METAP: HOW TO TRANSFER YOUR KNOWLEDGE ON LEARNING HIDDEN PHYSICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-based meta-learning methods have primarily focused on classical machine learning tasks such as image classification and function regression, where they were found to perform well by recovering the underlying common representation among a set of given tasks. Recently, PDE-solving deep learning methods, such as neural operators, are starting to make an important impact on learning and predicting the response of a complex physical system directly from observational data. Since the data acquisition in this context is commonly challenging and costly, the call of utilization and transfer of existing knowledge to new and unseen physical systems is even more acute.

Herein, we propose a novel meta-learnt approach for transfer-learning knowledge between neural operators, which can be seen as transferring the knowledge of solution operators between governing (unknown) PDEs with varying parameter fields. With the key theoretical observation that the underlying parameter field can be captured in the first layer of the neural operator model, in contrast to typical final-layer transfer in existing meta-learning methods, our approach is a provably universal solution operator for multiple PDE solving tasks. As applications, we demonstrate the efficacy of our proposed approach on PDE-based datasets and a real-world material modeling problem, demonstrating that our method can handle complex and nonlinear physical response learning tasks while greatly improving the sampling efficiency in new and unseen tasks.

1 INTRODUCTION

Few-shot learning is an important problem in machine learning, where new tasks are learned with a very limited number of labelled datapoints (Wang et al., 2020). In recent years, significant progress has been made on few-shot learning using meta-learning approaches (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Finn et al., 2017; Santoro et al., 2016; Antoniou et al., 2018; Ravi & Larochelle, 2016; Nichol & Schulman, 2018; Raghu et al., 2019; Tripuraneni et al., 2021; Collins et al., 2022). Broadly speaking, given a family of tasks, some of which are used for training and others for testing, meta-learning approaches aim to learn a shared multi-task representation that can generalize across the different training tasks, and result in fast adaptation to new and unseen testing tasks. Although most of meta-learning learning developments focus on conventional machine learning problems such as image classification, function regression, and reinforcement learning, studies on few-shot learning approaches for complex physical system modeling problems have been limited. The call of developing a few-shot learning approach for complex physical system modeling problems is just as acute, while the typical understanding of how multi-task learning should be applied on this scenario is still nascent.

As an motivating example, we consider the scenario of new material discovery in the lab environment, where the material model is built based on experimental measurements of its responses subject to different loadings. Since the physical properties (such as the mechanical and structural parameters) in different material specimens vary, the model learnt from experimental measurements on one specimen would have a large generalization error on future specimens. That means, the data-driven model has to be trained repeatedly with a large number of material specimens, which makes the learning process inefficient. Further, experimental measurement acquisition of these specimens is often challenging and expensive. In some problems, a large amount of measurements are not even

feasible. For example, in the design and testing of biosynthetic tissues, performing repeated loading would potentially induces the cross-linking and permanent set phenomenon, which notoriously alter the tissue durability (Zhang & Sacks, 2017). As a result, it is critical to learn the physical response model of a new specimen with samples sizes as small as possible. Furthermore, since many characterization methods to obtain underlying material mechanistic and structural properties would require the use of destructive methods (Misfeld & Sievers, 2007; Rieppo et al., 2008), in practice many physical properties are not measured and can only be treated as hidden and unknown variables. We likely only have limited access to the measurements on the complex system responses caused by the change of these physical properties.

Herein, we consider the meta-learning of multiple complex physical systems (as tasks), such that all these tasks are governed by a common PDE with different (hidden) physical parameter fields. Formally, assume that we have a distribution $p(\mathcal{T})$ over tasks, each task $\mathcal{T}^\eta \sim p(\mathcal{T})$ corresponds to a hidden physical property field $\mathbf{b}^\eta(\mathbf{x}) \in \mathcal{B}(\mathbb{R}^{d_b})$ that contains the task-specific mechanistic and structural information in our material modeling example. On task $\mathcal{T}^\eta$, we have a number of observations on the loading field $\mathbf{g}_i^\eta(\mathbf{x}) \in \mathcal{A}(\mathbb{R}^{d_g})$ and the corresponding physical system response field $\mathbf{u}_i^\eta(\mathbf{x}) \in \mathcal{U}(\mathbb{R}^{d_u})$. Here, i is the sample index, $\mathcal{B}$, $\mathcal{A}$ and $\mathcal{U}$ are Banach spaces of function taking values in $\mathbb{R}^{d_b}$, $\mathbb{R}^{d_g}$ and $\mathbb{R}^{d_u}$, respectively. Let $\Omega \subset \mathbb{R}^s$ be the domain of interests, we assume that all task can be modeled with a parametric PDE form

$$\mathcal{K}_{\mathbf{b}^\eta(\mathbf{x})}[\mathbf{u}_i^\eta](\mathbf{x}) = \mathbf{g}_i^\eta(\mathbf{x}), \quad \mathbf{x} \in \Omega. \tag{1}$$

Here, $\mathcal{K}_\mathbf{b}$ is the operator representing the possibly unknown governing law, e.g., balance laws.

For task $\mathcal{T}^\eta$, our modeling goal is to learn the solution operator $\mathcal{G}^\eta : \mathcal{A} \to \mathcal{U}$ of equation 7, such that the learnt model can predict the corresponding physical response field $\mathbf{u}(\mathbf{x})$ for any loading field $\mathbf{g}(\mathbf{x})$. Without transfer learning, one needs to learn a surrogate solution operator for each task only based on the data pairs on this task, and repeat the training for every task. The learning procedure would require a relatively large amount of observation pairs and training time for each task. Therefore, this physical-based modeling scenario raises a key question: *Given knowledge on a number of parametric PDE solving tasks with different unknown parameters, how can one efficiently learn the best surrogate solution operator for a new and unknown parameter, with only a small set of training data pairs?*

To address this question, we introduce MetaP, a novel meta-learnt approach for transfer-learning knowledge between neural operators, which can be seen as transferring the knowledge of solution operators between governing (unknown) PDEs with varying hidden parameter fields. Our **main contributions** are:

- MetaP is the first neural-operator-based approach for multi-task learning, which not only preserves the generalizability to different resolutions and input functions from its integral neural operator architecture, but also improves sampling efficiency on new tasks – for comparable accuracy, MetaP saves the required number of measurements by $\sim$90%.
- With rigorous operator approximation analysis, we made the key observation that the hidden parameter field can be captured by adapting the first layer of the neural operator model, in contrast to typical final-layer transfer in existing meta-learning methods. By construction, MetaP serves as a provably universal solution operator for multiple PDE solving tasks.
- From synthetic, benchmark, to a real-world biological tissue datasets, the proposed method consistently outperforms existing baseline gradient-based meta-learning methods.

2 Background and Related Work

In this section we introduce the relevant materials on hidden physics learning, neural operators, and gradient-based meta-learning methods, which will later complement the definition of our method.

2.1 Hidden Physics Learning and Neural Operators

In real-world physical problems, predicting and monitoring complex system responses are ubiquitous in many applications. For many decades, physics-based PDEs have been commonly employed

for modeling these systems, then traditional numerical methods were employed to solve these PDEs and provide predictions for desired system responses. However, three fundamental challenges exist in utilizing the classical PDE-based modeling approach in modeling complex physical systems in real-world applications. First, the choice of governing PDE laws is often determined *a priori* and free parameters are often tuned to obtain agreement with experimental data. This fact makes the rigorous calibration and validation process challenging. Second, traditional numerical methods are solved for specific boundary and initial conditions, as well as loading or source terms. Therefore, they are not generalizable for other operating conditions and hence not effective for real-time prediction. Third, complex PDE systems such as turbulence flows and heterogeneous materials modeling problems usually require a very fine discretization, and therefore very challenging and time-consuming for traditional solvers.

To provide an efficient surrogate model for physical responses, machine learning methods may hold the key. Recently, there has been significant progress in the development of deep neural networks (NNs), focusing on learning the hidden physics of a complex system (Ghaboussi et al., 1998; 1991; Carleo et al., 2019; Karniadakis et al., 2021; Zhang et al., 2018; Cai et al., 2022; Pfau et al., 2020; He et al., 2021; Besnard et al., 2006). Among these methods, the neural operators show particular promises in resolving the above challenges. In contrast to classical neural networks which maps between finite-dimensional vectors, neural operators aim to learn maps between inputs of a dynamical system and its state, so that the network can serve as a surrogate for a solution operator (Li et al., 2020a;b;c; You et al., 2022a; Ong et al., 2022; Gupta et al., 2021; Lu et al., 2019; 2021a; Goswami et al., 2022a). Comparing with the classical NNs, the most notable advantages of neural operators are resolution independence and generalizability to different input instances. The former implies that the accuracy of the prediction is invariant with respect to the resolution of input parameters such as loading conditions, while the later renders a computing advantage on prediction efficiency – once the neural operator is trained, solving for a new instance of the input parameter only requires a forward pass of the network. Moreover, comparing with the classical PDE modeling approaches, neural operators requires no knowledge of the underlying PDE, only data. All these advantages make neural operators promising tools to PDE learning tasks. Examples include modeling the unknown physics law of real-world problems (Yin et al., 2022a; Goswami et al., 2022a; Yin et al., 2022b), and providing efficient solution operator for PDEs (Li et al., 2020a;b;c; Lu et al., 2021b).

On the other hand, data in scientific applications are often scarce and incomplete, which calls for a learning approach of low sample requirements and high generalization capabilities. Although neural operators possess generalizability across different resolutions and input instances such as loading conditions, to the best of our knowledge, no existing work have addressed the transferability of multiple neural operators on the same type of PDE system but with different physical properties. Through the meta-learning techniques, our work fulfills the demand of this few-shot learning problem.

2.2 Gradient-Based Meta-Learning Methods

One highly successful meta-learning algorithm has been Model Agnostic Meta-Learning (MAML) (Finn et al., 2017), which led to the development of a series of related gradient-based meta-learning (GBML) methods Raghu et al. (2019); Nichol & Schulman (2018); Antoniou et al. (2018); Hospedales et al. (2020). Although MAML does not explicitly try to learn a useful representation, in (Raghu et al., 2019) it was found that MAML tends to learn a shared representation across tasks then performing local adaptation on its final layer. Based on such observations, a modified version of MAML, known as the Almost-No-Inner-Loop algorithm (ANIL) (Raghu et al., 2019) was proposed, which freezes the representation during local adaptation. Recently, theoretical analysis (Collins et al., 2022) was provided for MAML and ANIL on multi-task linear representation learning scenarios, and found that the driving force causing MAML and ANIL to recover the underlying representation is the adaptation of the final layer of their models, which harnesses the underlying task diversity to improve the representation in all directions of interest.

Although MAML and the general meta-learning approaches have achieved impressive performance in some machine-learning applications such as the image classification and reinforcement learning scenarios, a few work (Kailkhura et al., 2019; Mai et al., 2021; Zhang et al., 2022; Goswami et al., 2022b) has studied the hidden physics learning under meta or even transfer setting. None of these works have provided a generic approach nor theoretical understanding on how to transfer the multi-

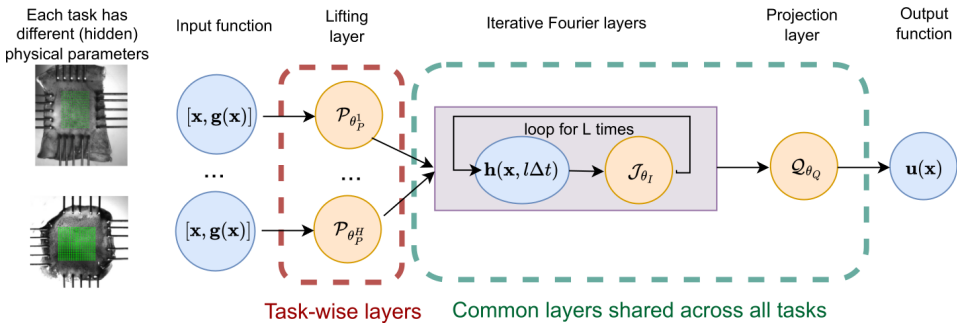


Figure 1: The architecture of MetaP based on an integral neural operator model.

task knowledge between a series of complex physical systems, such that all these tasks are governed by a common parametric PDE with different physical parameters.

3 Meta-Learnt Neural Operator

3.1 Integral Neural Operators

Here, we first state the base model of this work without the meta aspect. The integral neural operators, first proposed in (Li et al., 2020a) and further developed in (Li et al., 2020b;c; You et al., 2022a;c) comprises of three building blocks. First, the input function, $\mathbf{g}(\mathbf{x}) \in \mathcal{A}$, is lifted to a higher dimensional representation via $\mathbf{h}(\mathbf{x}, 0) = \mathcal{P}[\mathbf{g}](\mathbf{x}) := P(\mathbf{x})[\mathbf{x}, \mathbf{g}(\mathbf{x})]^T + \mathbf{p}(\mathbf{x})$. Here, $P(\mathbf{x}) \in \mathbb{R}^{(s+d_g)\times d_h}$ and $\mathbf{p}(\mathbf{x}) \in \mathbb{R}^{d_h}$ define an affine pointwise mapping, which are often taken as constant parameters, i.e., $P(\mathbf{x}) \equiv P$ and $\mathbf{p}(\mathbf{x}) \equiv \mathbf{p}$. Then, the feature vector function $\mathbf{h}(\mathbf{x}, 0)$ goes through an iterative layer block where the layer update is defined via the action of the sum of a local linear operator, a nonlocal integral kernel operator, and a bias function: $\mathbf{h}(\cdot, (l+1)\Delta t) = \mathcal{J}_{l+1}[\mathbf{h}(\cdot, l\Delta t)]$, for $l = 0, \cdots, L-1$. Here, $\mathbf{h}(\cdot, j\Delta t)$, $j = 0, \cdots, L := T/\Delta t$, is a sequence of functions representing the values of the network at each hidden layer, taking values in $\mathbb{R}^{d_h}$. $\mathcal{J}_1, \cdots, \mathcal{J}_L$ are the nonlinear operator layers, defined by the particular choice of networks. In this work, we employ the implicit Fourier neural operator (IFNO) as the base model, because of its theoretical universal approximation property in PDE solving tasks (You et al., 2022c) and robustness in complex physical response modeling tasks (You et al., 2022b)[1]. In this case, the iterative layers are taken as $\mathcal{J}_1 = \cdots = \mathcal{J}_L = \mathcal{J}$, where

$$\begin{aligned} \mathbf{h}(\mathbf{x}, (l+1)\Delta t) &= \mathcal{J}[\mathbf{h}(\mathbf{x}, l\Delta t)] \\ &:= \mathbf{h}(\mathbf{x}, l\Delta t) + \Delta t \sigma \left(W\mathbf{h}(\mathbf{x}, l\Delta t) + \mathcal{F}^{-1}[\mathcal{F}[\kappa(\cdot; \mathbf{v})] \cdot \mathcal{F}[\mathbf{h}(\cdot, l\Delta t)]](\mathbf{x}) + \mathbf{c}(\mathbf{x}) \right). \end{aligned} \tag{2}$$

Here, $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively. In practice, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are computed using the FFT and its inverse to each component of $\mathbf{h}$ separately. Also, $\mathbf{c} \in \mathbb{R}^{d_h}$ defines a constant bias, $W \in \mathbb{R}^{d_h \times d_h}$ is the weight matrix, and $\mathcal{F}[\kappa(\cdot; \mathbf{v})] := R$ is a circulant matrix that depends on the convolution kernel κ. σ is an activation function, which is oftenly taken to be the popular rectified linear unit (ReLU) function. Finally, the output $\mathbf{u}(\cdot) \in \mathcal{U}$ is obtained through a projection layer. In particular, we project the last hidden layer representation $\mathbf{h}(\cdot, T)$ onto $\mathcal{U}$ as: $\mathbf{u}(\mathbf{x}) = \mathcal{Q}[\mathbf{h}(\cdot, T)](\mathbf{x}) := Q_2(\mathbf{x})\sigma(Q_1\mathbf{h}(\mathbf{x}, T) + \mathbf{q}_1(\mathbf{x})) + \mathbf{q}_2(\mathbf{x})$. Here, $Q_1(\mathbf{x}) \in \mathbb{R}^{d_Q \times d_h}$, $Q_2(\mathbf{x}) \in \mathbb{R}^{d_u \times d_Q}$, $\mathbf{q}_1(\mathbf{x}) \in \mathbb{R}^{d_Q}$ and $\mathbf{q}_2(\mathbf{x}) \in \mathbb{R}^{d_u}$ are the appropriately sized matrices and vectors that are part of the parameter set that we aim to learn. Similarly as for the lifting layer, $Q_1(\mathbf{x})$, $Q_2(\mathbf{x})$, $\mathbf{q}_1(\mathbf{x})$ and $\mathbf{q}_2(\mathbf{x})$ are also often taken as constant parameters, which will be denoted as Q_1, Q_2, $\mathbf{q}_1$ and $\mathbf{q}_2$, respectively. In the following, we denote the set trainable parameters in the lifting layer as θ_P, the parameter from the iterative layer block as θ_I, and the parameters in the projection layer as θ_Q.

As such, the neural operator can be employed to learn an approximation for the solution operator, $\mathcal{G}$, of equation 7. Let $\mathcal{D} := \{(\mathbf{g}_i, \mathbf{u}_i)\}_{i=1}^N$ be a support set of observations where the input $\{\mathbf{g}_i\} \subset \mathcal{A}$ is a

[1]We also point out that the proposed multi-task strategy is generic and hence also applicable to any other integral neural operators (Li et al., 2020a;b;c; You et al., 2022a).

sequence of independent and identically distributed (i.i.d.) random fields from a known probability distribution μ on $\mathcal{A}$, and $\mathbf{u}_i(\mathbf{x}) \in \mathcal{U}$, possibly noisy, is the observed corresponding solution. We then learn the system response by constructing a surrogate operator: $\tilde{\mathcal{G}}[\mathbf{g};\theta](\mathbf{x}) := \mathcal{Q}_{\theta_Q} \circ (\mathcal{J}_{\theta_I})^L \circ \mathcal{P}_{\theta_P}[\mathbf{g}](\mathbf{x}) \approx \mathbf{u}(\mathbf{x})$, where the parameter set $\theta = [\theta_P, \theta_I, \theta_Q]$ is obtained by solving the optimization problem:

$$\min_{\theta\in\Theta} \mathcal{L}_{\mathcal{D}}(\theta) = \min_{\theta\in\Theta} \mathbb{E}_{\mathbf{f}\sim\mu}[C(\tilde{\mathcal{G}}[\mathbf{g};\theta], \mathcal{G}[\mathbf{g}])] \approx \min_{\theta\in\Theta} \sum_{i=1}^{N} [C(\tilde{\mathcal{G}}[\mathbf{g}_i;\theta], \mathbf{u}_i)]. \tag{3}$$

Here C denotes a properly defined cost functional which is often taken as the the mean square error.

3.2 Base Meta Model with MAML and ANIL

To transfer the multi-task knowledge between a series of complex systems governed by different hidden physical parameters, we proposed to leverage the integral neural operator with a meta-learning setting. A straightforward approach would be to simply apply MAML and ANIL to a neural operator architecture, which will be treated as the baselines of our studies. Here we formally state our implementation of ANIL and MAML for the problem described above.

MAML. The MAML algorithm aims to find an initialization, $\tilde{\theta}$, across all tasks, so that new tasks can be learnt with very few examples. First, we draw a batch $\{\mathcal{T}^\eta\}_{\eta=1}^{H}$ of H tasks from $p(\mathcal{T})$. For each task $\mathcal{T}^\eta$, we split the available loading field/response field data pairs $\mathcal{D}^\eta := \{(\mathbf{g}_i^\eta(\mathbf{x}), \mathbf{u}_i^\eta(\mathbf{x}))\}_{i=1}^{N_\eta}$ to a support set of samples, $\mathcal{S}^\eta$, which will be used for inner loop updates, and a target set of samples, $\mathcal{Z}^\eta$, for outer loop updates. Then, for the inner loop we let $\theta_0^\eta := \tilde{\theta}$ and $\theta^{\eta,i}$ be the task-wise parameter after i gradient updates. During each inner loop update, we compute

$$\theta^{\eta,i} = \theta^{\eta,i-1} - \alpha \nabla_{\theta^{\eta,i-1}} \mathcal{L}_{\mathcal{S}^\eta}(\theta^{\eta,i-1}), \quad \text{for } \eta = 1, \cdots, H, \tag{4}$$

where $\mathcal{L}_{\mathcal{S}^\eta}(\theta^{\eta,i-1})$ is the loss on the support set of the η-th task, and α is the step size. After m inner loop updates, we update the initial parameter $\tilde{\theta}$ with a fixed step size β:

$$\tilde{\theta} \leftarrow \tilde{\theta} - \beta \nabla_{\tilde{\theta}} \mathcal{L}_{\text{meta}}(\tilde{\theta}), \text{ where the meta-loss } \mathcal{L}_{\text{meta}}(\tilde{\theta}) := \sum_{\eta=1}^{H} \mathcal{L}_{\mathcal{Z}^\eta}(\theta^{\eta,m}). \tag{5}$$

Then, on the test task, $\mathcal{T}^{\text{test}}$, we perform inner loop adaptation based on few labeled samples $\mathcal{D}^{\text{test}} := \{(\mathbf{g}_i^{\text{test}}(\mathbf{x}), \mathbf{u}_i^{\text{test}}(\mathbf{x}))\}_{i=1}^{N^{\text{test}}}$ and obtain the approximated solution operator model on the test task as $\tilde{\mathcal{G}}[\mathbf{g};\theta^{\text{test},m}]$.

ANIL. In , ANIL was proposed as a modified version of MAML with inner loop updates only for the final layer. As such, the inner loop update formulation equation 4 is modified as

$$\theta_Q^{\eta,i} = \theta_Q^{\eta,i-1} - \alpha \nabla_{\theta_Q^{\eta,i-1}} \mathcal{L}_{\mathcal{S}^\eta}(\theta_Q^{\eta,i-1}), \quad \text{for } \eta = 1, \cdots, H, \tag{6}$$

where $\theta_Q^{\eta,i}$ is the task-wise parameter on the projection layer after i gradient updates. Then, we perform the same outer loop updates following equation 5.

3.3 MetaP: A Novel Meta-Learnt Neural Operator Architecture

We now propose MetaP, which *applies task-wise adaptation only to the first layer, i.e., the lifting layer*, with the full algorithm outlined in Algorithm 1. Similar as in other meta-learning approaches (Yoon et al., 2018; Vanschoren, 2018; Yang & Kwok, 2022; Kalais & Chatzis, 2022), the algorithm consists of two phases: (1) a meta-train phase which learns shared iterative layers parameters θ_I and projection layer parameters θ_P from existing tasks; (2) a meta-test phase which transfers the learned knowledge and rapidly learning surrogate solution operators for unseen tasks with unknown physical parameter field, where only a few test samples are required.

To see the inspiration of the proposed architecture, without loss of generality, we assume that the underlying task parameter field $\mathbf{b}^\eta(\mathbf{x})$ is normalized and satisfying $\left|\left|\mathbf{b}^\eta(\mathbf{x}) - \overline{\mathbf{b}}(\mathbf{x})\right|\right|_{L^2(\Omega)} \leq 1$ for all $\eta \in \{1, \cdots, H\}$, where $\overline{\mathbf{b}} := \mathbb{E}_{\mathcal{T}^\eta\sim p(\mathcal{T})}[\mathbf{b}^\eta]$. Denoting $\mathcal{F}_{\mathbf{u}}[\mathbf{b}] := \mathcal{K}_{\mathbf{b}}[\mathbf{u}]$ as a function from $\mathcal{B}$ to $\mathcal{A}$, we can take the Fréchet derivative of $\mathcal{F}$ with respect to $\mathbf{b} - \overline{\mathbf{b}}$ and obtain:

$$\mathcal{K}_{\mathbf{b}^\eta}[\mathbf{u}] = \mathcal{F}_{\mathbf{u}}[\overline{\mathbf{b}}] + D\mathcal{F}_{\mathbf{u}}[\overline{\mathbf{b}}](\mathbf{b}^\eta - \overline{\mathbf{b}}) + o(\left|\left|\mathbf{b}^\eta - \overline{\mathbf{b}}\right|\right|_{L^2(\Omega)}).$$

Meta-Train Phase:
Require: a batch $\{\mathcal{T}^\eta\}_{\eta=1}^H$ of known tasks and available data pairs $\mathcal{D}^\eta := \{(\mathbf{g}_i^\eta(\mathbf{x}), \mathbf{u}_i^\eta(\mathbf{x}))\}_{i=1}^{N^\eta}$ on each task
Output: common parameters θ_I^* and θ_Q^* across all tasks
1. randomly initialize θ_I, θ_Q, and $\{\theta_P^\eta\}_{\eta=1}^H$
2. solve the optimization problem:

$$\{\theta_I^*, \theta_Q^*, \{\theta_P^{\eta,*}\}_{\eta=1}^H\} = \underset{\{\theta_I, \theta_Q, \{\theta_P^\eta\}_{\eta=1}^H\}}{\operatorname{argmin}} \sum_{\eta=1}^H \mathcal{L}_{\mathcal{D}^\eta}([\theta_P^\eta, \theta_I, \theta_Q])$$

Meta-Test Phase:
Require: a test task $\mathcal{T}^{\text{test}}$ and few labelled data pairs $\mathcal{D}^{\text{test}} := \{(\mathbf{g}_i^{\text{test}}(\mathbf{x}), \mathbf{u}_i^{\text{test}}(\mathbf{x}))\}_{i=1}^{N^{\text{test}}}$
Output: the task-wise parameter $\theta_P^{\text{test},*}$ and the corresponding surrogate PDE solution operator $\tilde{\mathcal{G}}[\mathbf{g}; [\theta_P^{\text{test},*}, \theta_I^*, \theta_Q^*]](\mathbf{x})$ for the test task
3. solve for the lift layer parameter from the optimization problem:

$$\theta_P^{\text{test},*} = \underset{\theta_P^{\text{test}}}{\operatorname{argmin}} \mathcal{L}_{\mathcal{D}^{\text{test}}}([\theta_P^{\text{test}}, \theta_I^*, \theta_Q^*])$$

Algorithm 1: MetaP for Few-Shot Learning of New PDE Solver with Hidden Physical Parameters

Substituting the above formulation into equation 1, we obtain

$$\mathcal{F}_{\mathbf{u}_i^\eta}[\overline{\mathbf{b}}] + D\mathcal{F}_{\mathbf{u}_i^\eta}[\overline{\mathbf{b}}](\mathbf{b}^\eta - \overline{\mathbf{b}}) \approx \mathbf{g}_i^\eta.$$

Denoting $\mathbf{F}_1[\mathbf{b}^\eta] := [\mathbf{1}, \mathbf{b}^\eta - \overline{\mathbf{b}}]$ and $\mathbf{F}_2[\mathbf{u}_i^\eta] := [\mathcal{F}_{\mathbf{u}_i^\eta}[\overline{\mathbf{b}}], D\mathcal{F}_{\mathbf{u}_i^\eta}[\overline{\mathbf{b}}]]$, we can actually reformulated equation 1 into a more generic form:

$$\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}) \cdot \mathbf{F}_2[\mathbf{u}_i^\eta](\mathbf{x}) = \mathbf{g}_i^\eta(\mathbf{x}), \quad \mathbf{x} \in \Omega. \tag{7}$$

We point out that this parametric PDE form is indeed very general and finds applications in many science and engineering applications – besides our motivating example on material modeling, examples also include the monitoring of biological tissue degeneration problems (Zhang & Sacks, 2017), the detection of subsurface flows (Dejam et al., 2017), the nondestructive evaluation and inspection in aviation (Fallah et al., 2019), and the prediction of concrete structures deterioration (Wei et al., 2021), just to name a few.

In the following, we show that MetaP are universal solution finding operators for the multi-task problem equation 7, in the sense that they can approximate a fixed point method to a desired accuracy. For simplicity, we consider a $1D$ domain Ω, and scalar functions $\mathbf{F}_1[\mathbf{b}^\eta]$, $\mathbf{F}_2[\mathbf{u}_i^\eta]$. These functions are assumed to be sufficiently smooth and measured at uniformly distributed nodes $\chi := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, with $\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_j) \neq 0$ for all η and j. Then, equation 7 can be formulated as an implicit system of equations:

$$\mathcal{H}(\mathbf{U}_i^{\eta,*}; \tilde{\mathbf{G}}_i^\eta) := \begin{bmatrix} \mathbf{F}_2[\mathbf{u}_i^\eta](\mathbf{x}_1) - \mathbf{g}_i^\eta(\mathbf{x}_1)/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_1) \\ \vdots \\ \mathbf{F}_2[\mathbf{u}_i^\eta](\mathbf{x}_M) - \mathbf{g}_i^\eta(\mathbf{x}_M)/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_M) \end{bmatrix} = \mathbf{0}, \tag{8}$$

where $\mathbf{U}_i^{\eta,*} := [\mathbf{u}_i^\eta(\mathbf{x}_1), \mathbf{u}_i^\eta(\mathbf{x}_2), \ldots, \mathbf{u}_i^\eta(\mathbf{x}_M)]$ is the solution we seek, $\tilde{\mathbf{G}}_i^\eta := [\mathbf{g}_i^\eta(\mathbf{x}_1)/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_1), \mathbf{g}_i^\eta(\mathbf{x}_2)/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_2), \ldots, \mathbf{g}_i^\eta(\mathbf{x}_M)/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_M)]$ is the reparameterized loading vector, and $\mathbf{G}_i^\eta := [\mathbf{g}_i^\eta(\mathbf{x}_1), \mathbf{g}_i^\eta(\mathbf{x}_2), \ldots, \mathbf{g}_i^\eta(\mathbf{x}_M)]$ is the original loading vector. Here, we notice that when seeing equation 8 as an implicit problem of $\mathbf{U}_i^{\eta,*}$ and $\tilde{\mathbf{G}}_i^\eta$, it is actually independent of the task parameter field $\mathbf{b}^\eta$, i.e., this problem is task-independent. Therefore, in the later contents we refer to equation 8 without the task index, as $\mathcal{H}(\mathbf{U}^*; \tilde{G})$ for notation simplicity. Taking $\mathbf{U}^0 := [\mathbf{x}_1, \cdots, \mathbf{x}_M]$ as the initial guess, we aim to show that for any desired accuracy $\varepsilon > 0$, one can find a sufficiently large $L > 0$ and sets of parameters $\theta^\eta = \{\theta_P^\eta, \theta_I, \theta_Q\}$, such that the resultant MetaP model satisfies

$$\left|\left|\mathcal{Q}_{\theta_Q} \circ (\mathcal{J}_{\theta_I})^L \circ \mathcal{P}_{\theta_P^\eta}([\mathbf{U}^0, \mathbf{G}^\eta]^{\mathrm{T}}) - \mathbf{U}^{\eta,*}\right|\right|_{l^2(\mathbb{R}^M)} \leq \varepsilon, \quad \forall \mathbf{G}^\eta \in \mathbb{R}^M.$$

To get our main theoretical results, we need the assumption that there exists a sufficiently good iterative method, i.e., the following properties on $\mathbf{U}^*$ and $\mathcal{R}$ hold true:

Assumption 1. *There exists a fixed point equation,* $\mathbf{U} = \mathbf{U} + \mathcal{R}(\mathbf{U}, \tilde{\mathbf{G}})$ *for the implicit problem equation 8, such that* $\mathcal{R} : \mathbb{R}^{2M} \mapsto \mathbb{R}^M$ *is a continuous function satisfying* $\mathcal{R}(\mathbf{U}, \tilde{\mathbf{G}}) = \mathbf{0}$ *and* $||\mathcal{R}(\hat{\mathbf{U}}, \tilde{\mathbf{G}}) - \mathcal{R}(\tilde{\mathbf{U}}, \tilde{\mathbf{G}})||_{l^2(\mathbb{R}^M)} \leq m||\hat{\mathbf{U}} - \tilde{\mathbf{U}}||_{l^2(\mathbb{R}^M)}$ *for any two vectors* $\hat{\mathbf{U}}, \tilde{\mathbf{U}} \in \mathbb{R}^M$*. Here,* $m > 0$ *is a constant independent of* $\tilde{\mathbf{G}}$*.*

Assumption 2. *With the initial guess* $\mathbf{U}^0 := [\mathbf{x}_1, \cdots, \mathbf{x}_M]$*, the fixed-point iteration*

$$\mathbf{U}^{l+1} = \mathbf{U}^l + \mathcal{R}(\mathbf{U}^l, \tilde{\mathbf{G}}), \quad l = 0, 1, \ldots.$$

converges, i.e., for any given $\varepsilon > 0$*, there exists an integer* L *such that*

$$||\mathbf{U}^l - \mathbf{U}^*||_{l^2(\mathbb{R}^M)} \leq \varepsilon, \quad \forall l > L,$$

for all possible input instances $\tilde{\mathbf{G}} \in \mathbb{R}^M$ *and their corresponding solutions* $\mathbf{U}^*$*.*

Then, we will have our universal approximation theorem for MetaP as below, with its detailed proof provided in Appendix:

Theorem 1 (Universal approximation)**.** *Given Assumptions 1-2, let the activation function* σ *for all iterative kernel integration layers be the ReLU function, and the activation function in the projection layer be the identity function. Then for any* $\varepsilon > 0$*, there exist sufficiently large layer number* $L > 0$ *and feature dimension number* $d_h > 0$*, such that one can find a parameter set for the multi-task problem,* $\theta^\eta = [\theta_P^\eta, \theta_I, \theta_Q]$*, such that the corresponding IFNO model satisfies*

$$\left|\left|\mathcal{Q}_{\theta_Q} \circ (\mathcal{J}_{\theta_I})^L \circ \mathcal{P}_{\theta_P^\eta}([\mathbf{U}^0, \mathbf{G}^\eta]^{\mathrm{T}}) - \mathbf{U}^{\eta,*}\right|\right| \leq \varepsilon, \quad \forall \mathbf{G}^\eta \in \mathbb{R}^M,$$

for all tasks.

4 Experiments

In this section, we demonstrate the empirical effectiveness of the proposed MetaP approach. Specifically, we conduct experiments on a synthetic dataset from a nonlinear PDE solving problem, a benchmark dataset of heterogeneous materials subject to large deformation, and a real-world dataset from biological tissue mechanical testing, and compare the proposed method against competitive GBML baselines. All of the experiments are tested using PyTorch with Adam optimizer. For a fair comparison, for each method, we tune the hyperparameters, including the learning rates, the decay rates, and the regularization coefficient, to minimize the error on a validation dataset. In all tests we considered the averaged relative error, $||\mathbf{u}_{i,pred} - \mathbf{u}_i||_{L^2(\Omega)}/||\mathbf{u}_i||_{L^2(\Omega)}$, as the error metric (lower means better).

4.1 Synthetic Data Sets and Ablation Study

We first consider the PDE solution finding problem of the Holzapfel-Gasser-Odgen (HGO) model (Holzapfel et al., 2000), which describes the deformation of hyperelastic, anisotropic, and fiber-reinforced materials. Here in, different tasks correspond to different material parameter sets, $\{k_1, k_2, E, \nu, \alpha\}$, where k_1 and k_2 are fiber modulus and the exponential coefficient, respectively, E is the Young's modulus, ν is the Poisson ratio, and α is the fiber angle direction from the reference direction. For more details of the model we refer interested readers to (Holzapfel et al., 2000; Yin et al., 2022b). In this example the physical response of interests is the displacement field $\mathbf{u} : [0, 1]^2 \to \mathbb{R}^2$ subject to different traction loadings applied on the top edge of this material. Therefore, we take the input function $\mathbf{g}(\mathbf{x})$ as the padded traction loading field, and the output function as the corresponding displacement field.

To investigate the performance of MetaP in few-shot learning, we generate 60 training tasks and 5 in-distribution test tasks by sampling the physical parameters $k_1, k_2 \sim \mathcal{U}[0.1, 1]$, $E \sim \mathcal{U}[0.5, 1.5]$, $\nu \sim \mathcal{U}[0.1, 0.49]$, and $\alpha \sim \mathcal{U}[\pi/10, \pi/2]$. Here $\mathcal{U}$ stands for uniform distribution. To further evaluate the generalizability when the physical parameters in test tasks are outside the training regime, we also generate 5 out-of-distribution test tasks with physical parameters from different distributions, where $k_1, k_2 \sim \mathcal{U}[1, 1.5]$, $E \sim \mathcal{U}[1.5, 2]$, $\nu \sim \mathcal{U}[0.01, 0.1]$, and $\alpha \sim \mathcal{U}[\pi/2, 3\pi/4]$. For each training task, we generate 500 data pairs $\mathcal{D}^\eta := \{(\mathbf{g}_i^\eta, \mathbf{u}_i^\eta)\}_{i=1}^{500}$, by sampling the vertical traction loading from a Gaussian random field. Then, the corresponding ground-truth displacement field is obtained using

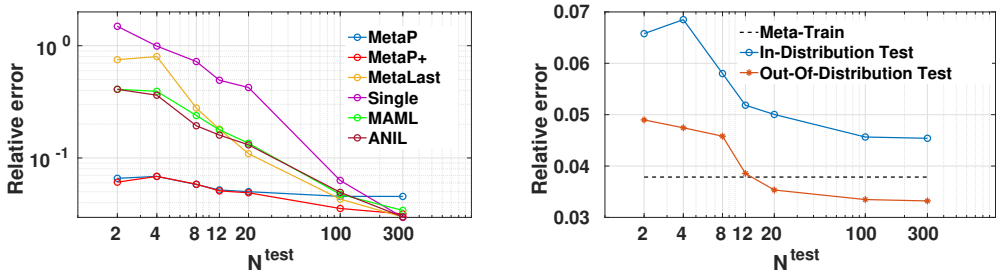


Figure 2: Results on a synthetic data set. Left: The ablation study comparison on test errors in the in-distribution test. Right: The relative error of MetaP in in-distribution and out-of distribution tests.

the finite element method implemented in FEniCS (Alnæs et al., 2015). For the test tasks, we train with $N^{\text{test}} = \{2, 4, 8, 12, 20, 100, 300\}$ numbers of labelled data pairs, and evaluate the resultant model on an additional dataset with 200 data pairs. An 8-layer IFNO is employed as the base model.

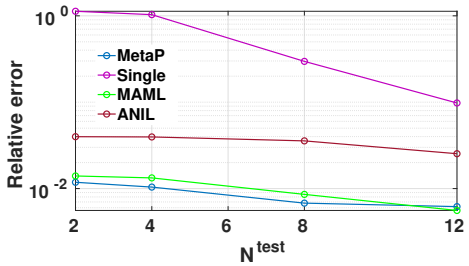
Ablation Study. We first conduct an ablation study on our proposed algorithm, listed in Algorithm 1, with three settings. 1) Follow the meta-train and meta-test phases as in Algorithm 1, with task-wise adaptation only to the lifting layer in both phases (denotes as "MetaP"). 2) After MetaP, perform an additional fine-tuning step to all parameters in the meta-test phase (denotes as "MetaP+"). With this test, we aim to investigate if our algorithm has successfully identified all the common features in the iterative and projection layers. 3) Apply task-wise adaptation only to the projection layer in both meta-train and meta-test phases (denoted as "MetaLast"). With this test, we study if the successful "adapting last layers" strategy in image classification problems would also apply for our PDE solving problem. Besides these three settings, we also report the few-shot learning results with three baseline methods. 1) Learn a neural operator model only based on the data set, $\mathcal{D}^{\text{test}}$, on the test task (denoted as "Single"). 2) MAML. 3) ANIL. As shown in the left plot of Figure 2, MetaP and MetaP+ are both able to quickly adapt with few data pairs – to achieve a test error below 5%, "Single" requires 100 data pairs, while MetaP and MetaP+ requires only 2 data pairs. On the other hand, MetaLast, MAML and ANIL have similar performance. They only narrowly outperform "Single", and all require 100 data pairs to achieve a $< 5\%$ test error. This observation verifies our theoretical analysis: on the multi-task parametric PDE solution operator learning problem, one should adapt the first layer, not the last ones. Moreover, when comparing MetaP and MetaP+ we can see that the additional fine-tune step barely improves the performance, especially in the few-sample regime. This fact verifies the efficacy of MetaP, and indicates that our method has successfully capture the underlying task diversity by adapting the first layer, so no further fine-tuning is required.

In-Distribution and Out-Of-Distribution Tests. On the right plot of Figure 2, we demonstrate the relative test error of MetaP in both in-distribution and out-of-distribution tasks, with the averaged meta-train error on the training tasks in dashed line as a reference. We can see that these two test errors are both in a similar scale as the error on training tasks. Surprisingly, the error from out-of-distribution test tasks is even slightly smaller than the in-distribution test error. These results validates the good generalization performance of MetaP.

4.2 Benchmark Mechnical MNIST Datasets

To further test the capability of MetaP on benchmark datasets, we test MetaP and the three baseline methods on Mechanical MNIST (Lejeune, 2020). Mechanical MNIST is a dataset of heterogeneous material undergoing large deformation. It contains 70,000 heterogeneous material specimens, and each specimen is governed by the Neo-Hookean material with a varying modulus converted from the MNIST bitmap images. On each specimen, 32 loading/response data pairs are provided[2], under the uniaxial extension, shear, equibiaxial extension, and confined compression load scenarios, respectively. Here in, we randomly select 100 specimens as the training tasks, and a new and unseen specimen as the test task. On the meta-test phase, we reserve 20 data pairs on the test task for eval-

[2]We have excluded small deformation samples, i.e., samples with the maximum displacement magnitude ≤ 0.1.



Model	$N^{\text{test}} = 12$	$N^{\text{test}} = 500$
MetaP	**0.71%**	0.64%
Single	6.74%	**0.41%**
MAML	10.28 %	0.79%
ANIL	12.74 %	0.77%

Figure 3: Comparison of MetaP and three baseline methods on the benchmark dataset (MechnicalMNIST, left plot) and the real-world dataset (heart valve tissue, right table, with bold cases highlighting the best models).

uation, then train each model under the few-shot learning setting with $N^{\text{test}} = \{2, 4, 8, 12\}$ labelled data pairs. All approaches are developed based on an 8-layer IFNO model.

We present the results in the left plot of Figure 3. The neural operator model learned by MetaP again outperforms the state-of-the-art GBML models. Moreover, we notice that our MetaP model achieves $< 5\%$ when using only 1 labelled data pair on the test task, while the "Single" model has around 100% error even using 8 labelled data pairs, due to overfitting. This fact highlights the importance of learning across multi-tasks in engineering applications – when the total number of measurements on each specimen is limited, it is necessary to transfer the knowledge across specimens. Moreover, we notice that in this example ANIL is the least effective GBML method, probably due to the inefficacy of the adapting last layers strategy.

4.3 Application on Real-World Data Sets

We now take a step further to demonstrate the performance of our method on a real-world physical response dataset which is NOT generated by sovling PDEs. We consider the problem of learning the mechanical response of multiple biological tissue specimens from DIC displacement tracking measurements. As demonstrated in Figure 1, we measure the biaxial loading of tricuspid valve anterior leaflet (TVAL) specimens from a porcine heart, such that each specimen (as a task) corresponds to a different region of the leaflet. Due to the material heterogeneity of biological tissues, these specimens are with different mechanical and structural properties.

In this task, we aim to model the tissue response by learning a neural operator mapping the boundary displacement loading to the interior displacement field on each tissue specimen. On each specimen, we have 500 available data pairs. Due to the challenges of obtaining the experimental tissue, only 12 specimens are available for the purpose of meta-train, and the performance will be evaluated on another specimen. Therefore, this example also stands for a common challenging setting in real-world applications: we not only have the few-shot learning challenge, but also suffer from the difficulty from limited available training tasks. With a 4-layer IFNO as the base model, we train MetaP with the three baseline approaches based on $N^{\text{test}} = \{12, 500\}$ samples, then evaluate the performance on another 200 samples. The results are provided on the right of Figure 3. As one can see, MetaP performs the best with low data samples among all the methods, and still beat our MAML and ANIL variants when $N^{\text{test}} = 500$. Interestingly, MAML and ANIL did not even beat the "Single" method, possibly due to the low number of training tasks.

5 Conclusion

In this paper we propose the first neural-operator-based meta-learning approach that are designed to achieve good transferability in learning complex physical system responses with significant improvement in sample efficiency. The first layer adaption used by our method is theoretically motivated and shown to be the universal solution operator for multiple PDE solving tasks. We demonstrate the effectiveness of our proposed MetaP algorithm on various synthetic and real-world datasets, showing promises over baseline methods. For future work, we will investigate the applicability of the proposed approach to other neural operators.

REFERENCES

Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

Gilles Besnard, François Hild, and Stéphane Roux. "finite-element" displacement fields analysis from digital images: application to portevin–le châtelier bands. *Experimental mechanics*, 46(6): 789–803, 2006.

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, pp. 1–12, 2022.

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

Liam Collins, Aryan Mokhtari, Sewoong Oh, and Sanjay Shakkottai. Maml and anil provably learn representations. *arXiv preprint arXiv:2202.03483*, 2022.

Morteza Dejam, Hassan Hassanzadeh, and Zhangxin Chen. Pre-darcy flow in porous media. *Water Resources Research*, 53(10):8187–8210, 2017.

F Fallah, Mazdak Ghajari, and Yasser Safa. Computational modelling of dynamic delamination in morphing composite blades and wings. *The International Journal of Multiphysics*, 13(4):393–430, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, volume 70, pp. 1126–1135. PMLR, 2017.

J Ghaboussi, JH Garrett Jr, and Xiping Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics*, 117(1):132–153, 1991.

Jamshid Ghaboussi, David A Pecknold, Mingfu Zhang, and Rami M Haj-Ali. Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, 42(1):105–126, 1998.

Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed neural operators. *2022 arXiv preprint arXiv:2207.05748*, 2022a.

Somdatta Goswami, Katiana Kontolati, Michael D Shields, and George Em Karniadakis. Deep transfer learning for partial differential equations under conditional shift with deeponet. *arXiv preprint arXiv:2204.09810*, 2022b.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=LZDiWaC9CGL`.

Qizhi He, Devin W Laurence, Chung-Hao Lee, and Jiun-Shyan Chen. Manifold learning based data-driven modeling for soft biological tissues. *Journal of Biomechanics*, 117:110124, 2021.

Gerhard A Holzapfel, Thomas C Gasser, and Ray W Ogden. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity and the Physical Science of Solids*, 61(1):1–48, 2000.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *2020 arXiv preprint arXiv:2004.05439*, 2020.

Bhavya Kailkhura, Brian Gallagher, Sookyung Kim, Anna Hiszpanski, and T Yong-Jin Han. Reliable and explainable machine-learning methods for accelerated material discovery. *npj Computational Materials*, 5(1):1–9, 2019.

Konstantinos Kalais and Sotirios Chatzis. Stochastic deep networks with linear competing units for model-agnostic meta-learning. In *International Conference on Machine Learning*, pp. 10586–10597. PMLR, 2022.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pp. 0. Lille, 2015.

Emma Lejeune. Mechanical mnist: A benchmark dataset for mechanical metamodels. *Extreme Mechanics Letters*, 36:100659, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:NeurIPS 2020, 2020b.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier NeuralOperator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2020c.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021a.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *arXiv preprint arXiv:2111.05512*, 2021b.

Haoxin Mai, Tu C Le, Takashi Hisatomi, Dehong Chen, Kazunari Domen, David A Winkler, and Rachel A Caruso. Use of meta models for rapid discovery of narrow bandgap oxide photocatalysts. *iScience*, pp. 103068, 2021.

Martin Misfeld and Hans-Hinrich Sievers. Heart valve macro-and microstructure. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1484):1421–1436, 2007.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Yong Zheng Ong, Zuowei Shen, and Haizhao Yang. IAE-NET: Integral autoencoders for discretization-invariant learning. 03 2022. doi: 10.13140/RG.2.2.25120.87047/2.

David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Jarno Rieppo, Jarmo Hallikainen, Jukka S Jurvelin, Ilkka Kiviranta, Heikki J Helminen, and Mika M Hyttinen. Practical considerations in the use of polarized light microscopy in the analysis of the collagen network in articular cartilage. *Microscopy research and technique*, 71(4):279–287, 2008.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Nilesh Tripuraneni, Chi Jin, and Michael Jordan. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pp. 10434–10443. PMLR, 2021.

Joaquin Vanschoren. Meta-learning: A survey. *2018 arXiv preprint arXiv:1810.03548*, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Congjie Wei, Charles S Wojnar, and Chenglin Wu. Hydro-chemo-mechanical phase field formulation for corrosion induced cracking in reinforced concrete. *Cement and Concrete Research*, 144: 106404, 2021.

Hansi Yang and James Kwok. Efficient variance reduction for meta-learning. In *International Conference on Machine Learning*, pp. 25070–25095. PMLR, 2022.

Minglang Yin, Ehsan Ban, Bruno V Rego, Enrui Zhang, Cristina Cavinato, Jay D Humphrey, and George Em Karniadakis. Simulating progressive intramural damage leading to aortic dissection using DeepONet: an operator–regression neural network. *Journal of the Royal Society Interface*, 19(187):20210670, 2022a.

Minglang Yin, Enrui Zhang, Yue Yu, and George Em Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Computer Methods in Applied Mechanics and Engineering, in press*, pp. 115027, 2022b.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 7343–7353, 2018.

Huaiqian You, Yue Yu, Marta D'Elia, Tian Gao, and Stewart Silling. Nonlocal kernel network (NKN): A stable and resolution-independent deep neural network. *Journal of Computational Physics*, pp. arXiv preprint arXiv:2201.02217, 2022a.

Huaiqian You, Quinn Zhang, Colton J Ross, Chung-Hao Lee, Ming-Chen Hsu, and Yue Yu. A physics-guided neural operator learning approach to model biological tissues from digital image correlation measurements. *Journal of Biomechanical Engineering, accepted for publication*, pp. arXiv preprint arXiv:2204.00205, 2022b.

Huaiqian You, Quinn Zhang, Colton J Ross, Chung-Hao Lee, and Yue Yu. Learning deep implicit fourier neural operators (IFNOs) with applications to heterogeneous material modeling. *Computer Methods in Applied Mechanics and Engineering*, 398:115296, 2022c. doi: https://doi.org/10.1016/j.cma.2022.115296.

Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and E Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14):143001, 2018.

Lu Zhang, Huaiqian You, and Yue Yu. Metanor: A meta-learnt nonlocal operator regression approach for metamaterial modeling. *arXiv preprint arXiv:2206.02040*, 2022.

Will Zhang and Michael S Sacks. Modeling the response of exogenously crosslinked tissue to cyclic loading: The effects of permanent set. *Journal of the Mechanical Behavior of Biomedical Materials*, 75:336–350, 2017.

A PROOF OF THEOREM 1

Here, we prove that the MetaP is universal, i.e., give a fixed point method satisfying Assumptions 1-2, one can find parameter sets θ^η whose output approximates $\mathbf{U}^{\eta,*}$ to a desired accuracy, $\varepsilon > 0$, for all $\eta = 1, \cdots, H$ tasks. For the task-wise parameters, with a slight abuse of notation, we denote $P^\eta \in \mathbb{R}^{d_h M \times (d_g+s)M}$ as the collection of the pointwise weight matrices at each discretization point in χ for the η-th task, and $\mathbf{p}^\eta \in \mathbb{R}^{d_h M}$ for the bias in the lifting layer. Then, for the parameters shared among all tasks, in the iterative layer we denote $\mathbf{C} = [\mathbf{c}(\mathbf{x}_1), \cdots, \mathbf{c}(\mathbf{x}_M)] \in \mathbb{R}^{d_h M}$ as the collection of pointwise bias vectors $\mathbf{c}(\mathbf{x}_i)$, $W \in \mathbb{R}^{d_h \times d_h}$ for the local linear transformation, and $R = \mathcal{F}[\kappa(\cdot;\mathbf{v})] \in \mathbb{C}^{d_h \times d_h \times M} \in \mathbb{C}^{d_h \times d_h \times M}$ for the Fourier coefficients of the kernel κ. For simplicity, here we have assumed that the Fourier coefficient is not truncated, and all available frequencies are used. Then, for the projection layer we seek $Q_1 \in \mathbb{R}^{d_Q M \times d_h M}$, $Q_2 \in \mathbb{R}^{d_u M \times d_Q M}$, $\mathbf{q}_1 \in \mathbb{R}^{d_Q M}$ and $\mathbf{q}_2 \in \mathbb{R}^{d_u M}$. For the simplicity of notation, in this section we organize the feature vector $\mathbf{H} \in \mathbb{R}^{d_h M}$ in a way such that the components corresponding to each discretization point are adjacent, i.e., $\mathbf{H} = [\mathbf{H}(\mathbf{x}_1), \cdots, \mathbf{H}(\mathbf{x}_M)]$ and $\mathbf{H}(\mathbf{x}_i) \in \mathbb{R}^{d_h}$.

We point out that under this circumstance, we have the (discretized) iterative layer can be written as

$$\begin{aligned}\mathcal{J}[\mathbf{H}(l\Delta t)] =& \mathbf{H}(l\Delta t) + \Delta t \sigma \left(\tilde{W}\mathbf{H}(l\Delta t) + \mathrm{Re}(\mathcal{F}_{\Delta x}^{-1}(R \cdot \mathcal{F}_{\Delta x}(\mathbf{H}(l\Delta t)))) + \mathbf{C}\right) \\ =& \mathbf{H}(l\Delta t) + \Delta t \sigma \left(V\mathbf{H}(l\Delta t) + \mathbf{C}\right),\end{aligned}$$

with

$$V := \mathrm{Re}\begin{bmatrix} \sum_{n=0}^{M-1} R_{n+1} + W & \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi\Delta x n}{M}) & \dots & \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi(M-1)\Delta x n}{M}) \\ \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi\Delta x n}{M}) & \sum_{n=0}^{M-1} R_{n+1} + W & \dots & \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi(M-2)\Delta x n}{M}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi(M-1)\Delta x n}{M}) & \sum_{n=0}^{M-1} R_{n+1}\exp(\frac{2i\pi(M-2)\Delta x n}{M}) & \dots & \sum_{n=0}^{M-1} R_{n+1} + W \end{bmatrix}.$$

Here, $R \in \mathbb{C}^{M \times d_h \times d_h}$ with $R_i \in \mathbb{C}^{d_h \times d_h}$ being the component associated with each discretization point $\mathbf{x}_i \in \chi$, $V \in \mathbb{R}^{d_h M \times d_h M}$, $\mathbf{C} \in \mathbb{R}^{d_h M}$, $\tilde{W} := W \oplus W \oplus \cdots \oplus W$ is a $d_h M \times d_h M$ block diagonal matrix formed by $W \in \mathbb{R}^{d_h \times d_h}$, $\mathcal{F}_{\Delta x}$ and $\mathcal{F}_{\Delta x}^{-1}$ denote the discrete Fourier transform and its inverse, respectively. By further taking $R_2 = \cdots = R_M = W = 0$, a $d_h \times d_h$ matrix with all its elements being zero, it suffices to show the universal approximation property for an iterative layer as follows:

$$\mathcal{J}(\mathbf{H}(l\Delta t)) := \mathbf{H}(l\Delta t) + \Delta t \sigma \left(\tilde{V}\mathbf{H}(l\Delta t) + \mathbf{C}\right)$$

where $\tilde{V} := \mathbf{1}_{[M,M]} \otimes V$ with $V \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{1}_{[m,n]}$ being an m by n all-ones matrix.

To be more precise, we will prove the following theorem:

Theorem 1 (Universal approximation)**.** *Let $\mathbf{U}^{\eta,*} = [\mathbf{u}(\mathbf{x}_1), \mathbf{u}(\mathbf{x}_2), \dots, \mathbf{u}(\mathbf{x}_M)]$ be the ground-truth solution of η-th task that satisfies Assumptions 1-2, the activation function σ for all iterative kernel integration layers be the ReLU function, and the activation function in the projection layer be the identity function. Then for any $\varepsilon > 0$, there exist sufficiently large layer number $L > 0$ and feature dimension number $d_h > 0$, such that one can find a parameter set for the multi-task problem, $\theta^\eta = [\theta_P^\eta, \theta_I, \theta_Q]$ with the corresponding IFNO model satisfies*

$$\left|\left|\mathcal{Q}_{\theta_Q} \circ (\mathcal{J}_{\theta_I})^L \circ \mathcal{P}_{\theta_P^\eta}([\mathbf{U}^0, \mathbf{G}^\eta]^{\mathrm{T}}) - \mathbf{U}^{\eta,*}\right|\right| \leq \varepsilon, \quad \forall \mathbf{G}^\eta \in \mathbb{R}^M.$$

For the proof of this main theorem, we need the following approximation property of a shallow neural network, with its detailed proof provided in You et al. (2022c):

Lemma 1. *Given a continuous function $\mathcal{T} : \mathbb{R}^{2M} \mapsto \mathbb{R}^M$, and a non-polynomial and continuous activation function σ, for any constant $\hat{\varepsilon} > 0$ there exists a shallow neural network model $\hat{\mathcal{T}} := S\sigma(B\mathbf{X} + A)$ such that*

$$||\mathcal{T}(\mathbf{X}) - \hat{\mathcal{T}}(\mathbf{X})||_{l^2(\mathbb{R}^M)} \leq \hat{\varepsilon}, \quad \forall \mathbf{X} \in \mathbb{R}^{2M},$$

for sufficiently large feature dimension $\hat{d} > 0$. Here, $S \in \mathbb{R}^{M \times \hat{d}M}$, $B \in \mathbb{R}^{\hat{d}M \times 2M}$, and $A \in \mathbb{R}^{\hat{d}M}$ are matrices/vectors which are independent of $\mathbf{X}$.

We now proceed to the proof of Theorem 1:

Proof. Since all $\mathbf{U}^{\eta,*}$ satisfies Assumptions 1-2, for any $\varepsilon > 0$, we first pick a sufficiently large integer L such that the L-th layer iteration result of this fixed point formulation satisfies $||\mathbf{U}^L - \mathbf{U}^{\eta,*}||_{l^2(\mathbb{R}^M)} \leq \frac{\varepsilon}{2}$ for all tasks. By taking $\hat{\varepsilon} := \frac{m\varepsilon}{2(1+m)^L}$ in Lemma 1, there exists a sufficiently large feature dimension $\hat{d}$ and one can find $S \in \mathbb{R}^{M\times\hat{d}M}$, $B \in \mathbb{R}^{\hat{d}M\times 2M}$, and $A \in \mathbb{R}^{\hat{d}M}$, such that $\hat{\mathcal{R}}(\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta) := S\sigma(B[\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta]^{\mathrm{T}} + A)$ satisfies

$$||\mathcal{R}(\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta) - \hat{\mathcal{R}}(\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta)||_{l^2(\mathbb{R}^M)} = ||\mathcal{R}(\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta) - S\sigma(B[\mathbf{U}^\eta, \tilde{\mathbf{G}}^\eta]^{\mathrm{T}} + A)||_{l^2(\mathbb{R}^M)} \leq \hat{\varepsilon} = \frac{m\varepsilon}{2(1+m)^L},$$

where m is the contraction parameter of $\mathcal{R}$, as defined in Assumption 1. By this construction, we know that S has independent rows. Denoting $\tilde{d} := \hat{d} + 1 > 0$, there exists the right inverse of S, which we denote as $S^+ \in \mathbb{R}^{(\tilde{d}-1)M\times M}$, such that

$$SS^+ = I_M, \quad S^+S := \tilde{I}_{(\tilde{d}-1)M},$$

where I_M is the M by M identity matrix, $\tilde{I}_{(\tilde{d}-1)M}$ is a $(\tilde{d}-1)M$ by $(\tilde{d}-1)M$ block matrix with each of its element being either 1 or 0. Hence, for any vector $Z \in \mathbb{R}(\tilde{d}-1)M$, we have $\sigma(\tilde{I}_{(\tilde{d}-1)M}Z) = \tilde{I}_{(\tilde{d}-1)M}\sigma(Z)$. Moreover, we note that S has a very special structure: from the $((i-1)(\tilde{d}-1)+1)$-th to the $(i(\tilde{d}-1))$-th column of S, all nonzero elements are on its i-th row. Correspondingly, we can also choose S^+ to have a special structure: from the $((i-1)(\tilde{d}-1)+1)$-th to the $(i(\tilde{d}-1))$-th row of S^+, all nonzero elements are on its i-th column. Hence, when multiplying S^+ with $\mathbf{U}$, there will be no entanglement between different components of $\mathbf{U}$. That means, S^+ can be seen as a pointwise weight function.

We now construct the MetaP as follows. In this construction, we choose the feature dimension as $d_h := \tilde{d}M$. With the input $[\mathbf{U}^0, \mathbf{G}^\eta] \in \mathbb{R}^{2M}$, for the lift layer we set

$$P^\eta := \mathbf{1}_{[M,1]} \otimes \begin{bmatrix} S^+ & \mathbf{0} \\ \mathbf{0} & D^\eta \end{bmatrix} = \underbrace{\begin{bmatrix} S^+ & \mathbf{0} & S^+ & \mathbf{0} & \cdots & S^+ & \mathbf{0} \\ \mathbf{0} & D^\eta & \mathbf{0} & D^\eta & \cdots & \mathbf{0} & D^\eta \end{bmatrix}^{\mathrm{T}}}_{\text{repeated for } M \text{ times}} \in \mathbb{R}^{d_h M\times 2M},$$

and $\mathbf{p}^\eta := \mathbf{0} \in \mathbb{R}^{d_h M}$. Here, $D^\eta := \mathrm{diag}[1/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_1), \cdots, 1/\mathbf{F}_1[\mathbf{b}^\eta](\mathbf{x}_M)]$. As such, the initial layer of feature is then given by

$$\mathbf{H}^{\eta,0} = P^\eta([\mathbf{U}^0, \mathbf{G}^\eta]^{\mathrm{T}}) = \mathbf{1}_{[M,1]} \otimes [S^+\mathbf{U}^0, D^\eta\mathbf{G}^\eta]^{\mathrm{T}} = \mathbf{1}_{[M,1]} \otimes [S^+\mathbf{U}^0, \tilde{\mathbf{G}}^\eta]^{\mathrm{T}} \in \mathbb{R}^{dM}.$$

Here, we point out that P^η and $\mathbf{p}^\eta$ can be seen as pointwise weight and bias functions, respectively.

Next we construct the shared iterative layer $\mathcal{J}$, by setting

$$V := \begin{bmatrix} \tilde{I}_{(\tilde{d}-1)M}B/M \\ 0 \end{bmatrix} \begin{bmatrix} S/\Delta t & \mathbf{0} \\ \mathbf{0} & I_M/\Delta t \end{bmatrix}, \ \tilde{V} := \mathbf{1}_{[M,M]} \otimes V, \ \text{ and } \boldsymbol{C} := \mathbf{1}_{[M,1]} \otimes \begin{bmatrix} \tilde{I}_{(\tilde{d}-1)M}A/\Delta t \\ \mathbf{0} \end{bmatrix}.$$

Note that $\tilde{V}$ is independent of η, and falls into the formulation of V, by letting $R_1 = V$ and $R_2 = R_2 = \cdots = R_M = W = 0$. For the $l+1$-th layer of feature vector, we then arrive at

$$\begin{aligned}
&\mathbf{H}((l+1)\Delta t) = \mathbf{H}(l\Delta t) + \Delta t\sigma\left(\tilde{V}\mathbf{H}(l\Delta t) + \mathbf{C}\right) \\
&= \mathbf{H}(l\Delta t) + \left(I_M \otimes \begin{bmatrix} S^+S & \mathbf{0} \\ \mathbf{0} & I_M \end{bmatrix}\right)\sigma\left(\left(\mathbf{1}_{[M,1]} \otimes \begin{bmatrix} B/M \\ \mathbf{0} \end{bmatrix}\right)\left(\mathbf{1}_{[1,M]} \otimes \begin{bmatrix} S & \mathbf{0} \\ \mathbf{0} & I_M \end{bmatrix}\right)\mathbf{H}(l\Delta t) + \mathbf{1}_{[M,1]} \otimes \begin{bmatrix} A \\ \mathbf{0} \end{bmatrix}\right),
\end{aligned}$$

where $\mathbf{H}(l\Delta t) = [\hat{\mathbf{h}}_1^{l\Delta t}, \hat{\mathbf{h}}_2^{l\Delta t}, \ldots, \hat{\mathbf{h}}_{2M-1}^{l\Delta t}, \hat{\mathbf{h}}_{2M}^{l\Delta t}]^{\mathrm{T}}$ denotes the (spatially discretized) hidden layer feature at the $l-$th iterative layer of the IFNO. Subsequently, we note that the second part of the feature vector, $\hat{\mathbf{h}}_{2j}^{l\Delta t} \in \mathbb{R}^M$, satisfies

$$\hat{\mathbf{h}}_{2j}^{(l+1)\Delta t} = \hat{\mathbf{h}}_{2j}^{l\Delta t} = \cdots = \hat{\mathbf{h}}_{2j}^0 = \tilde{\mathbf{G}}^\eta, \quad \forall l = 0, \cdots, L-1, \forall j = 1, \cdots, M$$

Hence, the first part of the feature vector, $\hat{\mathbf{h}}_{2j-1}^{l\Delta t} \in \mathbb{R}^{(\tilde{d}-1)M}$, satisfies the following iterative rule:

$$\hat{\mathbf{h}}_{2j-1}^{(l+1)\Delta t} = \hat{\mathbf{h}}_{2j-1}^{l\Delta t} + S^+S\sigma(B[S\hat{\mathbf{h}}_{2j-1}^{l\Delta t}, \tilde{\mathbf{G}}^\eta]^{\mathrm{T}} + A), \quad \forall l = 0, \cdots, L-1, \forall j = 1, \cdots, M,$$

and

$$\hat{\mathbf{h}}_1^{(l+1)\Delta t} = \hat{\mathbf{h}}_3^{(l+1)\Delta t} = \cdots = \hat{\mathbf{h}}_{2M-1}^{(l+1)\Delta t}.$$

Finally, for the projection layer $\mathcal{Q}$, we set the activation function in the projection layer as the identity function, $Q_1 := I_{d_h M}$ (the identity matrix of size $d_h M$), $Q_2 := [S, \mathbf{0}] \in \mathbb{R}^{M \times d_h M}$, $\mathbf{q}_1 := \mathbf{0} \in \mathbb{R}^{d_h M}$, and $\mathbf{q}_2 := \mathbf{0} \in \mathbb{R}^M$. Denoting the output $\mathbf{U}^\eta := \mathcal{Q}_{\theta_P} \circ (\mathcal{J}_{\theta_I})^L \circ \mathcal{P}_{\theta_L^\eta}([\mathbf{U}^0, \mathbf{G}^\eta]^{\mathrm{T}})$, we now show that $\mathbf{U}^\eta$ can approximate $\mathbf{U}^{\eta,*}$ with a desired accuracy ε:

$$\begin{aligned}
||\mathbf{U}^\eta - \mathbf{U}^{\eta,*}|| &\leq ||\mathbf{U}^\eta - \mathbf{U}^L||_{l^2(\mathbb{R}^M)} + ||\mathbf{U}^L - \mathbf{U}^{\eta,*}||_{l^2(\mathbb{R}^M)} \\
&\leq ||S\hat{\mathbf{h}}_1^{L\Delta t} - \mathbf{U}^L||_{l^2(\mathbb{R}^M)} + \frac{\varepsilon}{2} \quad (\textit{by Assumption 2}) \\
&\leq ||S\hat{\mathbf{h}}_1^{(L-1)\Delta t} - \mathbf{U}^{L-1}||_{l^2(\mathbb{R}^M)} + ||\hat{\mathcal{R}}(S\hat{\mathbf{h}}_1^{(L-1)\Delta t}, \tilde{\mathbf{G}}) - \mathcal{R}(\mathbf{U}^{L-1}, \tilde{\mathbf{G}})||_{l^2(\mathbb{R}^M)} + \frac{\varepsilon}{2} \\
&\leq ||S\hat{\mathbf{h}}_1^{(L-1)\Delta t} - \mathbf{U}^{L-1}||_{l^2(\mathbb{R}^M)} + ||\hat{\mathcal{R}}(S\hat{\mathbf{h}}_1^{(L-1)\Delta t}, \tilde{G}b) - \mathcal{R}(S\hat{\mathbf{h}}_1^{(L-1)\Delta t}, \tilde{G}b)||_{l^2(\mathbb{R}^M)} \\
&+ ||\mathcal{R}(S\hat{\mathbf{h}}_1^{(L-1)\Delta t}, \tilde{G}b) - \mathcal{R}(\mathbf{U}^{L-1}, \tilde{G}b)||_{l^2(\mathbb{R}^M)} + \frac{\varepsilon}{2} \\
&\leq (1+m)||S\hat{\mathbf{h}}_1^{(L-1)\Delta t} - \mathbf{U}^{L-1}||_{l^2(\mathbb{R}^M)} + \frac{m\varepsilon}{2(1+m)^L} + \frac{\varepsilon}{2} \quad (\textit{by Lemma 1 and Assumption 1}) \\
&\leq \frac{m\varepsilon}{2(1+m)^L}(1 + (1+m) + (1+m)^2 + \cdots + (1+m)^{L-1}) + \frac{\varepsilon}{2} \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

□