lin, G.; Yue, M. DeepONet-grid-ug: A trustworthy deep operator framework for predicting the power grid's post-fault trajectories. *Neurocomputing* **2023**, *535*, 166–182. [\[CrossRef\]](#)

45. Sahin, I.; Moya, C.; Mollaali, A.; Lin, G.; Paniagua, G. Deep operator learning-based surrogate models with uncertainty quantification for optimizing internal cooling channel rib profiles. *Int. J. Heat Mass Transf.* **2024**, *219*, 124813. [[CrossRef](#)]
46. Sun, Y.; Moya, C.; Lin, G.; Yue, M. Deepgraphonet: A deep graph operator network to learn and zero-shot transfer the dynamic response of networked systems. *IEEE Syst. J.* **2023**, *17*, 4360–4370. [[CrossRef](#)]
47. Moya, C.; Lin, G. Fed-DeepONet: Stochastic Gradient-Based Federated Training of Deep Operator Networks. *Algorithms* **2022**, *15*, 325. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.