

DIMON: Learning Solution Operators of Partial Differential Equations on a Diffeomorphic Family of Domains

Minglang Yin^a, Nicolas Charon^b, Ryan Brody^{a,e}, Lu Lu^c, Natalia Trayanova^{a,d,e*}, Mauro Maggioni^{d,f*}

^a*Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD*

^b*Department of Mathematics, University of Houston, Houston, TX*

^c*Department of Statistics and Data Science, Yale University, New Haven, CT*

^d*Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD*

^e*School of Medicine, Johns Hopkins University, Baltimore, MD*

^f*Department of Mathematics, Johns Hopkins University, Baltimore, MD*

Abstract

The solution of a PDE over varying initial/boundary conditions on multiple domains is needed in a wide variety of applications, but it is computationally expensive if the solution is computed de novo whenever the initial/boundary conditions of the domain change. We introduce a general operator learning framework, called DIffomorphic Mapping Operator leaRNING (DIMON) to learn approximate PDE solutions over a family of domains $\{\Omega^\theta\}_\theta$, that learns the map from initial/boundary conditions and domain Ω_θ to the solution of the PDE, or to specified functionals thereof. DIMON is based on transporting a given problem (initial/boundary conditions and domain Ω^θ) to a problem on a reference domain Ω^0 , where training data from multiple problems is used to learn the map to the solution on Ω^0 , which is then re-mapped to the original domain Ω^θ . We consider several problems to demonstrate the performance of the framework in learning both static and time-dependent PDEs on non-rigid geometries; these include solving the Laplace equation, reaction-diffusion equations, and a multiscale PDE that characterizes the electrical propagation on the left ventricle. This work paves the way toward the fast prediction of PDE solutions on a family of domains and the application of neural operators in engineering and precision medicine.

keywords: Deep Learning, Neural Operator, Partial Differential Equations, Parametric Domains

author contributions: M.Y., N.C., M.M, and N.T. designed research; M.Y., N.C., and R.B. performed research; M.Y., L.L., and M.M. analyzed data; M.Y., N.C., L.L., and M.M. contributed to the theoretical results; M.Y., N.C., R.B., L.L., M.M., and N.T wrote the paper.

Autho Declaration: Declare no competing interests.

Corresponding authors: To whom correspondence should be addressed. E-mail: Mauro Maggioni: mauromaggionijhu@icloud.com and Natalia Trayanova: ntrayanova@jhu.edu

*Corresponding authors: ntrayanova@jhu.edu and mauromaggionijhu@icloud.com

1. Significance

Recent advancements in deep learning have been used to expedite the estimation of partial differential equation (PDE) solutions. However, there still lacks a generic framework that enables learning across a family of domains, hindering the expansion of deep learning in PDE-based engineering and medical problems. In this study, we tackle this challenge by developing a generic framework that combines diffeomorphism and deep learning for fast prediction of PDE solutions on a family of domains without retraining. The proposed framework paves the way for the utilization of deep learning for engineering or medical applications, such as design and optimization.

2. Introduction

The solution of partial differential equations (PDEs) for multiple initial and boundary conditions, and over families of domains or shapes are essential in a wide variety of disciplines, including applied mathematics, engineering, or medical sciences, with applications such as topology/design optimization and clinical prognostication being particularly demanding in terms of requiring the solution of PDEs on multiple domains and with many different initial and boundary conditions. Independent approximations of the solution in each instance using standard numerical methods, although effective, can be computationally costly, and techniques that aim at improving efficiency by exploiting relationships between solutions over different shapes and initial/boundary conditions have received significant attention in the research community.

Within the emerging field of scientific machine learning, many approaches have been proposed to approximate the solutions of PDEs by deep learning models. The idea of solving PDEs with neural networks (NN) can be traced back at least to 90s (see e.g. [13, 33]), and has received much attention in the last decade due to the advance in computational ability for training deep neural networks (DNN) (see e.g. [10, 32, 39, 51, 62]). In particular, using deep learning to learn the PDE solution operator, termed neural operator, is of particular interest as it enables approximating PDE solutions with varying boundary/initial conditions, and over parametric families of PDEs (e.g. modeling local material properties) [36, 38].

However, the need to estimating PDE solutions on various domains or shapes is yet to be fully addressed. Current frameworks for neural operators are formulated on a domain with fixed boundaries, and even slight variations in the shape of the learning domain require retraining the neural network. Previous attempts to extend the learning ability on various geometries adopted approaches based on level-sets [37], convolutional neural networks [26], among others [35], although with rather significant limitations in each of these studies. The challenge of learning operators on various geometries emanates from learning functional mappings, not on a fixed, but a family of Banach spaces over various domains, yet none of the previous studies provided a comprehensive and generic formulation that reconciles the learning problems with their supporting approximation theorem.

Herein, we propose a theoretically sound and computationally accurate operator learning framework, termed DIffomorphic Mapping Operator leaRNING (DIMON), that extends operator learning to a family

of diffeomorphic domains. DIMON combines neural operators with diffeomorphic mappings between domains/shapes, formulating the learning problem on a unified reference domain (“template”) Ω^0 , but with respect to a suitably modified PDE operator. PDE solutions along with input functions on a family of diffeomorphically equivalent shapes are each mapped onto the reference domain Ω^0 via a diffeomorphism. This map can also be interpreted as a change of variables in the PDE on the original shape, yielding a corresponding PDE on Ω^0 , parameterized by the diffeomorphism. A neural operator is trained to learn the latent solution operator of this parameterized PDE on the reference domain Ω^0 , where the parameter depends on the mapping from the original domain to the reference domain. The original operator is equivalent to the composition of the learned latent operator with the inverse mapping (pushforward). Learning this parameterized family of PDE operators using neural operators then yields a way of solving PDEs on the family of domains.

Although the DIMON framework requires no restriction on the structure of the neural operators, in order to demonstrate its utility we extend the universal approximation theorem in [29] and show that, in practice, the latent operator on Ω^0 can be approximated by adopting a multiple-input operator (MIONet). In addition, DIMON is flexible with the choice of the diffeomorphic mapping algorithms and reference domain Ω^0 , which facilitates the adoption of existing mapping algorithms under our framework [44, 58]. We demonstrate the ability of our framework with three examples, including learning static and dynamic PDE problems on both parameterized and non-parameterized synthetic domains. In particular, the last example approximates functionals of PDE solutions (in fact, PDEs couple with a rather high-dimensional ODE system) for determining electrical wave propagation in patient-specific geometries of the heart, demonstrating its utility in learning on realistic shapes and realistic dynamics in precision medicine applications.

3. Results

3.1. Problem Formulation

In this section, we present the formulation of DIMON for learning operators on a family of diffeomorphic domains, along with the unfolding of how a pedagogical example problem is set up and solved.

Setup. Consider a family $\{\Omega^\theta\}_{\theta \in \Theta}$ of bounded, open domains with Lipschitz boundaries in \mathbb{R}^d , with Θ a compact subset of \mathbb{R}^p . Assume that for each shape parameter $\theta \in \Theta$ there is a C^2 embedding φ_θ from the “template” Ω^0 to \mathbb{R}^d , which induces a C^2 diffeomorphism (i.e. a bijective C^2 map with C^2 inverse) between Ω^0 and Ω^θ . We further assume that the map $\Theta \rightarrow C^2(\Omega^0, \mathbb{R}^d)$, defined by $\theta \mapsto \varphi_\theta$ is continuous, when C^2 is endowed with the usual C^2 -norm that makes it a Banach space. Assume that for each $\theta \in \Theta$ there is a solution operator $\mathcal{G}_\theta : X_1(\Omega^\theta) \times \dots \times X_m(\Omega^\theta) \rightarrow Y(\Omega^\theta)$ mapping functions $v_1^\theta, \dots, v_m^\theta$ on Ω^θ or $\partial\Omega^\theta$, representing a set of initial/boundary conditions, to the solution $u^\theta(\cdot; \mathbf{v}^\theta)$ on Ω^θ of a fixed PDE, where $\mathbf{v}^\theta := (v_1^\theta, \dots, v_m^\theta)$.

Example: Let us consider the problem of solving the Laplace equation on a family of bounded open domains $\{\Omega^\theta\}_{\theta \in \Theta}$ of \mathbb{R}^2 with Lipschitz regular boundaries, where Θ is a compact subset of \mathbb{R}^p . We

assume that Ω^θ is, for each $\theta \in \Theta$, diffeomorphic to a fixed domain $\Omega^0 \subset \mathbb{R}^2$, which we call the reference domain. For example, if Ω^θ is a simply connected domain (different from the whole \mathbb{R}^d) for each θ , Ω^0 may be chosen to be the open unit disk $\{x \in \mathbb{R}^d : \|x\|_2 < 1\}$, and φ_θ smooth (in fact, holomorphic if \mathbb{R}^2 is identified with \mathbb{C}) bijection of Ω^θ onto Ω^0 exists by the Riemann mapping theorem.

For each $\theta \in \Theta$, we are interested in obtaining the solution of the PDE

$$\begin{cases} \Delta u = 0 & \text{in } \Omega^\theta, \\ g = v_1^\theta & \text{on } \partial\Omega^\theta. \end{cases} \quad (1)$$

for v_1^θ a function on $\partial\Omega^\theta$ with $v_1^\theta \in H^{\frac{1}{2}}(\partial\Omega^\theta)$, and with the boundary condition intended in the trace sense. In this setting, it is known that there exists a unique weak solution $u^\theta(\cdot; v_1^\theta) \in H^1(\Omega^\theta)$ to this boundary problem, c.f. for instance [22] or [28]. We are interested in learning the solution operator $\mathcal{G}_\theta(v_1^\theta) := u^\theta(\cdot; v_1^\theta)$ for this Laplace equation over the various domains Ω^θ with boundary conditions v_1^θ .

We may represent this family $\{\mathcal{G}_\theta\}_{\theta \in \Theta}$ of operators defined on functions on different domains by an operator \mathcal{F}_0 dependent on the shape parameter θ and on functions that are all defined on the reference domain Ω^0 , by suitable compositions with the diffeomorphisms φ_θ and φ_θ^{-1} , as follows:

$$u^\theta(\cdot; \mathbf{v}^\theta) = \mathcal{G}_\theta(v_1^\theta, \dots, v_m^\theta) = \mathcal{F}_0(\theta, \underbrace{v_1^\theta \circ \varphi_\theta, \dots, v_m^\theta \circ \varphi_\theta}_{\mathbf{v}^\theta \circ \varphi_\theta}) \circ \varphi_\theta^{-1} = \mathcal{F}_0(\theta, \underbrace{v_{\theta,1}^0, \dots, v_{\theta,m}^0}_{\mathbf{v}_\theta^0}) \circ \varphi_\theta^{-1} = u_\theta^0(\cdot; \mathbf{v}_\theta^0) \circ \varphi_\theta^{-1}, \quad (2)$$

where the second equality defines \mathcal{F}_0 , the third equality defines $v_k^0 := v_k^\theta \circ \varphi_\theta$, and the fourth equality shows that $u_\theta^0(\cdot; \mathbf{v}_\theta^0) := u^\theta(\cdot; \mathbf{v}^\theta) \circ \varphi_\theta$. The “latent” operator $\mathcal{F}_0 : \Theta \times X_1(\Omega^0) \times \dots \times X_m(\Omega^0) \rightarrow Y(\Omega^0)$ maps $\theta \in \Theta$ and a set of functions v_k^0 in some Banach Spaces ($X_k(\Omega^0), \|\cdot\|_{X_k(\Omega^0)}$), for $k = 1, \dots, m$, of functions on Ω^0 or $\partial\Omega^0$, to a target function u_θ^0 in a Banach Space ($Y(\Omega^0), \|\cdot\|_{Y(\Omega^0)}$) of functions over Ω^0 . As above, we used the vector notation \mathbf{v}^0 to denote (v_1^0, \dots, v_m^0) . One can view \mathcal{F}_0 as the operator that outputs the pull back of a solution of a PDE on Ω^θ to the reference domain Ω^0 via the diffeomorphism φ_θ ; this is analogous to the *Lagrangian* form of the solution operator in contrast with the *Eulerian* form \mathcal{G}_θ defined on the varying domain using the terminology of [27].

Example: Going back to the example of the Laplace equation, the operator \mathcal{F}_0 to approximate maps a shape parameter θ and boundary function v_1^θ to the (weak) solution of (1) pulled back on the template domain Ω^0 . Formally, denoting $u_\theta^0(\cdot; v_\theta^0) = \mathcal{F}_0(\theta, v_\theta^0)$, where $v_\theta^0 = v_1^\theta \circ \varphi_\theta$, $u_\theta^0(\cdot; v_\theta^0)$ can be equivalently viewed as the solution of a uniformly elliptic Dirichlet problem on Ω^0 , namely

$$\begin{cases} \varphi_\theta^* \Delta (\varphi_\theta^{-1})^* u_\theta^0(\cdot; v_\theta^0) = 0 & \text{in } \Omega^0 \\ g = v_\theta^0 & \text{on } \partial\Omega^0 \end{cases}, \quad (3)$$

in which φ_θ^* denotes the operator mapping an arbitrary function h on Ω_θ to the function $h \circ \varphi_\theta$ on Ω^0 .

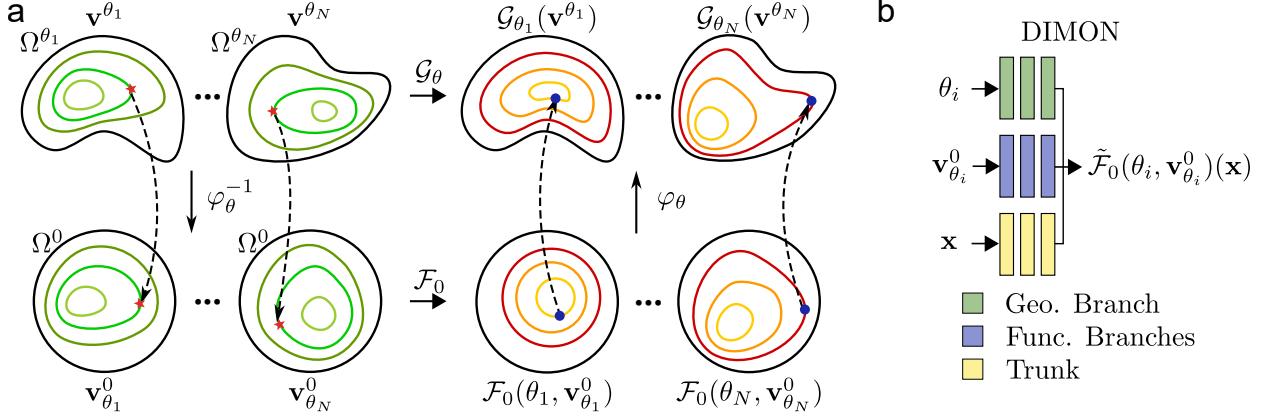


Figure 1: **Schematics of DIMON for operator learning on a family of diffeomorphic domains.** **a:** A partial differential operator \mathcal{G}_θ is posed on Ω^{θ_i} together with functions \mathbf{v}^{θ_i} ($i = 1, \dots, N$) as input. A map φ_θ , parameterized by the shape parameter θ , transports \mathbf{v}_θ^0 posed on reference domain Ω^0 onto Ω^θ and identifies point correspondences across all Ω^θ and Ω^0 (indicated by red marks and blue dots). There exists a latent operator \mathcal{F}_0 such that $\mathcal{G}_\theta(\mathbf{v}^\theta) = \mathcal{F}_0(\theta, \mathbf{v}_\theta^0) \circ \varphi_\theta^{-1}$. **b:** The operator \mathcal{F}_0 is approximated by a neural operator $\tilde{\mathcal{F}}_0$: a shape parameter θ (typically estimated from data) and the transported function \mathbf{v}_θ^0 are the inputs of branch networks; the coordinates of the reference domain x are the inputs of the trunk network.

Note that the specific coefficients of this linear second-order elliptic operator $L_\theta = \varphi_\theta^* \Delta (\varphi_\theta^{-1})^*$ could be written explicitly based on the first and second order derivatives of φ_θ , and are in particular continuous functions on Ω^0 , c.f. Appendix A for more details.

Learning problem. Given N observations $\{(\Omega^{\theta_i}, v_1^{\theta_i}, \dots, v_m^{\theta_i}, \mathcal{G}_{\theta_i}(v_1^{\theta_i}, \dots, v_m^{\theta_i}))\}_{i=1}^N$ and a diffeomorphism φ_θ , where θ_i and $v_k^{\theta_i}$ are all independently sampled at random from a probability measure on Θ and on compact subsets of X_k , respectively, we aim at constructing an estimator of \mathcal{F}_0 defined in (2).

We choose the estimator in the form of a neural operator $\tilde{\mathcal{F}}_0$, which takes as inputs the shape parameter θ , the transported input functions \mathbf{v}_θ^0 , and outputs an approximation to $\mathcal{F}_0(\theta, \mathbf{v}_\theta^0)$. We visualize the framework in Fig. 1a and the high-level architecture of the neural operator in Fig. 1b; we discuss more details of the neural operator and its approximation ability in Sec. 3.2.

The framework discussed above extends immediately to at least the following settings:

- time-dependent operators, by concatenating time t to x (see Example 2);
- operators that involve vector- or tensor-valued functions, by replacing the pullback operation on scalar functions with an appropriate action of diffeomorphisms on the vector and tensor fields. For example, the action of φ_θ on a vector field $v : \Omega^0 \rightarrow \mathbb{R}^d$ is the vector field defined for all $x_\theta = \varphi_\theta(x) \in \Omega^\theta$ by: $J\varphi_\theta \cdot v(x)$ where $J\varphi_\theta$ denotes the Jacobian matrix of the transformation. This action can be further extended to covariant/contravariant tensor fields of any type.

3.2. Universal Approximation Theorem on a Diffeomorphic Family of Domains

In this section, we extend the applicability of the universal approximation theorem (UAT) presented in Jin et al. [29] on learning parametric domains for multi-input operators under DIMON framework. In

short, the UAT in [29] guarantees the approximation ability of a multi-branch DeepONet for nonlinear and continuous operators with the assumption of fixed domains. Transporting functions defined on various domains via φ_θ onto a reference domain with a fixed boundary naturally aligns well with the assumption of UAT, hence enabling the ability to learn on various domains. To differentiate the mapping φ and the shape of Ω^θ , we define a shape parameter $\theta \in \Theta \subset X_0$ in a Banach space X_0 and introduce the branch for encoding geometric information, referred to as Geo. branch. Therefore, the adapted UAT can be written as:

Theorem 1 (Adapted from [29]). *Assume that X_0, X_1, \dots, X_m, Y are Banach Spaces, $\Theta \subset X_0, K_i \subset X_i$ ($i = 1, \dots, m$) are compact sets, and X_i ($i = 0, \dots, m$) has a Schauder basis. If $\mathcal{F}_0 : \Theta \times K_1 \times \dots \times K_m \rightarrow Y$ is a continuous operator, then for any $\epsilon > 0$, there exists a neural network $\tilde{\mathcal{F}}_0$ with the form given in Eq. (8), such that*

$$\left\| \mathcal{F}_0 - \tilde{\mathcal{F}}_0 \right\|_{C(\Theta \times K_1 \times \dots \times K_m, Y)} < \epsilon,$$

where the norm C is the sup norm.

Despite its generality, the use of Theorem 1 for the approximation of solution operators of PDEs typically requires extra analysis, in particular to ensure that the assumptions on \mathcal{F}_0 are satisfied, namely the well-posedness and continuity with respect to the domain geometry as well as initial/boundary conditions: this can be non-trivial for many PDEs. While it is out of the scope of the present paper to discuss this in a systematic fashion, we will detail below and in the appendix the particular case of the Laplace equation introduced above, under suitable regularity assumptions.

Example: Going back to the example of the Laplace equation of Sec. 3.1, the key issue to apply Theorem 1 in this case is to determine adequate functional spaces for the solution and boundary function under which $u_\theta^0(\cdot; v_\theta^0)$ is first well-defined but also continuous with respect to v_θ^0 and θ . The existence and uniqueness of weak solutions in $H^1(\Omega^0)$ for smooth domains and $v_\theta^0 \in H^{1/2}(\partial\Omega^0)$ follow from classical results on elliptic PDEs (see e.g. [22] Chap.8), and have been extended to Lipschitz regular domains in [28]. Regularity estimates for this class of PDEs also guarantee the continuity of the solution with respect to the boundary function v_θ^0 . However, to the authors' knowledge, regularity with respect to domain deformations (and thus θ) is much less known in such a general setting. Based on the continuity of the mapping $\theta \mapsto \varphi_\theta$ from Θ to $\mathcal{C}^2(\Omega^0, \mathbb{R}^d)$ and by adapting the ideas presented in [27], one can recover the solution's continuity with respect to θ albeit under the stronger assumption that Ω^0 has a C^2 boundary. For the sake of concision, we refer to Appendix A for some additional details. With such assumptions, the conclusion of Theorem 1 holds and it is thus possible to approximate the solution operator up to any precision using the operator learning framework developed in this paper.

3.3. Algorithms and practical considerations

In practice, the domain Ω^θ may be available only in discretized form, and numerical methods are used to approximate the solution of a PDE on the domain of Ω^θ . For all the examples presented in this work,

we adopt a FEM with a piece-wise linear basis to generate a training set of PDE solutions, consistent with the assumption $u^\theta \in H^1(\Omega^\theta)$ on the weak solution. One could adopt numerical methods with higher-order polynomials, such as spectral element methods [31, 47] or hp -FEM [3], in order to approximate smoother PDE solutions with higher accuracy; note that in this case the differentiability of φ_θ needs to be sufficiently high to preserve the smoothness solution after composition.

3.3.1. Choosing the reference domain Ω^0

In general, there is no canonical choice for Ω^0 , or for the diffeomorphisms between Ω^0 and Ω^θ . Often the domains are unparameterized, given for instance meshes with no explicit point correspondences between them. For such unparameterized domains, a practical choice is then to set a reference domain that has a simple shape and is diffeomorphically equivalent to all the Ω^θ 's. For instance, we choose a hollow semi-sphere as Ω^0 as a simple shape visually close to the left ventricle in Example 3. In general, one may select one particular Ω^{θ_i} , with $\theta_i \in \Theta$, as reference, or estimate a template shape, constrained to lie in $\{\Omega^\theta\}$, from a given dataset of sample domains using one of the many possible approaches developed for this purpose including, among others, [1, 15, 25, 30, 41]. Although one can measure the similarity/distance between domains using a metric [42], i.e., Chamfer distance [20, 46], Earth Movers' distance [52, 53], or Riemannian metrics on shape spaces [4, 7, 23], this is not a prerequisite for successfully adopting this learning framework. This flexibility in choosing the reference domain facilitates a wide set of applications.

3.3.2. Estimating and encoding the domain geometry with the shape parameter

Once a reference domain Ω^0 is chosen, the diffeomorphisms φ_θ that map Ω^0 to the domains Ω^θ need to be constructed. In the case of a family of parameterized shapes, as in the examples of Section 3.5 below, the φ_θ can be constructed naturally by composing parameterizations, which also yields point correspondences, see Sec. 5.3.

For unparameterized data however, such as the meshes in the example of Section 3.6 and 3.7, point correspondences are not a priori available and the mesh structure itself may not even be consistent from one data sample to another. In such cases, the maps φ_θ are not known and need to be approximated for each domain. A variety of models can be used for this purpose relying, for example, on maps that are area-preserving [17], quasi-conformal [14, 50] or quasi-isometric [55], and which can be estimated for instance via the versatile functional maps framework of [44, 45]. Yet, in some applications, one may need a larger class of possible deformations, especially if larger distortions occur between different domains. For that reason, in the present work, we choose to rather rely on the diffeomorphic registration setting introduced in [7], known as Large Deformation Diffeomorphic Metric Mapping (LDDMM). This model has several notable advantages when it comes to the applications we consider here. First, it generates transformations as flows of smooth vector fields that result in smooth diffeomorphisms of the whole ambient space \mathbb{R}^d , which is consistent with the theoretical setting developed in the previous section. From these estimated deformations, one can easily recover approximate point correspondences between Ω^0 and the other domains, as well as map functions but

also vector fields or tensor fields from one domain to another. Another interesting feature of LDDMM is the fact that it is connected to a Riemannian metric on the diffeomorphism group [61]. Thus, any optimal deformation path in that model can be interpreted as a geodesic and, as such, be represented compactly via some initial "momentum" field defined on Ω^0 . The Riemannian setting further enables the extension of many statistical techniques to the space of deformations, including methods such as principle component analysis (PCA) for dimensionality reduction, as we discuss next.

In the perspective of learning the solution operator over a family of diffeomorphic domains, we also seek a representation of the deformations φ_θ that can be effectively used as input for the operator learning neural network. The estimator of the map φ_θ generally leads to high-dimensional objects, either a large set of point correspondences between the mesh of Ω^0 and Ω^θ or as a set of momentum vectors attached to vertices of Ω^0 in the case of the LDDMM framework described above. In order to obtain an informative representation encoded by lower dimensional shape parameter $\tilde{\theta}$, we apply dimensionality reduction (herein PCA, see Sec. 5.2 for additional details). A first approach is to apply standard PCA to the set of displacement vectors $\{x - \varphi_\theta(x)\}_{x \in \text{Indmrk}(\Omega^0)}$, where $\text{Indmrk}(\Omega^0)$ is a fixed set of landmark points in Ω^0 . Alternatively, when deformations are estimated from the LDDMM model, we instead leverage the geodesic PCA method [21] on either the momentum or the deformation representation. In either case, projection onto the first p' principal components leads to our estimated low-dimensional representation $\tilde{\theta}$ of the shape parameter θ for each Ω^θ , and gives a more compact representation of φ_θ . Notice that we only adopt reduced-order PCA coefficients to represent the shape of Ω^θ for network training, but will still rely on the originally (non-reduced) estimated mappings φ_θ to transport the estimated solution on Ω^0 back to Ω^θ .

3.4. Algorithms

We summarize our procedure in Algorithm 1. The computational complexity is dictated by the complexity of solving the PDE on Ω^θ and the training for the neural network operator.

Algorithm 1 The algorithm of learning solution operators on parametric domains.

Data: $\{(\Omega^{\theta_i}, \mathbf{v}^{\theta_i}, u^{\theta_i}(\cdot; \mathbf{v}^{\theta_i}))\}_{i=1}^N$: numerical PDE solution $u^{\theta_i}(\cdot; \mathbf{v}^{\theta_i})$, with input functions \mathbf{v}^{θ_i} , on Ω^{θ_i}

1. set a reference domain Ω^0 and compute point correspondences between Ω^{θ_i} and Ω^0
2. calculate the reduced-order shape parameters $\tilde{\theta}_i$, using for example PCA
3. pushforward \mathbf{v}^{θ_i} and u^{θ_i} onto Ω^0 : $u_{\theta_i}^0 := u^{\theta_i} \circ \varphi_{\theta_i}$ and $\mathbf{v}_{\theta_i}^0 := \mathbf{v}^{\theta_i} \circ \varphi_{\theta_i}$
4. initialize and train a neural operator $\tilde{\mathcal{F}}_0$ to approximate $u_{\theta_i}^0$:

for epoch = 1: N_{train} **do**

- forward propagation $u_{NN,i} := \tilde{\mathcal{F}}_0(\tilde{\theta}, \mathbf{v}_{\theta_i}^0)$ for all θ_i in the training dataset
- compute the mean square error (MSE) as the loss $\mathcal{L} := \sum_i \text{MSE}(u_{NN,i}, u_{\theta_i}^0)$
- update network parameters to minimize \mathcal{L}

end for

5. pull network predictions back to Ω^{θ_i} using $u_{NN,i} \circ \varphi_{\theta_i}^{-1}$.

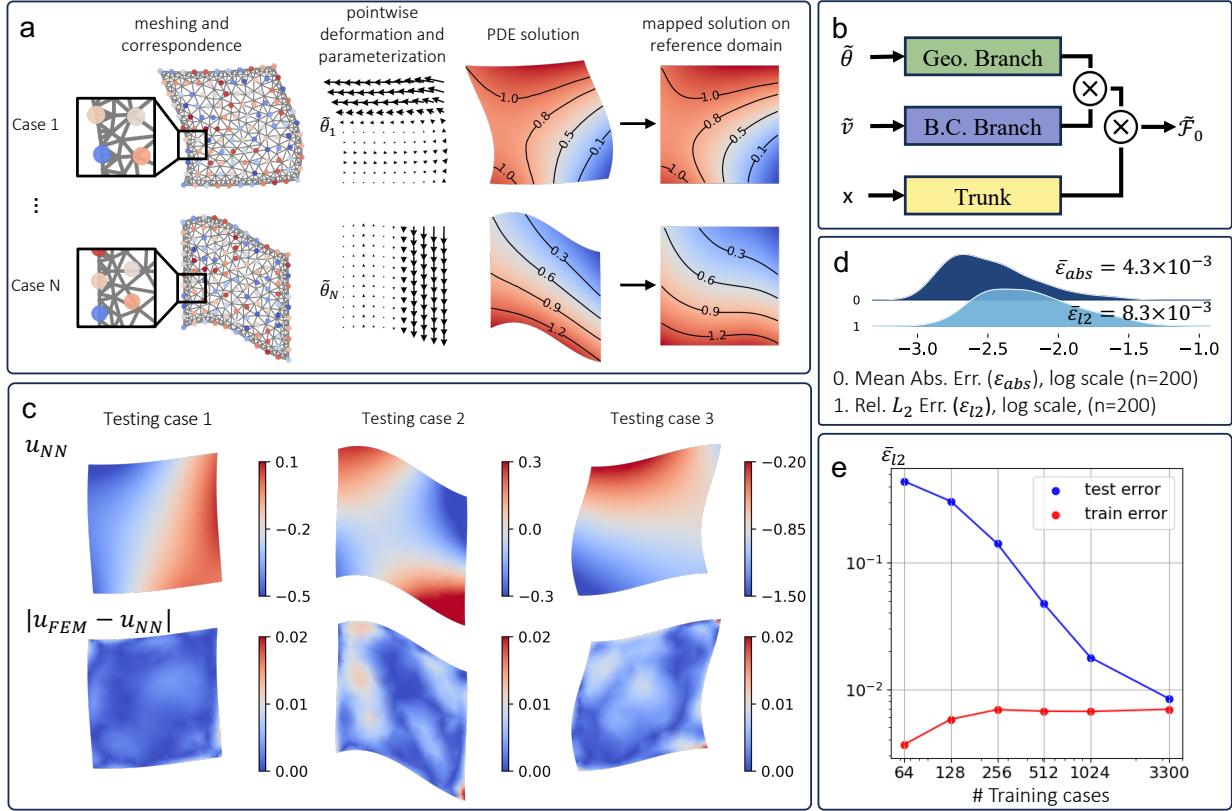


Figure 2: **DIMON for learning solution operator of the Laplace equation on parameterized 2D domains.** **a:** Data generation on N parametric 2D domains. Landmark points are matched across 2D domains with different meshing. Shape parameter $\tilde{\theta} \in \mathbb{R}^{p'}$ is calculated by PCA on the deformation field at the landmark points. Here, we adopt the first 15 principal components as $\tilde{\theta}$. The PDE is solved using finite elements and mapped onto the reference domain. **b:** Neural operator structure. Shape parameter $\tilde{\theta}$ and boundary condition \tilde{v} are fed into the geometry (Geo.) and boundary condition (B.C.) branch, respectively. Coordinates in the reference domain \mathbf{x} are the input of the trunk net. **c:** Network predictions (u_{NN}) and their absolute errors with numerical solutions ($|u_{FEM} - u_{NN}|$) on three testing domains. **d:** Distributions of mean absolute errors ϵ_{abs} and relative L_2 error (ϵ_{l2}) on the testing dataset in log scale. The testing dataset contains $n = 200$ cases. **e:** The average of ϵ_{l2} ($\bar{\epsilon}_{l2}$) for the training (red) and testing dataset (blue) is plotted against the number of training cases.

3.5. Example 1: Solving the Laplace Equation on Parametric 2D Domains

PDE problem. In this section, we present the network prediction for the pedagogical example of the Laplace equation discussed above. The PDE problem on a domain Ω^θ is

$$\begin{cases} \Delta u &= 0, \quad \text{in } \Omega^\theta, \\ u|_{\partial\Omega^\theta} &= v \sim \text{GRF}(0, k_l), \end{cases}$$

where for the boundary conditions we use the Gaussian Random Field (GRF) with the covariance kernel:

$$k_l(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/2l^2), \quad (4)$$

Data preparation. The process of data preparation is illustrated in Fig. 2a. Following the steps detailed in Method and Materials 5.3, we first generate 3,500 samples of the shape parameters θ , create the corresponding

domains Ω^{θ_i} , which are deformations of the unit square, which itself is set as the reference domain Ω^0 . We then impose a boundary condition sampled from a GRF with a length scale $l = 0.5$ and solve the governing equation using a finite element solver. At the end of data preparation, we map the PDE solution from Ω^{θ_i} to Ω^0 for network training.

Construction of φ_θ and $\tilde{\theta}$. We obtain a set of point correspondences between 11×11 uniformly distributed landmarks on Ω^0 to their images in Ω^{θ_i} , using the map φ_{θ_i} , which is considered known in this example (see Sec. 5.3). We visualize the deformation via vector fields in the second column of Fig. 2a, where the length of the arrows corresponds to the magnitude of the deformation. Next, with the identified point correspondence and deformation field, each domain Ω^{θ_i} can be parameterized using truncating principal component coefficients in PCA, yielding the low-dimensional shape parameter $\tilde{\theta}$. Although the diffeomorphisms in this example are explicitly known, we intentionally use PCA to parameterize the shape to demonstrate the generalization of our framework when the diffeomorphisms are unknown. A comparison of network performance using θ and $\tilde{\theta}$ can be found in Sec. Appendix B.

Network training and performance. Fig. 2b illustrates the network architecture for approximating the latent operator on the reference domain for this example. Since the new operator is a function of shape parameters and boundary conditions, the network has two branch networks for inputting the truncated shape parameter $\tilde{\theta}$ and boundary values represented by a set of discrete points, denoted as \tilde{v} . The trunk network takes the coordinates on the reference shape \mathbf{x} as input. We train the network on $n = 3,300$ training cases (see Appendix D for details) and use the remaining $n = 200$ cases for testing. Network predictions in Ω^0 are then mapped back to the original domain via φ_θ for further visualization and error analysis. Fig. 2c displays network predictions u_{NN} and the absolute errors $|u_{FEM} - u_{NN}|$ on three representative domains in the testing cases, indicating that network predictions are in good agreement with the numerical solutions. The maximum value of the absolute error is less than $2e - 2$. We summarize the statistics of the mean absolute error ε_{abs} and relative L_2 error $\varepsilon_{l2} := \|u_{FEM} - u_{NN}\|_2 / \|u_{FEM}\|_2$ in the testing cases by presenting their distributions in Fig. 2d. Both distributions are left-skewed when viewed in a logarithmic scale, with the averaged ε_{abs} and the mean relative L^2 error at 4.3×10^{-3} and 8.3×10^{-3} , respectively. The behavior of train and test error as a function of the number of training cases is represented in Fig. 2e. $\bar{\varepsilon}_{l2}$ of the testing dataset decreases to the same order of training error with an increasing number of training cases, from $\mathbf{O}(10^0)$ to $\mathbf{O}(10^2)$. Based on the above results, we conclude that our framework enables effective learning of the solution operator for the Laplace equation on a diffeomorphic family of domains.

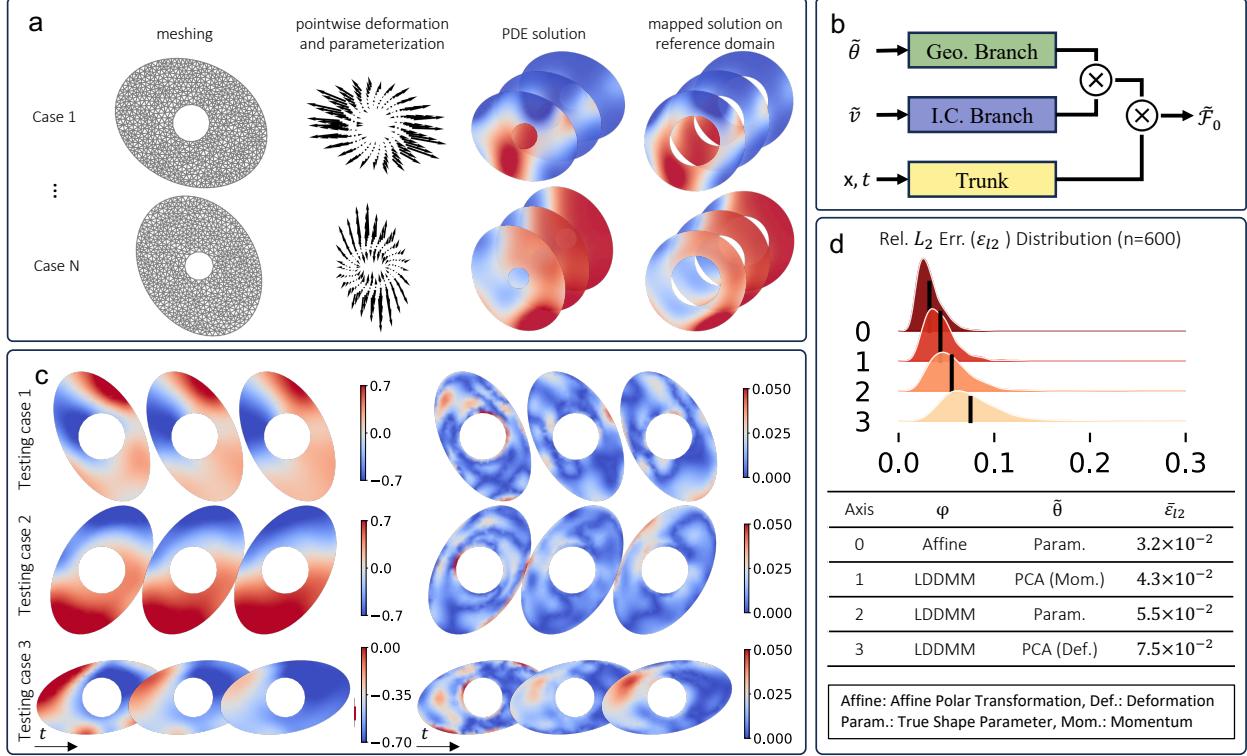


Figure 3: **DIMON for learning solution operator of the Reaction-Diffusion (RD) equation on parametric 2D domains.** **a:** Data preparation on N parametric domains. Each domain is parameterized by either the deformation or the momentum field at the landmark points with p' principal component coefficients. A finite element method is adopted to numerically solve the RD equation with various initial conditions (I.C.) on Ω^{θ_i} , set as an annulus, and map the solution to Ω^0 at different time snapshots. **b:** Neural operator structure. Truncated shape parameter $\tilde{\theta}$ and boundary condition \tilde{v} are fed into the geometry (Geo.) and boundary condition (B.C.) branch, respectively. Coordinates in the reference domain \mathbf{x} are the input of the trunk net. **c:** Network predictions with true shape parameters of the affine transformation at $t = 0, 1$, and 2s and their absolute errors on three testing domains. **d:** Average relative L_2 error ($\bar{\epsilon}_{l2}$) on the testing cases based on $\tilde{\theta}$ obtained from 4 approaches: affine with true shape parameters, LDDMM with PCA on momentum and deformation, and LDDMM with true shape parameters. The averaged $\bar{\epsilon}_{l2}$ for different approaches range from 3.2×10^{-2} to 7.5×10^{-2} , indicating robustness of DIMON with respect to the way the diffeomorphisms are learned and approximated. The number of testing cases is $n = 600$.

3.6. Example 2: Learning Reaction-Diffusion Dynamics on Parametric 2D Domains

PDE problem. We now consider a more challenging example, with a PDE that includes time and is nonlinear, namely a reaction-diffusion in the form

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u + R(u), \quad \text{with } R(u) = u(1 - u^2), \quad \text{in } \Omega^\theta \\ u|_{t=0} &= v, \quad \left. \frac{\partial u}{\partial n} \right|_{\partial \Omega^\theta} = 0. \end{aligned}$$

where the Ω^θ 's are elongated ellipses with a central hole, and Ω^0 is a circular annulus. The reaction term $R(u)$ is adopted from the Newell–Whitehead–Segel model [43] to describe post-critical Rayleigh–Bénard convection. We impose a Neumann boundary condition and impose initial conditions by sampling from a Gaussian Random Field (GRF).

Data preparation. We visualize the data generation and learning process in Fig. 3a: we generate $N = 6,000$ sets of shape parameters θ following the steps in Method and Materials 5.3. Next, we simulate the system dynamics from $t = 0$ to $t = 2$ with a finite element method and map the numerical solution at different snapshots to the reference domain.

Construction of φ_θ and $\tilde{\theta}$. To demonstrate the compatibility of our framework with generic methods of domain mapping, we adopt two approaches for acquiring point correspondences between Ω^{θ_i} and Ω^0 (Fig.3a), namely, affine polar transformation (referred to as “affine”) and LDDMM with details provided in Sec.5.1. The former approach converts the Cartesian coordinates of Ω^{θ_i} into polar coordinates with the radius normalized to $[0.5, 1.0]$. However, affine transformation in the example is less general than LDDMM as it depends on the particular shape of Ω^0 and Ω^{θ_i} . With the identified point correspondence, we proceed to calculate the truncated shape parameters $\tilde{\theta}$ using PCA based on 1) deformation from LDDMM and 2) momentum from LDDMM.

Network training and performance. Fig. 3b shows the architecture of the network. The network contains a branch for encoding the truncated shape parameter $\tilde{\theta}$ and one for the initial conditions f . \mathbf{x} and t are the coordinates in Ω^0 and time. We train a network with 5,400 training cases and examine its performance with 600 testing cases. Fig. 3c displays the predicted solutions and the absolute errors for three representative testing cases at $t = 0, 1$, and 2 s. $\tilde{\theta}$ are adopted as the true shape parameters. Qualitatively, the network predictions closely match the finite element solutions with low absolute errors. Fig. 3d illustrates the relative L_2 error distribution calculated over the 600 testing cases from various point correspondence algorithms, including 0) Affine with true parameter 1-3) LDDMM with PCA on momentum, true parameter, and PCA on deformation, where the true parameters refers to the shape parameters in Sec. 5.3. All of the distributions exhibit left skewness with means (black line) ranging from 3.2×10^{-2} to 7.5×10^{-2} . Mean errors from affine transformation are smaller than those of LDDMM due to the more uniform distribution of landmarks in Ω^{θ_i} and no matching error between Ω^0 and Ω^{θ_i} . A comparison of affine transformation and LDDMM is presented in Appendix C. Nevertheless, the above results indicate that DIMON can accurately learn the

reaction-diffusion dynamics on the diffeomorphic family of domains with a flexible choice of point-matching algorithms.

3.7. Example 3: Predicting patient-specific electrical wave propagation in the left ventricle

In this section, we demonstrate the capabilities of our framework on predicting electrical signal propagation on the left ventricles (LVs) reconstructed for clinical imaging scans of patients.

PDE problem. Human ventricular electrophysiology is described by the following mono-domain reaction-diffusion equation:

$$C_m u_t = -(I_{ion} + I_{stim}) + \nabla D \nabla u, \quad (5)$$

where C_m is the membrane capacitance, u is the transmembrane potential, D is a conductivity tensor controlled by the local fiber orientation, I_{stim} is the transmembrane current from external stimulation, and I_{ion} is the sum of all ionic current. A comprehensive description of I_{stim} , I_{ion} , model parameters, and the underlying ordinary differential equations can be found in [56, 57]. It is important to note that the system is essentially anisotropic and multiscale.

Data preparation. Fig. 4a provides an overview of training data generation. We collected 1,007 anonymized cardiac magnetic resonance (CMR) scans from patients at Johns Hopkins Hospital. However, our goal in this work is not to predict the full solution of this multiscale system at all times; we focus on two clinically relevant quantities, the distributions of activation times (ATs) and repolarization times (RTs) of a propagating electrical signal, defined as the time when local myocardium tissue is depolarized (activated) and repolarized. For each LV, we apply an external electrical stimulation (pacing) at 7 myocardium points (Fig. 4b) in each segment and record the local ATs and RTs for each elicited wave (visualized in Fig. 4a) for network training and testing. We provide more details of data generation in Methods and Materials 5.5.

Construction of φ_θ and $\tilde{\theta}$. We choose a hollow semi-sphere with an inner radius of 0.5 and an outer radius of 1 as Ω^0 for LVs as they are diffeomorphically equivalent. To calculate point correspondence between these non-rigid and unparameterized LV meshes with Ω^0 , we adopt a registration method by matching the universal ventricular coordinates (UVCs) [5, 6]. As depicted in Fig. 4d, UVC converts Cartesian coordinates of a LV into three new coordinates in fixed ranges: apicobasal ($\rho \in [0, 1]$), rotational ($\phi \in [0, 2\pi]$), and transmural ($\delta \in [0, 1]$). Here, ρ and δ represent a normalized “distance” to the basal and epicardial surfaces, while ϕ represents a rotational angle with respect to the right ventricle center. We calculate the UVCs for each LV in the cohort and establish their correspondence to Ω^0 by matching their UVCs. Once the point correspondence is attained, we perform PCA on the coordinate difference at landmark points to derive $\tilde{\theta}$, a low-dimensional representation of the shape parameter θ .

Notably, we assume the fiber field depends solely on the shape of the LV [6], thereby encoding the LV shape with $\tilde{\theta}$ simultaneously leads to encoding the fiber field. To illustrate how the fiber fields influence electrical propagation, we present three exemplary cases with ATs on the reference domain in Sec. Appendix E, each with the same pacing location.

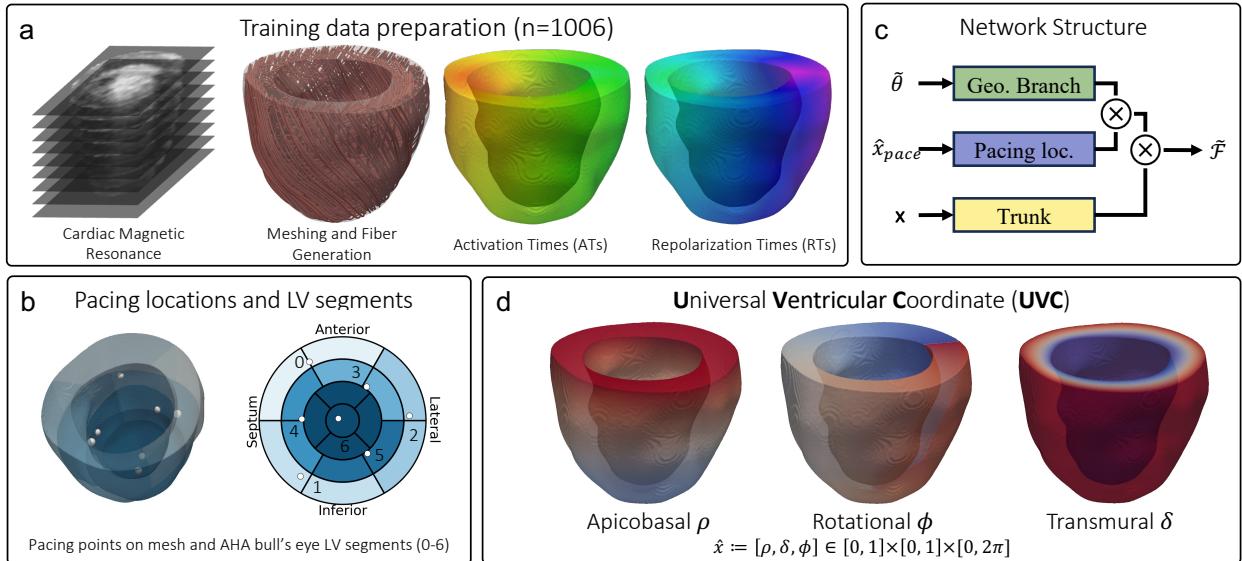


Figure 4: **Predicting electrical signal propagation on patient-specific left ventricles (LV).** a: Data preparation for the network training. The cohort assembled consists of 1007 cardiac magnetic resonance (CMR) scans of patients from Johns Hopkins Hospital, based on which LV meshes and fiber fields are generated. Next, we simulate a paced beat for 600 ms at an endocardial site until the LV is fully repolarized and record the activation times (ATs) and repolarization times (RTs) for network training. b: Pacing locations (marked in white balls) located on the endocardium with a reduced American Heart Association (AHA) LV segments from zone 0 to 6. c: Network structure. c: Universal ventricular coordinate (UVC) calculation for registering points across geometries.

Network training and performance. We study the neural network performance in predicting ATs and RTs on patient-specific LVs. The network architecture is presented in Fig. 4b. We input shape parameters $\tilde{\theta}$ along with pacing locations \hat{x}_{pace} into two separate branches. The trunk network takes UVCs (\mathbf{x}) of each training point as input. We present the network prediction in the top panel of Fig. 5. After random shuffling, we split the training dataset ($n=1007$) into 10 subsets with 100 geometries in sets 1-9 and 107 geometries in set 10. First, we train a network using geometries and ATs/RTs in sets 1-9 and set 10 for testing. We visualize predicted ATs and RTs along with their absolute errors for three representative LVs from the testing cohort. We observe that ATs and RTs on each LV with various pacing locations are faithfully predicted by our framework, with the maximum absolute error of ATs and RTs being around 20 ms (purple areas in the “Abs. Error” columns in Fig.5). This error is relatively small compared to the maximum values of ATs and RTs.

To quantitatively investigate how well the network predicts for the testing cases, we present the distribution of relative L_2 and absolute error in the bottom panel. Each row represents a different myocardial segment as defined in Fig. 4b. The error distributions are roughly centered around their respective means (black lines) with left skewness. However, for apical regions (segment 6), both errors are slightly larger than the rest of the regions, which could be attributed to the complex fiber architecture created by rule-based methods at the apex [19]. More specifically, we found that for the three cases with the largest error, the reconstructed LV shape unfaithfully reflects the true shape of LV due to the collective artifacts from data acquisition and imaging segmentation. We discuss this further in Sec. Appendix F.

Furthermore, we perform 10-fold cross-validation to quantify the generalization ability of the network, e.g., adopting sets 2-10 for training and set 1 for testing. The bottom right panel shows the relative and absolute errors from 10-fold cross-validation along with their means (0.018 and 0.045) and standard deviations. In general, relative errors of ATs from 10 iterations are larger than the RTs errors, though the magnitude of errors in ATs is smaller than the RTs.

Finally, we compare the computational cost between our neural operator approach and a traditional finite elements solver (openCARP, used widely by the cardiac computational modeling community) in Table 1. During the testing phase, our neural operator significantly accelerates the overall speed: it only requires less than 1 s on a single GPU to predict the ATs and RTs for a new patient, whereas it takes our solver openCARP 5 hours on 48 CPU nodes to simulate ATs and RTs in a single beat, not accounting for the additional time spent preprocessing and meshing. In conclusion, our framework demonstrates that a neural operator can efficiently predict patient-specific electric propagation on realistic heart geometries with satisfactory accuracy and substantial speedup in computing.

4. Discussion

We introduced the DIMON framework for learning solution operators of PDEs on a family of diffeomorphic domains using neural networks. DIMON combines diffeomorphic maps between domains with a neural

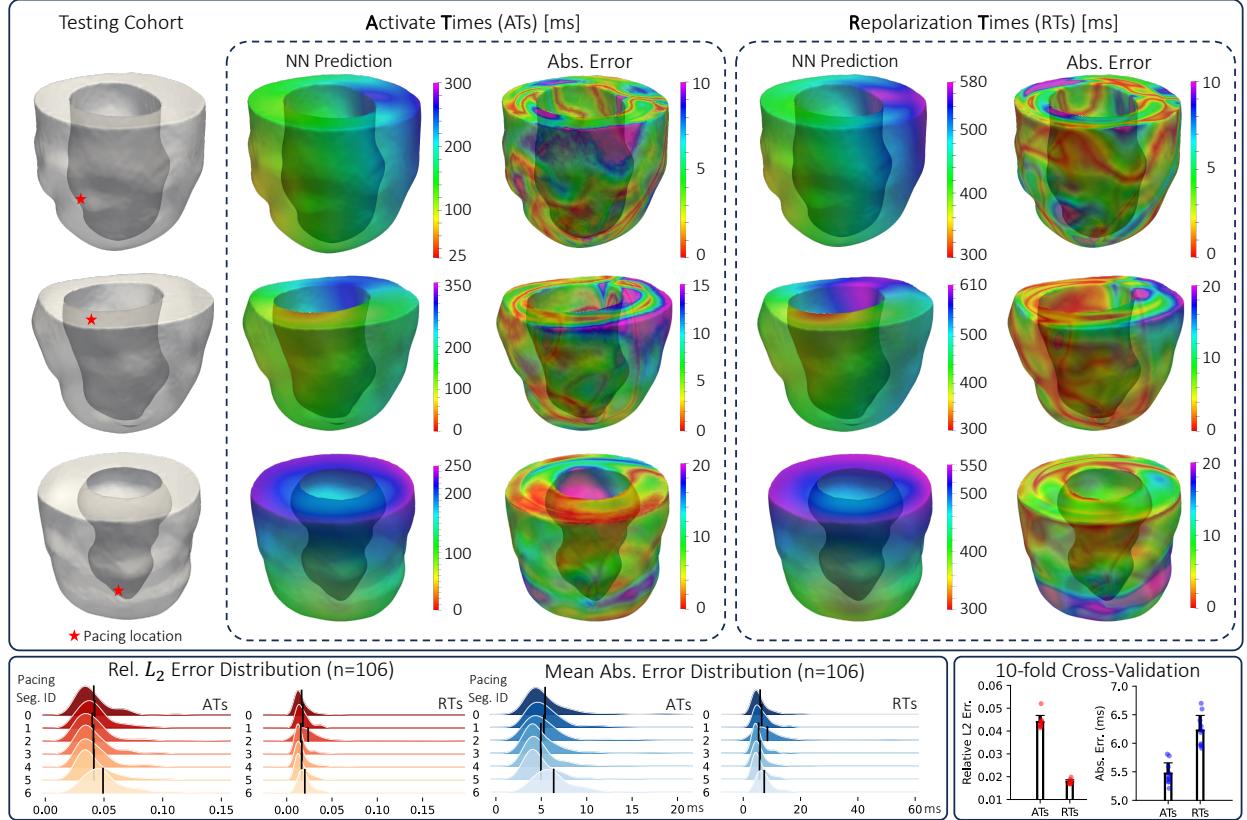


Figure 5: **Network prediction of ATs and RTs from the testing cohort.** Top panel: LV shape with their predicted ATs and RTs and absolute errors. Red stars are the pacing locations. The colorbar is selected to be consistent with the CARTO clinical mapping system, with red denoting early activation and progressing to purple for late activation. Bottom left panel: Relative and absolute error distributions for ATs and RTs. Bottom right panel: Bar plots of the average relative L_2 errors from 10-fold cross-validation.

	Wall time	Hardware
Neural Operator	<1s	1 GPU
openCARP (FEM)	5 hours	48 CPU cores

Table 1: Computational costs of neural operator and openCARP for prediction.

operator that learns a latent solution operator of PDEs, parameterized by the domain shape together with boundary and initial conditions. We first described the problem formulation, along with the unfolding of the solution of a pedagogical example. We then presented an extension of the approximation theorem for a neural operator to the setting introduced here that includes a diffeomorphic family of domains. Next, we discussed the overall workflow of DIMON, together with shape-matching algorithms for estimating in practice the diffeomorphisms between the shapes a reference shape. Finally, the performance of DIMON on operator learning is demonstrated by three representative examples, including solving static and dynamic problems on synthetic parameterized 2D domains or on realistic 3D geometries, including a data set of interest in cardiac electrophysiology and precision medicine. Networks were trained with various shape-matching algorithms and a compressed, low-dimensional shape parameter $\tilde{\theta}$, constructed using a variety of methods including LDDMM, affine transformation, or UVC coordinates, combined with the true shape parameter or PCA reduction on deformation/momentum fields. These examples demonstrate the capability of DIMON for efficient and accurate predictions of PDE solutions or functionals thereof on a variety of geometries.

DIMON is novel as it is the first systematic work that extends the learning ability of neural operators to PDEs on multiple geometries. We remove the restriction of learning on a fixed domain in [29, 38] by introducing a generic diffeomorphic mapping to the framework and transporting functions in Ω^θ to a fixed domain Ω^0 . In addition, we also show that, by adopting neural operator architecture in [29] and introducing the shape space Θ , DIMON is compatible with the approximation theorem described in [29], which provides support for the theoretical development and practical implementation of our framework.

The framework is quite general in many aspects. For example, although we chose MIONet as the baseline neural operator model in this work, DIMON is agnostic to the choice of neural operators, allowing to incorporate a variety of models, including [11, 24, 36, 40]. We also demonstrate, in Example 3, that the learning ability of DIMON reaches beyond solution operators of PDEs. The ATs/RTs are the times when the PDE solution, transmembrane potential, exceeds or decreases below certain thresholds, and therefore functionals of solutions of the PDE, rather than solutions themselves. Finally, in the direction of design and optimization, DIMON allows for the computation of the derivative of the cost functional with respect to the parameters that describe the domain shapes [54] via automatic differentiation. This, in turn, simplifies the calculation of admissible deformations in shape optimization. This framework conveniently allows imposing boundary conditions, or initial conditions, or local material properties on different geometries. There is also significant flexibility in selecting the diffeomorphic reference domain, and the algorithm to register Ω^θ to Ω^0 .

Our framework is significant to many downstream applications. One concrete example arises from cardiology. For instance, in cardiac electrophysiology, a novel approach to examine the arrhythmia inducibility of the substrate is to perform simulations based on the patient’s heart shape acquired from high-quality imaging scans. Though effective [49], the excessively high computational cost poses a significant challenge to its vast application and implementation in clinics. With our framework, one could train a neural operator that predicts electrical propagation on various hearts without the need to perform expensive numerical sim-

ulations. Our framework is also promising for topology optimization, where shapes are optimized in order to meet certain design requirements, and PDEs need to solved repeatedly on a large number of domains during such optimization procedure.

DIMON also has several limitations. We only consider mapping scalar functions in this work, whereas mapping vector or tensor fields is yet to be addressed given that it holds particular importance in representing stress and velocity in fluid and solid mechanics. Incorporating methods that transport vector/tensor fields across shapes or manifolds [2, 7, 18] can potentially address this issue and will be of interest in the future. Also, although PCA is indeed a general approach for parameterizing shapes irrespective of their topology, other local methods for shape parameterization, such as non-uniform rational B-spline (NURBS), can be adopted due to their explicit form of splines or polynomials [9, 59]. In addition, conducting a rigorous analysis of the network’s convergence rate is beyond the scope of this study, making it difficult to estimate a priori how large of a training set is needed in order to achieve a given accuracy of the solutions.

5. Material and Methods

5.1. Large Deformation Diffeomorphism Metric Mapping (LDDMM)

We give a brief summary of the LDDMM framework that is used specifically to estimate diffeomorphic mappings between the domains of Example 2, but that could more generally apply to a variety of different situations and geometric objects. The interested reader can further refer to [7] and [61] for additional details on the model.

The central principle of LDDMM is to introduce diffeomorphic transformations of \mathbb{R}^d which are obtained as the flows of time-dependent vector fields. More precisely, given a time-dependent integrable vector field $t \in [0, 1] \mapsto v(t, \cdot) \in C_0^1(\mathbb{R}^d, \mathbb{R}^d)$ (the space of C^1 vector fields on \mathbb{R}^d which vanish at infinity as well as their first derivatives). Then the flow map φ^v of v , defined by $\varphi^v(t, x) = x + \int_0^t v(s, \varphi^v(s, x))ds$, can be shown to be a C^1 -diffeomorphism of \mathbb{R}^d . The group of all such flows taken at the terminal time $t = 1$ can be further equipped with a metric as follows. One first restricts to a given Hilbert space V of vector fields in $C_0^1(\mathbb{R}^d, \mathbb{R}^d)$ (e.g. a Sobolev space of sufficient regularity or reproducing kernel Hilbert space with adequate kernel function) and define the energy of the flow map $\varphi^v(1, \cdot)$ by $\int_0^1 \|v(t, \cdot)\|_V^2 dt$. This provides a quantitative measure of deformation cost and, given two shapes Ω and Ω' in \mathbb{R}^d , an optimal transformation mapping Ω to Ω' can be found by introducing the registration problem:

$$\inf_{v \in L^2([0, 1], V)} \left\{ \int_0^1 \|v(t, \cdot)\|_V^2 dt \mid \varphi^v(\Omega) = \Omega' \right\} \quad (6)$$

In practice, it is common to further relax the boundary constraint $\varphi^v(\Omega) = \Omega'$ based on a similarity measure between the deformed domain $\varphi^v(\Omega)$ and Ω' . This is typically done to ensure robustness of the method to noise and inconsistencies between the two shapes.

Optimality conditions for the above problem can be derived from the Pontryagin maximum principle of optimal control (c.f. [61] Chap. 10) from which it can be shown that a minimizing vector field for (6) can

be parameterized by an initial costate (or momentum) variable. Furthermore, when Ω is a discrete shape (i.e. represented by a finite set of vertices), this initial momentum can be identified with a distribution of vectors over the vertices of Ω . This allows to reduce the registration problem to a minimization over the finite-dimensional initial momentum variable, which can be tackled for instance with a shooting approach. For the applications presented in this paper, we make use specifically of the implementation of diffeomorphic registration from the MATLAB library *FshapesTk*¹.

5.2. Principal Component Analysis

Consider p landmark points on Ω^0 with known point correspondences on each Ω^{θ_i} given N domains, $A \in \mathbb{R}^{p \times N}$ denotes the Euclidean difference between the target domains and the reference domain in each direction. We define $\bar{A} = A - \mu$, where \bar{A} is normalized to zero-mean. We perform the singular value decomposition (SVD) $\bar{A} = U\Sigma V^T$, where $\bar{A} \in \mathbb{R}^{N \times p}$, $U \in \mathbb{R}^{N \times p}$, $\Sigma \in \mathbb{R}^{p \times p}$, and $V^T \in \mathbb{R}^{p \times p}$. We truncate U and Σ to first p' rank and result in reduced matrices as $U_r \in \mathbb{R}^{N \times p'}$, $\Sigma_r \in \mathbb{R}^{p' \times p'}$, and $V_r^T \in \mathbb{R}^{p' \times p'}$. We define the coefficient matrix $D = U_r \Sigma_r$, where each row in $D \in \mathbb{R}^{N \times p'}$ is a reduced representation of the deformation from Ω^θ to Ω^0 , referred to as reduced shape parameter $\tilde{\theta}$.

5.3. Domain Generation

We generate the computational domain in Example 1 following the equations below.

$$\begin{cases} x = X + a_1 \cos(a_2\pi + a_3) \sin(a_2\pi + a_3), \\ y = Y + b_1 \sin^2(b_2\pi + b_3), \\ a_1, b_1, a_3, b_3 \sim U[-0.5, 0.5], \quad a_2, b_2 \sim U[0, 0.75], \end{cases}$$

where $X, Y \in [0, 1] \times [0, 1]$ are the coordinates in the reference domain and x, y are their corresponding coordinates on newly generated shape. a_i, b_j are shape parameters, all chosen independently as follows: a_1, a_3 and b_1, b_3 from the uniform distribution in $[-0.5, 0.5]$ and a_2, b_2 from the uniform distribution on $[0, 0.75]$.

In Example 2, we define the reference domain as an annulus with inner and outer radii at 0.5 and 1.0, respectively. The parametric family of domains is rotated ellipsis with a void. We sample the shape parameters following:

$$c_1 \sim U[1.5, 2.0], c_2 \sim U[1.0, 1.5], c_3 \sim U[0, \pi], c_4 \sim U[0.2, 0.8], \quad (7)$$

Here, c_1 and c_2 stand for the length of the semi-major and minor axes, c_3 is the rotational angle, and c_4 is the radius of the inner circle.

¹<https://plmlab.math.cnrs.fr/benjamin.charlier/fshapesTk>

5.4. Neural Operator

We briefly summarize the architecture of MIONet in this section [29]. Notice that our framework is inclusive of the architecture of the neural operator we use to approximate \mathcal{F}_0 . We denote $\tilde{\mathcal{F}}_0$ as an approximation of the mapping from the $m+1$ input functions $[\theta, v_1, \dots, v_m]$ to the output function u . We remove the subscript that denotes the domain for clarity. We then define $m+1$ branch networks that input a discrete representation of each function and a trunk network that takes coordinates $x \in \Omega^0$ as input. Finally, the output function value at point x is approximated by

$$\tilde{\mathcal{F}}_0(\theta, v_1, \dots, v_m)(x) = \mathcal{S} \left(\underbrace{\tilde{\mathbf{g}}_0(\varphi_{q_0}^0(\theta))}_{\text{branch}_0} \odot \underbrace{\tilde{\mathbf{g}}_1(\varphi_{q_1}^1(v_1))}_{\text{branch}_1} \odot \dots \odot \underbrace{\tilde{\mathbf{g}}_m(\varphi_{q_m}^m(v_m))}_{\text{branch}_m} \odot \underbrace{\tilde{\mathbf{f}}(x)}_{\text{trunk}} \right), \quad (8)$$

where $\tilde{\mathbf{g}}_i$ (called branch net i) and $\tilde{\mathbf{f}}$ (called trunk net) are neural networks, \odot is the Hadamard product, and \mathcal{S} is the summation of all the components of a vector [29]. Eq. (8) can be written in a simplified form as:

$$\tilde{\mathcal{F}}_0(\theta, v_1, \dots, v_m)(x) = \sum_{i=1}^k t_i \prod_{j=1}^m b_i^j b_i^{geo} + b.$$

b_i^{geo} is the output of Geo. branch given θ , t_i is the trunk network at i th neuron, and b_i^j is the output of i th neuron from j th branch network. We refer the readers to [29, 38] for more details. In this work, we adopt the mean squared error (MSE) as the loss function. Higher-order constraints can be added to the network as further penalization [60].

5.5. Cohort and Simulations Details in Example 3

We used data from the Hopkins Hypertrophic Cardiomyopathy (HCM) Registry, which includes patients who were treated at the HCM Center of Excellence from 2005 to 2021. Our dataset consists of de-identified late gadolinium-enhanced cardiac magnetic resonance (LGE-CMR) scans from over 850 patients. The LGE-CMR scans have slice thicknesses ranging from $6 - 10 \text{ mm}$ and spatial resolutions varying from 1.17 to 2.56 mm . Notice that some patients have multiple scans over time, which sums up to 1007 scans.

To simulate electrical propagation, we first segment the myocardium and the blood pool from the raw images. We then reconstruct a mesh and generate fibers for each LV based on the segmented images. The average length of mesh size is 400 microns, leading to the average number of vertices at the order of 4 millions [8, 49]. To account for the directionality of the electrical signal propagated along the fibers, we adopt the validated rule-based method proposed in [6] to generate the myocardial fibers in each patient's LV.

Next, we divide the myocardium surface into 7 segments, which is a simplified version of the standardized myocardium segmentation from American Heart Association (AHA) [12, 16]. Within each segment highlighted with a different color in Fig. 4, we simulate a paced beat for 600 ms at a randomly-picked endocardial site until the LV is fully repolarized. Specifically, ATs and RTs are chosen as the time when the transmembrane potential u reaches 0 mV from the resting state (-85 mV) and the time when u decreases

below -70 mV, recovering from depolarization. More details regarding the solver and simulation procedure can be found in [48, 49].

6. Acknowledgement

We would like to thank Dr. Zongyuan Li, Dr. Adityo Prakosa and Dr. Shane Loeffler for multiple discussions during the preparation of this manuscript. M.Y., R.B., N.T. are supported by grant NIH. M.Y. would like to acknowledge support from Heart Rhythm Society Fellowship. L.L. was supported by the U.S. Department of Energy [DE-SC0022953]. N.C. was supported by NSF grant DMS-1945224. M.M. is supported by AFOSR awards FA9550-20-1-0288, FA9550-21-1-0317, FA9550-23-1-0445.

Appendix A. Universal approximation theorem for the Laplace equation

We provide here a more detailed explanation of the case of the Laplace equation discussed in Secs. 3.1 and 3.2. We specifically focus on showing how the assumptions of Theorem 1 on the solution operator \mathcal{F}_0 can be satisfied with adequate regularity conditions. We will restrict to the case where the reference domain Ω^0 has a C^2 boundary (and thus so do all the domains Ω^θ) and where boundary functions belong to the space $H^{3/2}(\partial\Omega^0)$. Let us then consider $\varphi \in C^2(\Omega^0, \mathbb{R}^d)$ a C^2 diffeomorphism from Ω^0 to its image $\varphi(\Omega^0)$ as well as $g \in H^{3/2}(\partial\Omega^0)$. Following the discussion and notations in Sec. 3.2, we are interested in the solutions of the Dirichlet problem:

$$\begin{cases} \varphi^* \Delta (\varphi^{-1})^* u = 0 & \text{in } \Omega^0, \\ u = g & \text{on } \partial\Omega^0 \end{cases}$$

which are the pullback of the solutions of the Laplace equation on the deformed domain $\varphi(\Omega^0)$. To be mathematically rigorous, the boundary condition needs to be taken in the sense of the trace on Sobolev spaces, that is, $T(u) = g$ where $T : H^1(\Omega^0) \rightarrow H^{1/2}(\partial\Omega^0)$ is the trace operator on Ω^0 . The operator $L_\varphi = \varphi^* \Delta (\varphi^{-1})^*$ can be expressed more explicitly with respect to the derivatives of φ and takes the form:

$$L_\varphi u = \sum_{i,j=1}^d a_\varphi^{ij}(x) \partial_i \partial_j u + \sum_{i=1}^d b_\varphi^i(x) \partial_i u$$

and the coefficient functions are specifically given by:

$$a_\varphi^{ij}(x) = \sum_{k=1}^d \partial_i \psi_k(\varphi(x)) \partial_j \psi_k(\varphi(x)), \quad b_\varphi^i(x) = \Delta \psi_i(\varphi(x))$$

where $\psi = \varphi^{-1}$ is a C^2 map from $\varphi(\Omega^0) \rightarrow \Omega^0$ and ψ_j denotes the j -th coordinate of ψ . Given that both φ and ψ are C^2 maps, a_φ^{ij} are C^1 functions and b_φ^i continuous functions on Ω^0 . Furthermore, the strict ellipticity of the resulting operator L_φ can be easily shown using the continuity of the derivatives of the inverse mapping ψ on the compact domain $\varphi(\Omega^0)$. Since any boundary function $g \in H^{3/2}(\partial\Omega^0)$ is the trace of a function in $H^2(\Omega^0)$, it follows from [22] (Theorems 8.3 and 8.12) that the Dirichlet problem has a unique

weak solution $u_{\varphi,g}$ which further satisfies $u_{\varphi,g} \in H^2(\Omega^0)$ and the PDE $L_\varphi u_{\varphi,g}(x) = 0$ holds for almost all $x \in \Omega^0$.

In order to show the continuity of $u_{\varphi,g}$ with respect to φ and g , we shall rely on an implicit function argument inspired from [27]. Denoting by $\text{Emb}^2(\Omega^0, \mathbb{R}^d)$ the space of C^2 embeddings from Ω^0 to \mathbb{R}^d equipped with the standard \mathcal{C}^2 -norm, we introduce the following mapping:

$$G : H^2(\Omega^0) \times \text{Emb}^2(\Omega^0, \mathbb{R}^d) \times H^{3/2}(\partial\Omega^0) \rightarrow L^2(\Omega^0) \times H^{3/2}(\partial\Omega^0)$$

$$(u, \varphi, g) \mapsto (L_\varphi u, T(u) - g)$$

Note that u is a solution of the Dirichlet problem for φ and g if and only if $G(u, \varphi, g) = (0, 0)$. Furthermore, as L_φ and T are continuous linear operators, G is continuously Fréchet differentiable with respect to u and g . Moreover, with the assumptions made on the domain and on the functions regularity, it follows from the results of [27] Chap. 2 that G is also continuously Fréchet differentiable with respect to φ . Now, given any (u, φ, g) such that $G(u, \varphi, g) = (0, 0)$, we see that for any $v \in H^2(\Omega^0)$, $D_u G(u, \varphi, g) \cdot v = (L_\varphi v, T(v))$. Using again the existence and uniqueness of the solution to the Dirichlet problem stated in the previous paragraph (for φ and the boundary function $T(v)$), we deduce that $D_u G(u, \varphi, g)$ induces an isomorphism between $H^2(\Omega^0)$ and $L^2(\Omega^0) \times H^{3/2}(\partial\Omega^0)$. The implicit function theorem then allows to conclude that, locally, the solution $u_{\varphi,g}$ can be expressed as a Fréchet differentiable function of φ and g . In particular, $u_{\varphi,g}$ is a continuous function of the embedding φ and the boundary function $g \in H^{3/2}(\partial\Omega^0)$.

Going back to the illustrative example of Sec. 3.2, as it is also assumed that $\theta \mapsto \varphi_\theta$ is continuous, the above analysis shows that the solution operator \mathcal{F}_0 for that Dirichlet problem is indeed well-defined and continuous with respect to both the shape parameter $\theta \in \Theta$ and boundary condition $g^0 \in H^{3/2}(\Omega^0)$. Consequently, the universal approximation theorem applies to that operator restricted to $\Theta \times K$ where K is any compact subset of $H^{3/2}(\partial\Omega^0)$.

Appendix B. Network training with a reduced and a true shape parameter

To compare the network performance based on PCA coefficients and the true shape parameters ($a_{1,2,3}$ and $b_{1,2,3}$) in Example 1, we tabulate the mean absolute error and mean relative L_2 error in TableB.2. For consistency, we retrain the network in Example 1 using the true parameters as input (See. Sec. 5.3) and PCA coefficients truncated at 15th modes while keeping other environments the same, such as hidden layers, activation functions, training epochs, or GPU machines. It is shown in Table B.2 that network performance under two circumstances is almost the same, with the error using PCA being slightly lower than when training with the true shape parameters at 50,000 epochs.

Appendix C. Affine Polar Transformation and LDDMM

In Fig. C.6, we present how landmarks are distributed on Ω^0 and Ω^{θ_i} in Example 2. Affine polar transformation, as a special approach to match landmarks distributed on two shapes, has a more uniform point

	Abs. Err.	Rel. Err.
PCA coefficients ($n = 15$)	5.3×10^{-3}	9.5×10^{-3}
true shape parameter	5.9×10^{-3}	1.0×10^{-2}

Table B.2: **Network prediction errors given the true shape parameters and PCA coefficients.** PCA is truncated at 15th modes.

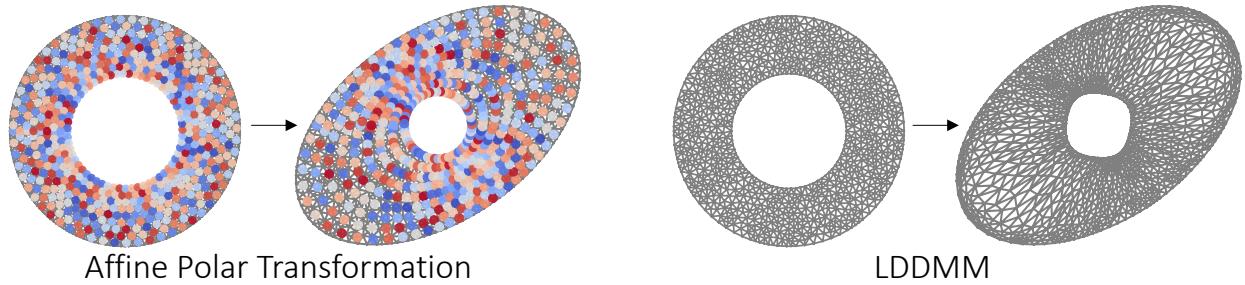


Figure C.6: **Affine Polar Transformation vs LDDMM.**

distribution. However, this approach lacks generality and can not be applied to general meshes. LDDMM, as a more general shape-matching algorithm, can register points distributed on arbitrary diffeomorphic shapes by morphing the mesh of Ω^0 to approximate Ω^{θ_i} . Yet, given its generality, it approximates the target shape with an error, displayed by the non-circular shape on the right panel of Fig. C.6, and typically leads to a less uniform distribution of landmarks. These two reasons influence the accuracy of the encoding of the shape and of the sampling of training data, which are likely the case of the slightly larger testing error when using LDDMM than when using affine transformations.

Appendix D. Network details

We present network structure and training details in this section. Table D.3 shows the network size, training epochs, and training time on a single GPU for each example. We only adopt feedforward neural networks (FNNs) for all subnetworks. Let us take Example 1 for instance, the input dimension of the Geo. branch in Example 1 is 20 with 2 hidden layers. All the other subnetworks have two or three hidden layers. The nonlinear activation function is a hyperbolic tangent function. The number of training epochs ranges from 10,000 to 200,000 for each example, depending on the complexity and the size of the training datasets. All training processes are completed on an NVIDIA GeForce RTX 2060 GPU. Training the network in Example 1 and 3 costs less than 0.3 hours and the training in Example 2 costs 8.1 hours. Notably in Example 3, we expand the 1D rotational angle θ into a 2D vector, $[\cos \theta, \sin \theta]$, to impose the periodicity of the ATs/RTs at $\theta = 0$ and 2π .

	Branch 1 (Geo)	Branch 2	Trunk	Epochs	Training Time (h)
Ex. 1	[20, 100, 100, 100]	[68, 150, 150, 150, 100]	[2, 100, 100, 100]	50,000	0.3
Ex. 2	[75, 300, 300, 200]	[50, 200, 200, 200, 200]	[2, 100, 100, 100]	100,000	8.1
Ex. 3	[180, 250, 250, 250, 200]	[3, 200, 200, 200, 200]	[4, 200, 200, 200, 200]	10,000	0.17

Table D.3: Network size, training epochs, and training time.

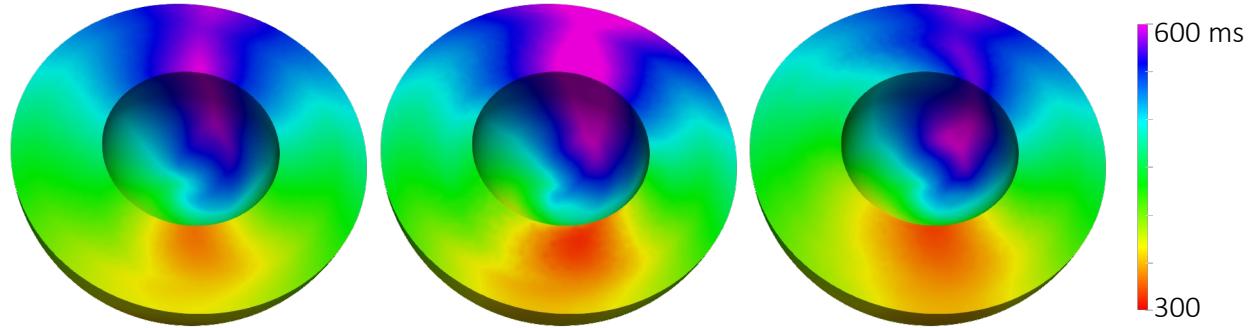


Figure E.7: RTs mapped in the canonical domain with stimulation at the same endocardial point.

Appendix E. Different PDE solutions on the same canonical shape

In Fig. E.7, we conduct simulations and display RTs from three distinct LVs on the canonical domain. For these three simulations, we keep all other variables, such as model parameters and pacing location, constant. Although the shapes of these LVs are distinctive, the UVCs of the pacing locations are the same. Thereby, any differences in RTs are solely induced by the variations in geometry (or “fibers”). Myocardium tissue is stimulated on one side of the endocardium, where electric signals travel around the LV and eventually converge on the side of the LV. In the left panel, we observe that the tissue around the pacing location takes longer to repolarize compared to the other LVs. It takes a long period for the middle geometry to return to the repolarization state. This is indicated by a larger region of purple color, representing a longer repolarization time. These simulations vividly illustrate how the propagation of cardiac electrical signals is influenced by the geometry of the LV.

Appendix F. Shape Outliers

Fig. F.8 presents three LV shapes from the case with the largest relative error in Fig. 5. It is evident that these shapes poorly reflect the true shape of the LV, which can be attributed to three main reasons: poor segmentation, low-quality data, and misaligned imaging. The deep learning algorithms used for segmentation may erroneously distinguish the myocardium from the blood pool, leading to poor shape reconstruction [34]. The issue is exacerbated when segmenting low-quality images. Also, we noticed that the MRI scans for a few patients are not aligned well due to patient’s breathing or inconsistent cardiac cycle during data collection.

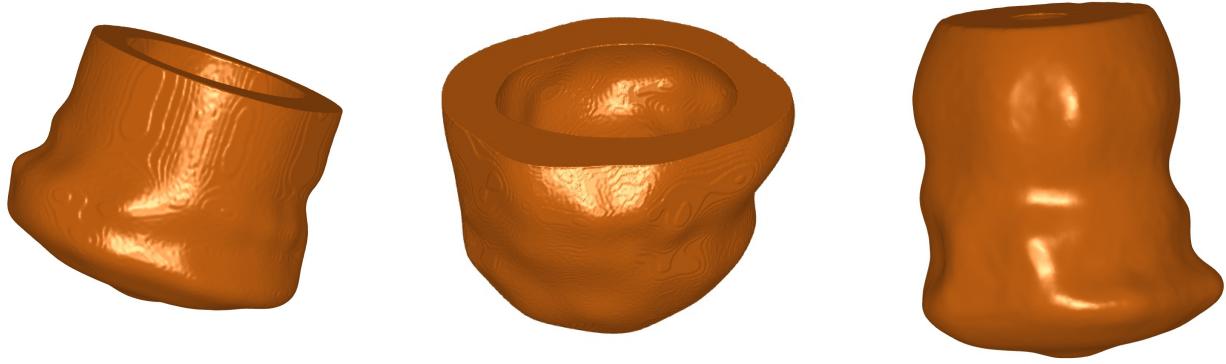


Figure F.8: **Bad shape reconstruction leads to larger errors that are presented in Fig. 5**

Calculating the distance between these outlier shapes to other shapes can help us to quantitatively understand the learning performance as a function of data distribution. Different metrics, including those used in large deformation diffeomorphic metric mapping (LDDMM), or Kendall’s distance, can be adopted to quantify the distance [7, 25]. The influence of the distance between shapes on learning performance of DIMON is an interesting problem, beyond the scope of this paper.

References

- [1] Avants, B. and J. C. Gee. Geodesic estimation for large deformation anatomical shape averaging and interpolation. *Neuroimage* 23:S139–S150, 2004.
- [2] Azencot, O., M. Ben-Chen, F. Chazal, and M. Ovsjanikov. An operator approach to tangent vector field processing. In: *Computer Graphics Forum*, volume 32, pp. 73–82 (Wiley Online Library, 2013).
- [3] Babuška, I. and M. Suri. The p-and hp versions of the finite element method, an overview. *Computer methods in applied mechanics and engineering* 80:5–26, 1990.
- [4] Bauer, M., P. Harms, and P. W. Michor. Sobolev metrics on shape space of surfaces. *Journal of Geometric Mechanics* 3:389–438, 2012.
- [5] Bayer, J., A. J. Prassl, A. Pashaei, J. F. Gomez, A. Frontera, A. Neic, G. Plank, and E. J. Vigmond. Universal ventricular coordinates: A generic framework for describing position within the heart and transferring data. *Medical image analysis* 45:83–93, 2018.
- [6] Bayer, J. D., R. C. Blake, G. Plank, and N. A. Trayanova. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Annals of biomedical engineering* 40:2243–2254, 2012.
- [7] Beg, M. F., M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision* 61:139–157, 2005.

- [8] Boyle, P. M., A. R. Ochs, R. L. Ali, N. Paliwal, and N. A. Trayanova. Characterizing the arrhythmogenic substrate in personalized models of atrial fibrillation: sensitivity to mesh resolution and pacing protocol in af models. EP Europace 23:i3–i11, 2021.
- [9] Bucelli, M., M. Salvador, A. Quarteroni *et al.* Multipatch isogeometric analysis for electrophysiology: simulation in a human heart. Computer Methods in Applied Mechanics and Engineering 376:113666, 2021.
- [10] Cai, S., Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. Acta Mechanica Sinica 37:1727–1738, 2021.
- [11] Cao, Q., S. Goswami, and G. E. Karniadakis. Lno: Laplace neural operator for solving differential equations. arXiv preprint arXiv:2303.10528 , 2023.
- [12] Cerqueira, M. D., N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan *et al.* Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the cardiac imaging committee of the council on clinical cardiology of the american heart association. Circulation 105:539–542, 2002.
- [13] Chen, T. and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE transactions on neural networks 6:911–917, 1995.
- [14] Choi, G. P. and L. M. Lui. Recent developments of surface parameterization methods using quasi-conformal geometry. Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision pp. 1–41, 2022.
- [15] Cury, C., J. Glaunès, M. Chupin, and O. Colliot. Analysis of anatomical variability using diffeomorphic iterative centroid in patients with alzheimer’s disease. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 5:350–358, 2017.
- [16] Deng, D., A. Prakosa, J. Shade, P. Nikolov, and N. A. Trayanova. Sensitivity of ablation targets prediction to electrophysiological parameter variability in image-based computational models of ventricular tachycardia in post-infarction patients. Frontiers in Physiology 10:628, 2019.
- [17] Desbrun, M., M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. In: Computer graphics forum, volume 21, pp. 209–218 (Wiley Online Library, 2002).
- [18] Donati, N., E. Corman, S. Melzi, and M. Ovsjanikov. Complex functional maps: A conformal link between tangent bundles. In: Computer Graphics Forum, volume 41, pp. 317–334 (Wiley Online Library, 2022).

- [19] Doste, R., D. Soto-Iglesias, G. Bernardino, A. Alcaine, R. Sebastian, S. Giffard-Roisin, M. Sermesant, A. Berhuezo, D. Sanchez-Quintana, and O. Camara. A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts. International journal for numerical methods in biomedical engineering 35:e3185, 2019.
- [20] Fan, H., H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 605–613 (2017).
- [21] Fletcher, P. T., C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE transactions on medical imaging 23:995–1005, 2004.
- [22] Gilbarg, D., N. S. Trudinger, D. Gilbarg, and N. Trudinger. Elliptic partial differential equations of second order, volume 224 (Springer, 1977).
- [23] Grenander, U. and M. I. Miller. Computational anatomy: An emerging discipline. Quarterly of applied mathematics 56:617–694, 1998.
- [24] Hao, Z., Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu. Gnot: A general neural operator transformer for operator learning. In: International Conference on Machine Learning, pp. 12556–12569 (PMLR, 2023).
- [25] Hartman, E., Y. Sukurdeep, E. Klassen, N. Charon, and M. Bauer. Elastic shape analysis of surfaces with second-order sobolev metrics: a comprehensive numerical framework. International Journal of Computer Vision 131:1183–1209, 2023.
- [26] He, J., S. Koric, S. Kushwaha, J. Park, D. Abueidda, and I. Jasiuk. Novel deeponet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads. arXiv preprint arXiv:2306.03645 , 2023.
- [27] Henry, D. Perturbation of the boundary in boundary-value problems of partial differential equations. 318 (Cambridge University Press, 2005).
- [28] Jerison, D. and C. E. Kenig. The inhomogeneous Dirichlet problem in Lipschitz domains. Journal of functional analysis 130:161–219, 1995.
- [29] Jin, P., S. Meng, and L. Lu. Mionet: Learning multiple-input operators via tensor product. SIAM Journal on Scientific Computing 44:A3490–A3514, 2022.
- [30] Joshi, S., B. Davis, M. Jomier, and G. Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. NeuroImage 23:S151–S160, 2004.
- [31] Karniadakis, G. and S. J. Sherwin. Spectral/hp element methods for computational fluid dynamics (Oxford University Press, USA, 2005).

- [32] Karniadakis, G. E., I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics* 3:422–440, 2021.
- [33] Lagaris, I. E., A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks* 9:987–1000, 1998.
- [34] Lefebvre, A. L., C. A. Yamamoto, J. K. Shade, R. P. Bradley, R. A. Yu, R. L. Ali, D. M. Popescu, A. Prakosa, E. G. Kholmovski, and N. A. Trayanova. Lassnet: A four steps deep neural network for left atrial segmentation and scar quantification. In: *Challenge on Left Atrial and Scar Quantification and Segmentation*, pp. 1–15 (Springer, 2022).
- [35] Li, Z., D. Z. Huang, B. Liu, and A. Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *arXiv preprint arXiv:2207.05209* , 2022.
- [36] Li, Z., N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895* , 2020.
- [37] Li, Z., N. B. Kovachki, C. Choy, B. Li, J. Kossaifi, S. P. Otta, M. A. Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli *et al.* Geometry-informed neural operator for large-scale 3d pdes. *arXiv preprint arXiv:2309.00583* , 2023.
- [38] Lu, L., P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence* 3:218–229, 2021.
- [39] Lu, L., X. Meng, Z. Mao, and G. E. Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review* 63:208–228, 2021.
- [40] Lu, L., R. Pestourie, S. G. Johnson, and G. Romano. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Physical Review Research* 4:023210, 2022.
- [41] Ma, J., M. I. Miller, and L. Younes. A bayesian generative model for surface template estimation. *Journal of Biomedical Imaging* 2010:1–14, 2010.
- [42] Miller, M. I., A. Trouvé, and L. Younes. On the metrics and euler-lagrange equations of computational anatomy. *Annual review of biomedical engineering* 4:375–405, 2002.
- [43] Newell, A. C. and J. A. Whitehead. Finite bandwidth, finite amplitude convection. *Journal of Fluid Mechanics* 38:279–303, 1969.
- [44] Ovsjanikov, M., M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)* 31:1–11, 2012.

- [45] Ovsjanikov, M., E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In: SIGGRAPH ASIA 2016 Courses, pp. 1–60 (2016).
- [46] Park, J. J., P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 165–174 (2019).
- [47] Patera, A. T. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of computational Physics* 54:468–488, 1984.
- [48] Plank, G., A. Loewe, A. Neic, C. Augustin, Y.-L. Huang, M. A. Gsell, E. Karabelas, M. Nothstein, A. J. Prassl, J. Sánchez *et al.* The opencarp simulation environment for cardiac electrophysiology. *Computer Methods and Programs in Biomedicine* 208:106223, 2021.
- [49] Prakosa, A., H. J. Arevalo, D. Deng, P. M. Boyle, P. P. Nikolov, H. Ashikaga, J. J. Blauer, E. Ghafoori, C. J. Park, R. C. Blake III *et al.* Personalized virtual-heart technology for guiding the ablation of infarct-related ventricular tachycardia. *Nature biomedical engineering* 2:732–740, 2018.
- [50] Praun, E. and H. Hoppe. Spherical parametrization and remeshing. *ACM transactions on graphics (TOG)* 22:340–349, 2003.
- [51] Raissi, M., P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378:686–707, 2019.
- [52] Rock, J., T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2484–2493 (2015).
- [53] Rubner, Y., C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision* 40:99–121, 2000.
- [54] Shukla, K., V. Oommen, A. Peyvan, M. Penwarden, L. Bravo, A. Ghoshal, R. M. Kirby, and G. E. Karniadakis. Deep neural operators can serve as accurate surrogates for shape optimization: a case study for airfoils. *arXiv preprint arXiv:2302.00807*, 2023.
- [55] Sun, X. and E. R. Hancock. Quasi-isometric parameterization for texture mapping. *Pattern Recognition* 41:1732–1743, 2008.
- [56] ten Tusscher, K. H., D. Noble, P.-J. Noble, and A. V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology-Heart and Circulatory Physiology* 286:H1573–H1589, 2004.

- [57] Ten Tusscher, K. H. and A. V. Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology* 291:H1088–H1100, 2006.
- [58] Vercauteren, T., X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage* 45:S61–S72, 2009.
- [59] Willems, R., K. L. Janssens, P. H. Bovendeerd, C. V. Verhoosel, and O. van der Sluis. An isogeometric analysis framework for ventricular cardiac mechanics. *Computational Mechanics* pp. 1–42, 2023.
- [60] Yin, M., E. Zhang, Y. Yu, and G. E. Karniadakis. Interfacing finite elements with deep neural operators for fast multiscale modeling of mechanics problems. *Computer methods in applied mechanics and engineering* 402:115027, 2022.
- [61] Younes, L. *Shapes and Diffeomorphisms*, volume 171 (Springer, 2019).
- [62] Yu, B. *et al.* The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics* 6:1–12, 2018.