Citation: Arbab, Hassan et al. "Linking Machine Learning with Multiscale Numerics: Data-Driven Discovery of Homogenized Equations." JOM, 72 (October 2020): 4444–4457 © 2020 The Author(s)

As Published: https://doi.org/10.1007/s11837-020-04399-8

Publisher: Springer US

Persistent URL: https://hdl.handle.net/1721.1/129683

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

Massachusetts Institute of Technology

Linking Machine Learning with Multiscale Numerics: Data-Driven Discovery of Homogenized Equations

Linking Machine Learning with Multiscale Numerics: Data-Driven Discovery of Homogenized Equations

Hassan Arbabi, Judith E. Bunder, Giovanni Samaey,
Anthony J. Roberts and Ioannis G. Kevrekidis [*][†]

September 25, 2020

Abstract

The data-driven discovery of partial differential equations (PDEs) consistent with spatiotemporal data is experiencing a rebirth in machine learning research. Training deep neural networks to learn such data-driven partial differential operators requires extensive spatiotemporal data. For learning coarse-scale PDEs from computational fine-scale simulation data, the training data collection process can be prohibitively expensive. We propose to transformatively facilitate this training data collection process by linking machine learning (here, neural networks) with modern multiscale scientific computation (here, equation-free numerics). These equation-free techniques operate over sparse collections of small, appropriately coupled, space-time subdomains ("patches"), parsimoniously producing the required macro-scale training data. Our illustrative example involves the discovery of effective homogenized equations in one and two dimensions, for problems with fine-scale material property variations. The approach holds promise towards making the discovery of accurate, macro-scale effective materials PDE models possible by efficiently summarizing the physics embodied in "the best" fine-scale simulation models available.

Keywords: machine learning, homogenization, equation-free approach

---

[*]Corresponding author: `yannisk@jhu.edu`

[†]H.A. is with the Department of Mechanical Engineering, Massachusetts Institute of Technology, and Department of Chemical and Biomolecular Engineering at Johns Hopkins University. J.E.B. and A.J.R. are with School of Mathematical Sciences, University of Adelaide. G.S. is with the Computer Science Department at KU Leuven. I.G.K. is with the Departments of Chemical and Biomolecular Engineering & Applied Mathematics and Statistics at Johns Hopkins University.

1   Introduction

Evolutionary models of materials dynamic behavior, in the form of Partial Differential Equations (PDEs), embodying conservation laws supplemented by appropriate closures, traditionally form the backbone of computational materials modeling. The requisite closures were initially mostly phenomenological, guided by experimentation, but in recent years such closures increasingly come from fine-scale, microscopic, possibly even *ab initio* computations. One of the signal promises of machine learning in its early days, and increasingly more so today, is the discovery of such evolutionary PDEs from spatiotemporal data — whether from physical observations or from computational observations obtained through fine-scale models.

Yet materials problems are inherently multiscale: fine-scale models typically operate at the atomistic level; but the questions asked, and the answers required, are usually macroscopic in nature [e.g. 1], and the models that describe macroscopic materials physics are coarse-scale, effective PDEs in terms of macroscopic observables.

When the macroscopic Quantities-of-Interest (QoI), or the "right variables", in terms of which an effective macroscopic PDE can be written, are known, there already exist data-driven, machine-learning-assisted techniques that will successfully approximate a PDE for the QoI evolution from macroscopic spatiotemporal observations. These techniques are based on the simple observation that *the law of the PDE* is a functional relation (a constraint) between local space and time derivatives of the field(s) of interest: for simple diffusion, the time derivative of the local density is a function of its second-order space derivative. From detailed spatiotemporal data, one can thus acquire extensive sampling of instantiations of this relation: every space-time point in a movie provides a training data point for a machine learning model that can learn a PDE right-hand-side (i.e., a differential operator) consistent with the observed movie, typically assuming the PDE is homogeneous in space-time. Such neural-network-based techniques were certainly proposed and implemented in the 1990s for ODEs [e.g. 2, 3] and PDEs [e.g. 4]. Interestingly, several neural net architectural features (recurrent networks, convolutional networks) that proved broadly successful later, arose naturally in the work described in these references. The field has exploded recently, and tools beyond deep neural networks (e.g., Gaussian Processes as well as Sparse Regression and Reservoir Computing) have been brought to the task [5, 6, 7, 8, 9, 10]. Gray box techniques (where part of the equation is known [e.g. 11, 12, 13]) as well as machine-learning techniques for solving known PDEs are increasingly successful [14, 15, 16, 17].

These approaches can learn approximate macroscopic PDEs based on data of macroscopic field evolution. The data may come from macroscopic measurements of physical experiments, or from coarse-grained observations of fine-scale simulations (e.g., running molecular dynamics, kinetic Monte Carlo or agent-based models, and observing the evolution of macroscopic, hydrodynamic level fields, such as densities, momenta or stresses). We recently discussed the learning of coarse-grained PDEs based on fine-scale simulators: the "inner", fine-scale simulator was a Lattice-Boltzmann (LB) model, while the "outer", coarse-grained fields were two concentration fields for a system of coupled reaction-diffusion equations [18]. The fine-scale LB simulation was performed "for all space and all times" over the domain of interest, which could lead to considerable computational effort. The goal of this paper it to alleviate, as much as possible, the extensive computational needs for the collection of the requisite macro-scale training data. Here we work on coarse-graining a class of problems different than the more "inner atomistic simulation" class illustrated by the LB example: we are interested in discovering coarse effective PDEs for materials that have micro-scale variations in their properties. In this case the "inner simulator" is the PDE that resolves that complete fine-scale material properties, while the coarse-grained PDE is the so-called "homogenized PDE" — a PDE that governs the effective, long-wavelength material response. One may loosely

2

think of this as a PDE for the locally averaged (over the fine-scale variations) material response. In applied mathematics, this problem is the purview of *homogenization theory* [19] (see also [20]): using asymptotic techniques, and under specific assumptions, one can derive a closed-form analytical expression for the homogenized PDE. The homogenized PDE solution is the part of the original solution that does not exhibit micro-scale variation. Therefore, the homogenized PDE is the right *effective* coarse-scale PDE for materials with microscopic heterogeneity. However, the analytical derivation of homogenized PDEs is typically based on assumptions and/or asymptotic arguments that may not be valid for many practical problems. In such (frequent!) cases, the data-driven discovery of a useful approximate effective PDE, may be the only practical alternative. Our plan in this paper is to mitigate this "training data collection" computation expense by taking advantage of modern multiscale scientific computing approaches (in particular, equation-free techniques) that are based on precisely the same assumption: that an effective equation exists, yet it is not available in closed form.

Equation-free multiscale computation comprises a suite of scientific computing techniques for solving macroscopic evolution equations based on parsimonious use of fine-scale simulators without ever deriving the macroscopic equations in closed form [21, 22]. Equation-free algorithms like coarse projective integration, gap-tooth and patch dynamics, as well as coarse bifurcation and stability analysis, are templated on established numerical techniques; the new element is that the quantities required for scientific computation (time derivatives, action of Jacobians) are not obtained from closed-form function evaluations, but rather from brief bursts of well-designed fine-scale simulation over short time intervals and small spatial domains. It is precisely this "closure on demand" property of equation-free computation that will help us collect the macroscopic data required for macroscopic PDE learning /approximation with significantly less computational effort. Such parsimonious design of micro-scale computations to collect macro-information adequate for learning, is the main ingredient of our work.

The rest of this paper is structured as follows. We start in Section 2 by describing homogenization problems arising from the modeling of materials with micro-scale property variation. The equation-free methods we use are described in Section 3; they help us collect large-space/long-time data at the homogenized level from minimal (small-space/short-time) finely resolved direct material simulation. We then use the collected macro-data to learn the effective, macro-scale, homogenized equation through the use of two neural network architecture variations. Section 4 describes our approach to data-driven PDE learning and the neural network architectures. In Section 5, we apply our framework to two illustrative problems. In each problem we compare the data-driven effective model we identify with analytically obtainable homogenized equations, thus establishing the utility but also highlighting some limitations of our approach. We conclude in Section 6 with a brief discussion of possible extensions and applications of our approach and some thoughts on the important issue of data-driven coarse variable selection for problems where even the appropriate QoI must be discovered from the data. A python implementation of our framework and the illustrative examples in Section 5 is available at https://github.com/arbabiha/homogenization_via_ML.

3

2 Homogenization problem

In many physical or engineering modeling problems, the properties of the material under investigation have variations (i.e., heterogeneity) at small spatial scales, while we are interested in predicting the response of the material at much larger scales. The *homogenization technique* is an asymptotic applied mathematics method that, under certain technical assumptions, derives (approximate) effective equations for the material response at large length scales. Here, we briefly review this technique through the example of diffusion in a material with heterogeneous diffusivity—comprehensive discussions are given elsewhere [19, 23].

Consider the unsteady diffusion problem for $u(x,t)$ in the one-dimensional spatial domain $[0,1]$ with homogeneous Dirichlet boundary conditions:

$$\begin{aligned}
\partial_t u(x,t) &= \partial_x \left[a\left(\frac{x}{\epsilon}\right) \partial_x u(x,t) \right], \\
u(x,0) &= u^0(x) \in L^2([0,1]), \\
u(0,t) &= u(1,t) = 0,
\end{aligned} \tag{1}$$

where the diffusivity $a(\cdot)$ is space-dependent, and periodic in its argument. We are interested in the case of $\epsilon \ll 1$, that is, the diffusivity has very rapid spatial oscillations. This is *the detailed problem*. Since ϵ is very small compared to the length scales of interest (e.g., the size of the domain), the goal of homogenization is to find an effective *homogenized PDE* that has no explicit dependence on ϵ, but its solution usefully approximates the solution to the detailed problem in its large-scale features. The classical approach to derive such a PDE is to expand the solution of the detailed problem in powers of ϵ,

$$u(x,t) = u_h(x,t) + \sum_{j=1}^{\infty} \epsilon^j u_j(x, x/\epsilon, t), \tag{2}$$

where the functions u_j, $j = 1, 2, \ldots$, are periodic in their second argument. The first term in the expansion, $u_h(x,t)$, is the homogenized solution and, to errors $O(\epsilon^2)$, it satisfies the homogenized PDE

$$\begin{aligned}
\partial_t u_h(x,t) &= \partial_x \left[a^* \partial_x u_h(x,t) \right], \\
u_h(x,0) &= u^0(x) \in L^2([0,1]), \\
u_h(0,t) &= u_h(1,t) = 0,
\end{aligned} \tag{3}$$

where *the effective diffusivity constant* is

$$a^* = \int_0^1 a(y) \left[1 - \frac{d}{dy}\chi(y) \right] dy. \tag{4}$$

In the above equation, $\chi(y)$ is the periodic solution to the so-called cell problem,

$$\frac{d}{dy}\left[a(y)\frac{d}{dy}\chi \right] = \frac{d}{dy} a(y), \tag{5}$$

which is usually accompanied by a normalization condition,

$$\int_0^1 \chi(y) dy = 0. \tag{6}$$

4

It is known that as $\epsilon \to 0$, the homogenized solution u_h converges to the solution of the detailed problem [19], and that at finite ϵ one can systematically find corrections to (3) for u_h [24]. Unlike the detailed field u, the homogenized solution does not have an explicit dependence on the fast-varying variable x/ϵ, and hence it represents the large-scale component of the detailed solution. When ϵ is small enough, one might consider approximating the homogenized solution by applying a low-pass filter to the detailed solution (i.e., filtering out variations with wavelengths of order ϵ or smaller). In this paper, we take the coarse scale (low-pass filtered, locally averaged) density to be our effective macroscopic variable—which we expect approximates u_h—and propose a framework to discover its evolution PDE from data.

Application of the rigorous analytical homogenization process, outlined above, to PDEs suffers from limitations because its validity requires somewhat stringent mathematical assumptions and asymptotic limits. For example, in the general case of non-periodic heterogeneity, no simple effective diffusivity formula can be derived; indeed, no systematic approach for deriving it yet exists [23]. On the other hand, direct numerical simulation of heterogeneous materials requires a computational power scaling as at least $(1/\epsilon)^d$, with d being the spatial dimension, which makes such problems computationally prohibitive.

3 Equation-free approach for homogenization

The equation-free approach, introduced by Kevrekidis et al. [21], when applied to homogenization problems [25, 22, 26, 27] provides a computational shortcut for approximating the homogenized solutions of systems when we only know the detailed problem. The homogenized equation is *assumed to exist, yet not available in closed form*. The basic idea behind the equation-free approach is to exploit the *implicitly assumed* existence of the macro-scale dynamics to speed up the micro-scale simulations and perform tasks like prediction, optimization or even bifurcation analysis at the large-scale level. In the homogenization case, the micro-scale evolution is given by detailed, fine-scale PDE and the macro-scale description is the effective PDE. When a mathematically obtained homogenized PDE is available and accurate, we expect the equation-free numerics to accurately approximate the solutions of this homogenized PDE.

Here, we utilize the equation-free approach for parsimonious generation of data: as explained below, the equation-free methodology guides us to simulate the detailed problem only in a fraction of space-time, to obtain training data for learning the effective (homogenized) equation, and hence, it substantially reduces the computational complexity of collecting training data. In describing the equation-free approach for homogenization, we closely follow [22] and we refer the reader to [26, 27, 28] for further analysis of this approach in the context of homogenization. A simplified representation of the equation-free approach is visualized in Figure 1.

A basic algorithm in the equation-free toolbox for spatially distributed systems is the *gap-tooth* scheme. In this approach, we populate the spatial domain with sparsely distributed small boxes (subdomains, called *teeth*) that only cover a small fraction of the spatial domain. The width of each tooth, h, is chosen to be considerably larger than ϵ but still small compared to the domain size. The distance between the centers of the teeth, Δx, is chosen large compared to h, so that the teeth occupy a small fraction of the full space. We only simulate the detailed problem within the teeth, with appropriately prescribed "coupling" boundary conditions (see below) and then define the gap-tooth solution u_g to be the detailed solution averaged over the tooth and reported at its
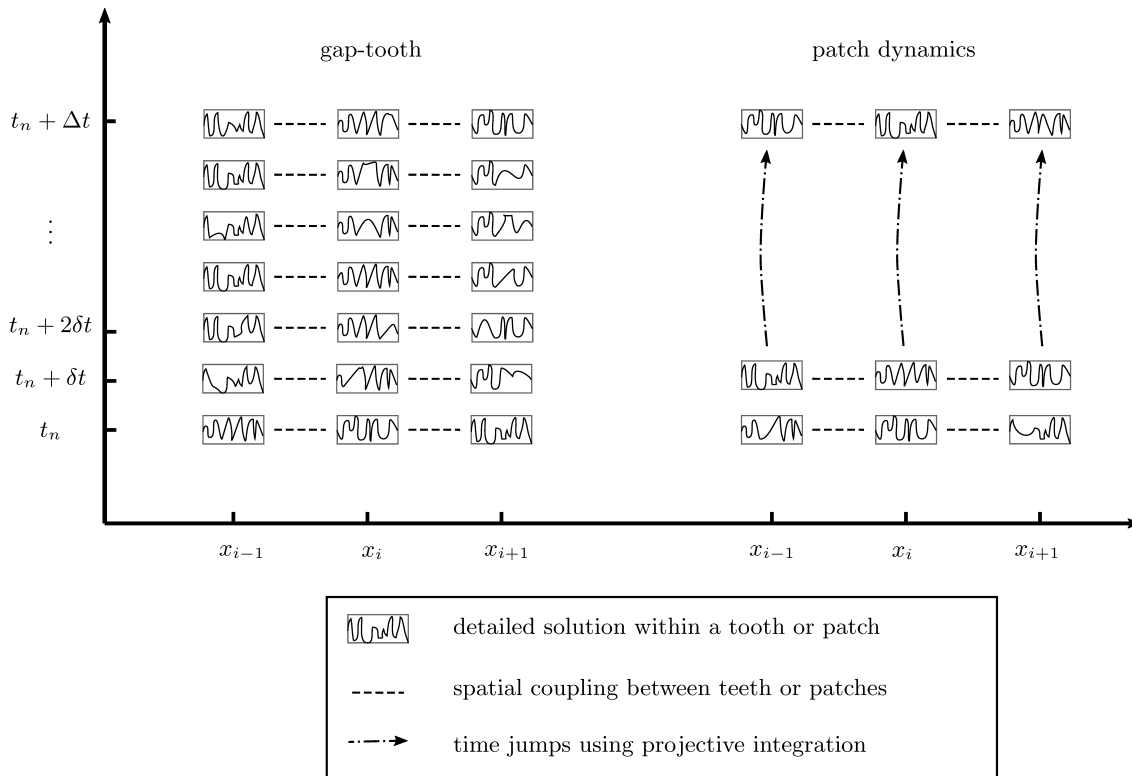
Figure 1: **Space-time grid of equation-free numerics.** In the gap-tooth scheme (left) the detailed problem is only simulated inside a sparse grid of teeth (boxes in the figure). The solution within adjacent teeth is coupled by implicitly utilizing the assumption that there exists a macro-scale variable which varies smoothly in the space. Similarly, in the patch dynamics, the simulation is carried out in sparse set of small boxes (called patches), but the smoothness of the macro-scale variable in time is leveraged to make the time grid sparse as well: the time derivative is estimated over the interval δt and the solution is extrapolated over $\Delta t \gg \delta t$ using that estimate.

midpoint. Let x_i be the position of the ith tooth center; then

$$u_g^i := \frac{1}{h} \int_{x_i - h/2}^{x_i + h/2} u(\xi, t) d\xi, \qquad i = 1, \dots, N \tag{7}$$

is the gap-tooth prediction/order-parameter/amplitude for the ith tooth, which approximates the discretized macro-scale variable, denoted by U. In many scenarios [e.g. 29, 30, 31] we can prove that such a macro-scale field U exists and is governed by an explicitly unknown PDE (here, hopefully, the homogenized PDE), and that variations of U can be faithfully estimated using its value on the grid of such teeth centers (rather than a highly resolved grid designed to capture the detailed, ϵ-scale material property variations). Based on the assumption of the appropriateness of such a macro-scale field behavior, artificial boundary conditions for each tooth have been devised for the meaningfully coupled simulation of the micro-scale problem across all teeth [e.g. 29, 28]. These teeth-edge conditions constitute an important ingredient of the gap-tooth scheme, as they couple the dynamics of adjacent teeth and allow the global pattern of the solution to emerge from the simulations in (seemingly) separate teeth. One typically assumes that the macro-scale field U resembles a polynomial between teeth, that is,

$$U(x, t_n) \approx p_i^k(x, t_n), \qquad x \in [x_i - h/2, x_i + h/2] \tag{8}$$

with p_i^k denoting a polynomial of even degree k within the ith tooth. The coefficients of this polynomial are determined so that it results in the same box averages as the detailed solution within tooth i and $k/2$ teeth to its left and right (see [22] for the explicit formula of p_i^k). Now, for simulating the detailed solution within the tooth i during the entire duration of the next time step, we may use, as the tooth-edge Neumann condition, the slope of the polynomial [28], that is,

$$\frac{1}{\epsilon} \int_{x_i + h/2 - \epsilon}^{x_i + h/2 + \epsilon} \partial_\xi u(\xi, t) d\xi = \partial_x p_i^k \big|_{x_i + h/2},$$
$$\frac{1}{\epsilon} \int_{x_i - h/2 - \epsilon}^{x_i - h/2 + \epsilon} \partial_\xi u(\xi, t) d\xi = \partial_x p_i^k \big|_{x_i - h/2}. \tag{9}$$

In this way, the detailed evolution in ith tooth is informed by the polynomial approximation of the macro-scale field near that tooth, which in turn, is informed by the value of the macro-scale field in the neighboring teeth. Further analysis of the gap-tooth scheme and a discussion of its convergence and accuracy can be found in [26, 28]. In particular, it was shown that the gap-tooth solution scheme leads to a finite-difference approximation of the homogenized PDE itself.

A more advanced algorithm in the equation-free toolkit is *patch dynamics with buffers*[22]. In this scheme, the simulations of the detailed problem is performed not only in a fraction of space, but also in a fraction of time. This is made possible through the use of (coarse) projective integration [32, 33]: the gap-tooth simulations of the detailed problem are carried out over a few short time steps, and the result is used to compute an estimate of the time-derivative for the macro-scale variable, $\partial_t U$. This estimate is then used to take a large time step in the simulation of U. Taking large time steps for U is plausible, because we have assumed the existence of a macro-scale closure for U, which justifies using coarser grids in time compared to those necessary for the accurate simulation of the detailed problem. However, and similar to the gap-tooth scheme, the macro-scale closure is not available, and thus cannot be explicitly used in the algorithm. Instead, the quantities required for scientific computation (like the time derivatives here) are *estimated* from judiciously designed partial simulations of the detailed problem.

The grid setup for patch dynamics with buffers is similar to the gap-tooth scheme, except that we extend each patch to width $H > h$, with a buffer zone around each tooth. The buffers provide an alternative way of imposing meaningful coupling boundary conditions across neighboring teeth [31]. Detailed simulations *are also performed* in the buffer zones, yet the patch dynamics solution reports averages over the tooth itself. The role of the buffer zone is thus to "protect" the core solution within the tooth from the disturbances caused by the boundary conditions on the buffers. Now the teeth exchange information through the process of *lifting* defined below.

To round off our brief outline of the patch dynamics scheme, we define two more notions in multi-scale computation. Firstly, the *restriction operator* is a function that maps the detailed solution within patches to the macro-scale variable. Here, the restriction operator is simply averaging over each tooth in (7). Secondly, the *lifting operator* is a (one-to-many) mapping from the macro-scale field to a consistent detailed solution profile. The lifting operator is not unique, since there are many detailed profiles that have the same average. A standard choice for lifting in homogenization is a polynomial expansion similar to (8) within the ith tooth *and its surrounding buffer zone.* In the explicit form, this polynomial can be written as

$$u_i(x) = \sum_{k=0}^{d} D_i^k(U) \frac{(x - x_i)^k}{k!}, \quad x \in \left[x_i - \frac{H}{2}, x_i + \frac{H}{2} \right], \tag{10}$$

where $D_i^k(U)$ is the finite-difference approximation for the kth spatial derivative at the ith tooth.

A time step of the patch dynamics simulation is performed as follows: given the current state of the macro-state field U_i^t, $i = 1, \ldots, N$, the state of the detailed solution within each tooth and its buffer is constructed using the lifting in (10). Then the detailed equation is marched forward in the teeth and their buffers. The new macro-state $U_i^{t+\delta t}$, $i = 1, \ldots, N$, is computed by averaging in (7). Then, the time-derivative of the macro-state in each tooth is estimated as

$$\frac{d}{dt} U_i(t) \approx \frac{U_i^{t+\delta t} - U_i^t}{\delta t}. \tag{11}$$

Next we perform a projective integration step: we use the above estimate within any time-stepping method (like forward Euler or Runge–Kutta), to march U forward in (long) time, and in particular, use a large time step $\Delta t \gg \delta t$ to approximate the new macro-state $U_i^{t+\Delta t}$, $i = 1, 2, \ldots, N$. The patch dynamics scheme, effectively, requires us to simulate the detailed problem only in a fraction of space (i.e., teeth and buffer) and a fraction of time (i.e., brief sequences of short time steps for computing the gap-tooth estimate (11)). Samaey et al. [27] discussed in detail this patch dynamics algorithm for homogenization including its convergence and choice of boundary conditions.

4   Learning of effective PDEs from data

The key contribution of this work is exploiting the equation-free approach for parsimonious generation of training data in learning effective homogenized PDEs. In general, although homogenized equations may exist for heterogeneous materials, there may be no systematic way to analytically discover them (for example in the case of non-periodic heterogeneity [23]). In such cases, one seems restricted to fully simulating the detailed problem, and then *performing coarse-grained observations of it* (e.g., via averaging or low-pass filtering) to obtain training data.

Here, we use the equation-free approach as the alternative, as it only requires simulation of the detailed problem only in a fraction of the total space and time domain. Figure 2 shows sample snapshots of data produced in an example by solving the homogenized PDE, the detailed PDE, and

the patch dynamics approximation of the effective PDE for the one-dimensional diffusion problem in Section 5.1. In this example, the patch dynamics allows us to simulate the detailed problem in less than 8% of the space and 0.1% of time domain, effectively reducing the space-time grid size by a factor of 10^4. We now use the data so produced to train neural nets that learn the time-evolution (i.e. the effective operator) of the effective (homogenized) system. To validate the approach, we evaluate the performance of the trained neural nets via comparison with the homogenized equations which *are known in closed form* for our test examples. We remind the reader that such closures are, in general, not available.

We consider two neural network learning approaches for discovering the homogenized PDE from the data. In the first architecture, we try to discover the *functional form* of the law of the PDE, i.e., the relation between spatial and temporal derivatives of the coarse field at every macro-scale space-time point:

$$\partial_t U = F\big(U, \partial_x U, \partial_{xx} U, \ldots\big). \tag{12}$$

The particular combination of derivatives required for successful learning of the homogenized equation may be unknown *a priori.* In previous work [18], we successfully utilized learning approaches such as diffusion maps and Gaussian process regression to identify a minimal subset of spatial derivatives that suffices to learn an effective model. Assuming that we have included a sufficient set of spatial derivative terms as input, the learning problem amounts to regressing the function F on its arguments (i.e., learning the right-hand-side differential operator [34, 35]) from data.

In the second approach, we model the law of the PDE directly in a *discretized* form: as a relation between the time derivative of the field at a point of interest and the values of the field at a number of points in the neighborhood of the point of interest. This can be thought of as a form of spatial discretization of the effective PDE for the macro-scale variable

$$\frac{d}{dt}U(x_j) = G\big(U(x_j), U(x_{j-1}), U(x_{j+1}), \ldots\big). \tag{13}$$

where x_j's denote points on a spatial grid around the point of interest ordered by the index j. The stencil for this learning problem should be large enough to contain sufficient information for successfully approximating the local dU/dt.

The above two architectures appear very similar: they employ the local profile in the neighborhood of the grid point of interest either (a) in the form of observed/estimated spatial derivative values *at that point*, or (b) in the form of a grid of neighboring field values, in order to approximate time evolution. Both approaches are capable of learning PDEs with spatially homogeneous character with various terms such as diffusion, reaction, linear or nonlinear advection and source or sink terms. The two approaches can be formally related to each other when the observed dynamics (possibly after fast initial transients) lie on a low-dimensional manifold, for example on a finite-dimensional inertial manifold [36, 37]. Theoretical justification for the second approach can also be provided through the center-manifold-based work in [38] towards the "holistic discretization", which has been taken advantage of in enhancing patch dynamics simulations [39, 22].

A practical advantage of the second approach is that it avoids the "off-line" estimation of spatial derivatives, which could be sensitive to noise and computational directional preferences (e.g., upwinding for advection problems); the obvious price to pay—along with the loss of translational invariance due to discretization—is the dependence of the learned network parameters on the particularly selected grid. Therefore, the learned model has to be retrained or adjusted for new spatial grids in the second architecture. This second approach has already been utilized in learning data-driven discretization schemes [e.g. 13].

We use the neural network architectures shown in Figure 3(a) and (b), respectively, to regress F and G. The first network consists of three fully connected layers with a Rectifying Linear Unit (ReLU)
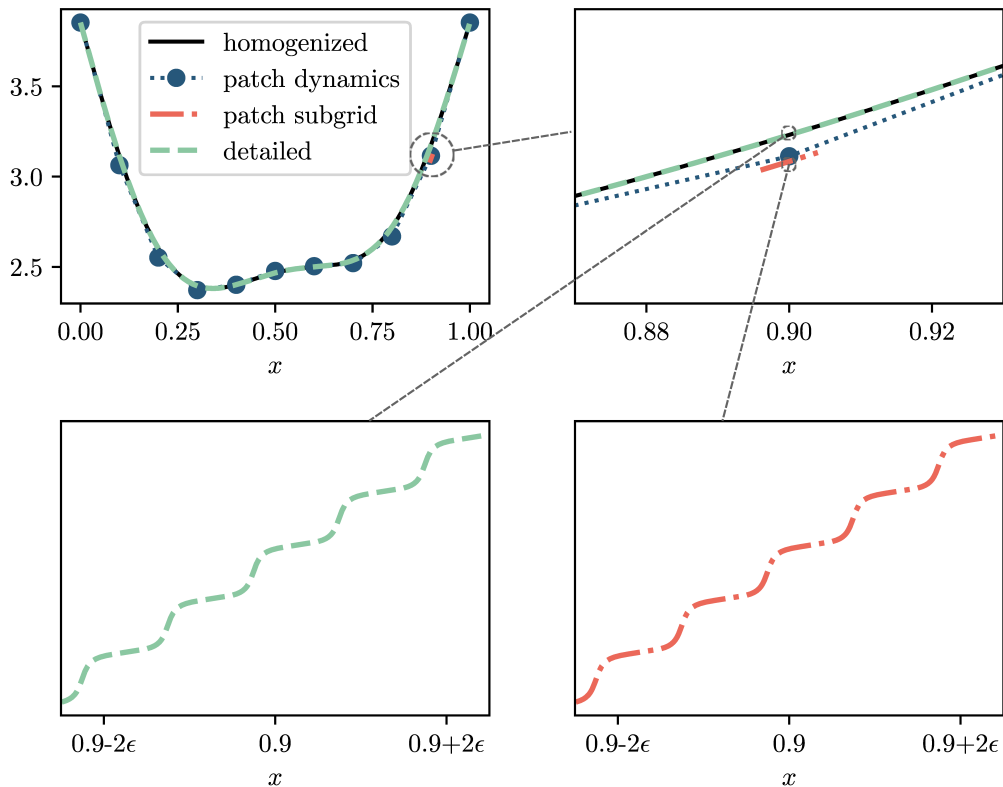
Figure 2: **A sample snapshot from different computational implementations of the one-dimensional heterogeneous diffusion problem in Section 5.1.** The top left panel demonstrates the agreement between the fully resolved, the homogenized, and the patch dynamics solution observed at large-scales. The bottom two panels show magnifications revealing the ϵ-scale variations of the solution for the fully resolved and the patch dynamics simulation ($\epsilon = 10^{-5}$). Instead of the fully resolved simulation, we use a patch dynamics sparse simulation ($10^4\times$ smaller space-time grid) to collect data for learning the effective dynamics. This particular snapshot is produced from an initial condition of the form (23) after running the simulations for 0.05 time units.

10

as the activation function in the first two layers. Recall that the action of a fully connected linear layer can be written as

$$z^{out} = A z^{in} + b, \quad b, z^{out} \in \mathbb{R}^n, \ z^{in} \in \mathbb{R}^m, \ A \in \mathbb{R}^{n \times m}. \tag{14}$$

On the other hand the action of ReLU layer is given by

$$z^{out} = \max(0, z^{in}). \tag{15}$$

Therefore our network represents a function of the form

$$z^{out} = A^{(3)} \max \left(A^{(2)} \max \left(A^{(1)} z^{in} + b^{(1)}, 0 \right) + b^{(2)}, 0 \right) + b^{(3)} \tag{16}$$

where the superscripts are the index of the fully connected layer. Entries of the matrices $A^{(i)}$, $i = 1, 2, 3$ and the bias vectors $b^{(i)}$, $i = 1, 2, 3$ are the parameters of the network which are to be optimized in the learning process.

The second network consists of three convolutional layers, with ReLU activation layers in between. Let us recall the mathematical description of a (linear) convolution layer action. Assume each layer consists of n_k kernels of size $2n + 1$. Let C_{lpq} denote the q-th kernel in the layer. Then the action of the layer on an input with p channels is given by

$$z^{out}_{iq} = \sum_p \sum_{j=-n}^{n} C_{jpq} z^{in}_{i+jp} + b_q, \quad q = 1, \ldots, n_k, \tag{17}$$

which, in the shorthand form is written as

$$z^{out} = C * z^{in}. \tag{18}$$

As such, our network represents the function

$$z^{out} = C^{(3)} * \left(C^{(2)} * \max(C^{(1)} * z^{in} + b^{(1)}, 0) + b^{(2)}, 0 \right) + b^{(3)} \tag{19}$$

where the superscripts are the index of the convolutional layer in the network. The entries of the tensors $C^{(i)}$, $i = 1, 2, 3$ and vectors $b^{(i)}$, $i = 1, 2, 3$ are to be learned from data. The first convolutional layer, with a sufficiently large kernel size, effectively acts as a finite-difference model of the spatial derivatives. Here, we set the kernel size of the next two layers to be one, which makes them, in effect, fully connected layers. This general type of shallow architecture has been demonstrated to suffice for capturing the nonlinearity of PDEs arising in fluid mechanics [13].
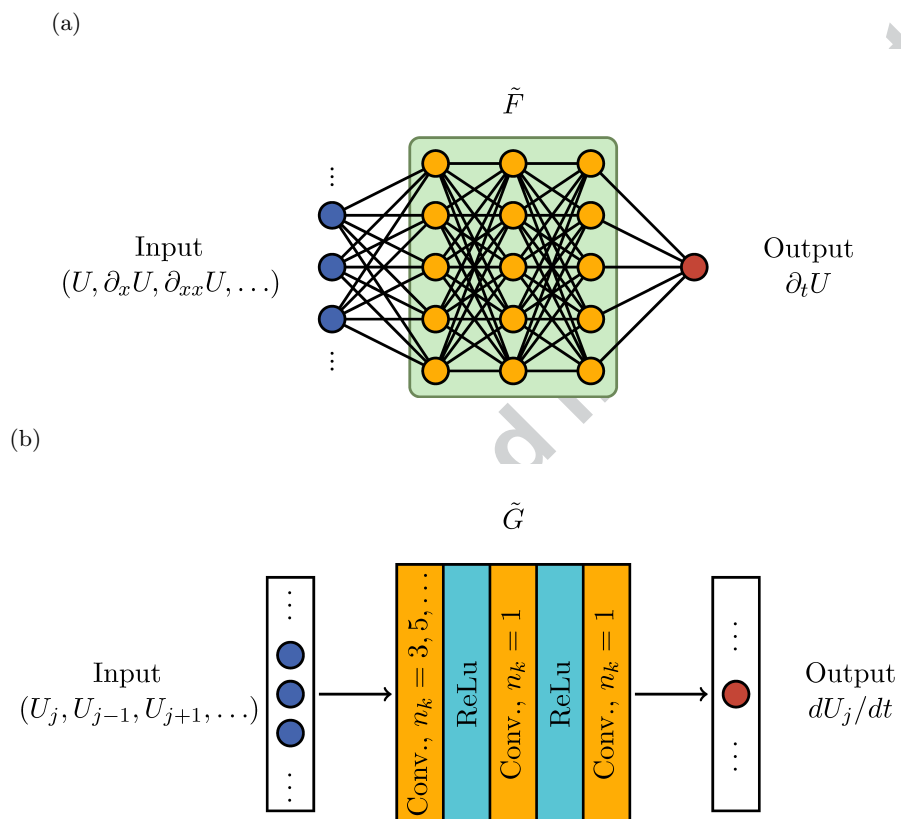
11

(a)

Figure 3: **Neural network architectures for approximate PDE discovery.** (a) A neural net architecture for learning the functional form of the PDE differential operator: there are three fully connected layers with 32 nodes and the first two layers have ReLU nonlinearity. (b) A neural net architecture for learning the PDE in its discretized form: there are three convolutional layers with 32 filters and ReLU activations in between. The last two convolution layers have kernels of size one and, in effect, act as fully connected layers.

12

5 Results

5.1 One-dimensional heterogeneous diffusion

In the first example, we consider the one-dimensional heterogeneous diffusion problem from [27]. The detailed evolution is governed by the PDE

$$\partial_t u(x,t) = \partial_x \left[a\!\left(\frac{x}{\epsilon}\right) \partial_x u(x,t) \right], \quad a\!\left(\frac{x}{\epsilon}\right) = 1.1 + \sin\left(2\pi\frac{x}{\epsilon}\right) \tag{20}$$

on the domain $[0,1]$ with $\epsilon = 10^{-5}$ and time-independent Dirichlet boundary conditions of

$$u(0,t) = u(0,0), \quad u(1,t) = u(1,0). \tag{21}$$

The corresponding homogenized equation is known to be

$$\partial_t u_h(x,t) = \partial_x \left[a^* \partial_x u_h(x,t) \right], \quad a^* \approx 0.45825686, \tag{22}$$

with similar boundary conditions.

To generate the training data, we use the patch dynamics scheme for (20) using 10 uniformly spaced teeth in the spatial domain ($\Delta x = 0.1$). Each tooth has a width of $h = 10^{-4}$ and it is surrounded by a buffer zone of width $H = 8 \times 10^{-3}$. The detailed simulation within each tooth and its buffer is carried out using a central finite-difference scheme with spatial resolution of $\delta x = 10^{-7}$ and time step size of $\delta t = 10^{-6}$ with Dirichlet boundary conditions imposed on the buffer zone boundaries. The macroscopic time step for advancing the macro-state variable is $\Delta t = 10^{-3}$. Therefore, we are only solving the detailed problem in 8% of the spatial domain and 0.1% of the time domain. Figure 2 shows sample snapshots of u for this problem arising during the computation using various alternative methods.

We simulate 8 trajectories of the system with initial conditions of the form

$$u(x,0) = \sum_{j=1}^{20} a_j \sin(2\pi l_j x + \phi_j) \tag{23}$$

where a_j, l_j and ϕ_j are randomly drawn from uniform distributions on $[-1,1]$, $[0,4]$ and $[0,2\pi]$, respectively. We simulate each trajectory over the time interval of $[0,1]$ and record snapshots of u and $\partial_t u$ every $\delta t_s = 10^{-3}$. We train both neural net architectures using the ADAM optimizer [40] with batch sizes of 64 and learning rate of 10^{-3}. For the optimization objective, we use the Mean Squared Error (MSE) of the network output (i.e., $\partial_t u$). In this example, we use a second-order finite-difference method with stencil size of three to estimate the spatial derivatives required as inputs by the first architecture, and a corresponding kernel size of three in the first convolutional layer of the second architecture.

To validate the learned models, we compare their performance to that of the known homogenized model in two test trajectories that have initial conditions of the form (23) but have not been used in the training. The (converged) homogenized solution to which we compare is computed using a second-order finite-difference discretization of (22) on a grid size of $dx = 5 \times 10^{-3}$ and `lsode` time stepper. As the error measure, we use the relative Mean Squared Error (rMSE) between the two snapshots, defined as

$$\text{rMSE}(v) = \frac{\sum_j \| v^j - v_h^j \|^2}{\sum_j \| v_h^j - \overline{v_h} \|^2} \tag{24}$$
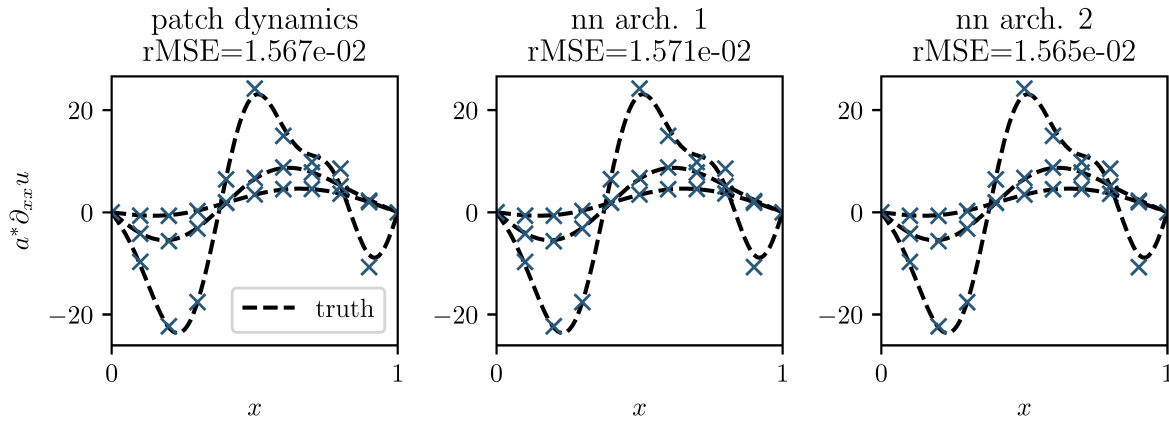
13

Figure 4: **Prediction of the RHS of an effective PDE via patch dynamics and neural nets**. "Truth" here denotes the solution of the homogenized equation on a fine grid.

where v^j denotes the jth snapshot predicted by the neural net and v_h^j is the associated snapshot from the homogenized model. As shown in Figure 4, both architectures are quite successful in accurately representing the RHS of the homogenized PDE, even though the training data is generated on a much coarser grid than the fully resolved data.

Next, we use the learned models to simulate a trajectory with a new initial condition from (23). We use the BDF method [41] as the time stepper on the same macroscopic grid. Comparison to the solution of the homogenized PDE (right column of Figure 5), suggests that the learned models exhibit, in addition to accuracy, the stability properties required for time integration of the effective equations.
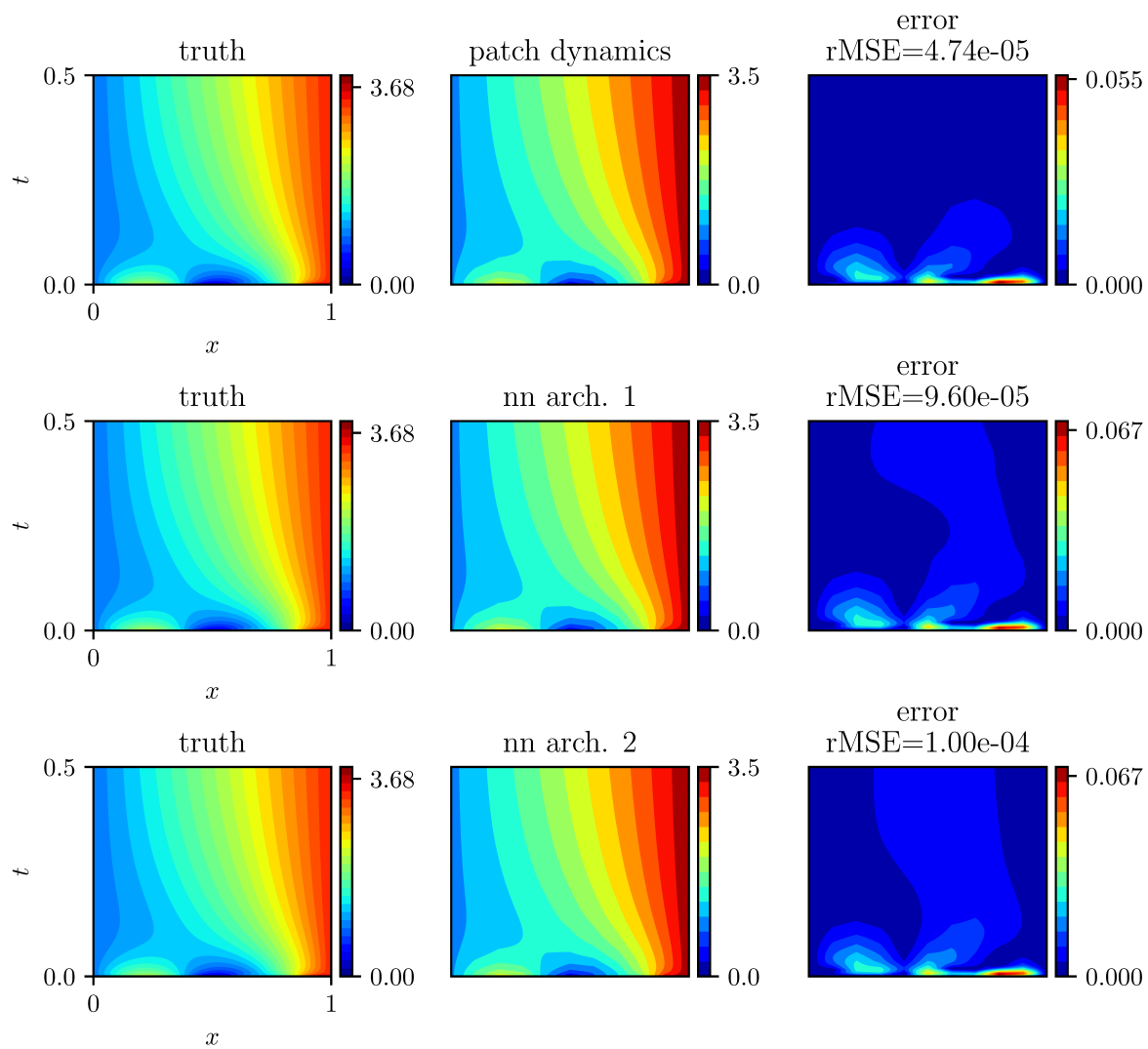
14

Figure 5: **Integration of a trajectory from an unseen initial condition via patch dynamics and neural nets**. "Truth" here denotes the solution of the homogenized equation on a sufficiently resolved grid.

5.2 Diffusion in two dimensions with lattice heterogeneity

In this example [31, 29], we consider diffusion on a two-dimensional periodic domain with lattice-type micro-scale heterogeneity. We define a two-dimensional lattice, indexed by (i, j), with the same spacing h in both directions on the bi-periodic domain $[0, 2\pi)^2$. The evolution of the micro-scale field $u_{i,j}$ is given by

$$\dot{u}_{i,j} = \left[\kappa_{i,j}^x(u_{i+1,j} - u_{i,j}) + \kappa_{i-1,j}^x(u_{i-1,j} - u_{i,j}) + \kappa_{i,j}^y(u_{i,j+1} - u_{i,j}) + \kappa_{i,j-1}^y(u_{i,j-1} - u_{i,j}) \right]/h^2 \tag{25}$$

where $\kappa_{i-1,j}^x$ is the diffusivity at half lattice point $(i+1/2, j)$ and $\kappa_{i-1,j}^y$ is the diffusivity at half lattice point $(i, j + 1/2)$. We assume that the heterogeneous diffusivity is 3-periodic in both directions and, in particular, given as

$$\kappa^x = \begin{bmatrix} 1.0566 & 0.6668 & 1.1568 \\ 6.5894 & 0.8683 & 2.4174 \\ 0.9473 & 1.1407 & 1.6610 \end{bmatrix}, \quad \kappa^y = \begin{bmatrix} 3.6355 & 0.4470 & 2.3896 \\ 0.8628 & 4.8558 & 0.2833 \\ 4.5025 & 1.5865 & 0.5679 \end{bmatrix}. \tag{26}$$

It is known [31] that there exists a homogenized equation, namely,

$$\partial_t u_h = 1.2644 \partial_{xx} u_h + 1.3398 \partial_{yy} u_h \tag{27}$$

which provides an approximation to the evolution of u at large length scales.

We use the gap-tooth scheme for parsimoniously simulating the micro-scale system. We use a uniformly spaced 16×16 grid of square teeth, each one with sides of $H = 4h = 0.0524$. These spatial patches approximately cover 1% of the space. We solve (25) for the micro-scaled grid points that are contained within our spatial patches under the appropriate boundary conditions for each patch. The macro-scale solution is then computed by averaging u on the interior points of the patch. Here, the simulation is performed for all times (no projective integration "jumps" are used); in equation-free terms this is a gap-tooth scheme, and the savings come only from the spatial sparsity of the simulation. The spatial patch boundary conditions are carefully chosen to couple the macro-scale solution on neighboring patches and approximately retrieve the global coarse pattern that would appear from simulating the fully resolved micro-scale system, as detailed elsewhere [31].

We simulate 100 trajectories where the initial conditions are given by

$$u(x, 0) = \sum_{j=1}^{10} a_j \sin(l_j^x x + \phi_j^x) \sin(l_j^y y + \phi_j^y) \tag{28}$$

and the parameters $a_j \in [-1, 1]$, $(l_j^x, l_j^y) \in \{1, 2, 3, 4, 5\}^2$, and $(\phi_j^x, \phi_j^y) \in [0, 2\pi]^2$ are drawn randomly from uniform distributions on their domain. We simulate the trajectories over $t \in [0, 1]$ and collect 100 snapshots of each at sampling intervals of 0.01. We use data from 85 trajectories (i.e., 8500 snapshots) to train our networks. The training parameters are chosen similar to the one-dimensional problem. In this example, we use FFT to estimate the spatial derivatives for the first neural net architecture, thus improving the estimates of the input to the neural net.

In this example our scheme, trained with gap-tooth simulation data averaged over a sparse grid of teeth, is less successful in providing an accurate closed evolution equation. Our neural networks (with various depths) cannot drive the MSE significantly below a few percent. We believe that this shortcoming can be rationalized: Figure 6 shows sample snapshots of $\partial_t u$ from the homogenized solution, which we take as the truth here, and the prediction error of our best trained neural networks. All the trajectories in this example converge to steady state where $\partial_t u = 0$. We observe that
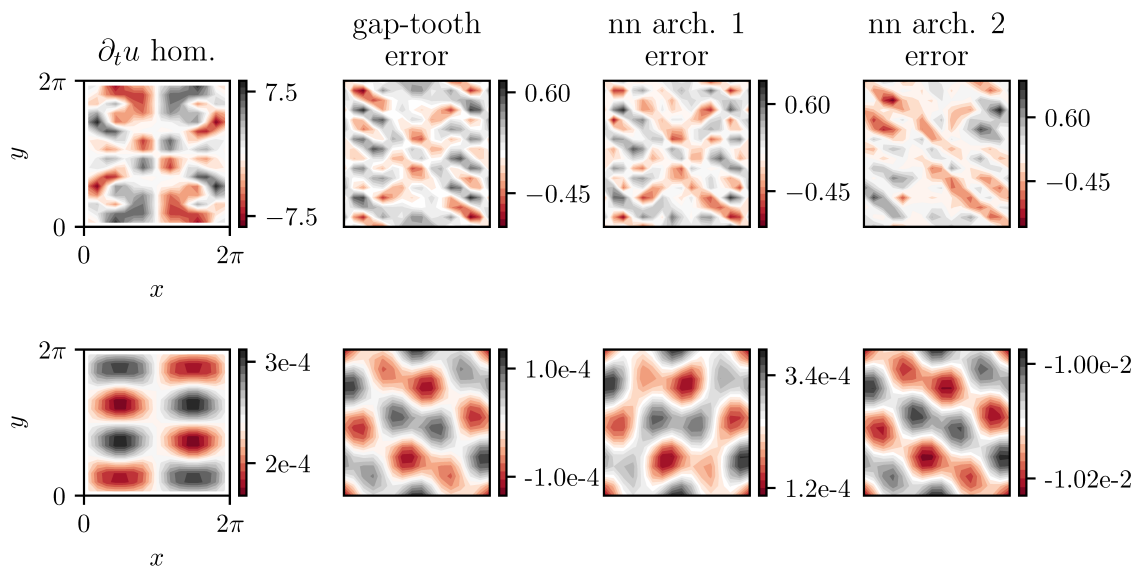
16

Figure 6: **Performance of models in estimating 2D PDE's right-hand-side.** The relative error of prediction is small for snapshots with large magnitude of $\partial_t u$ (e.g., first row) and large for small magnitudes of $\partial_t u$ (e.g., bottom row).

the relative prediction error becomes typically larger for snapshots with smaller magnitude of $\partial_t u$. A rationalization of this observation is that the choice of loss function (e.g., MSE in this work) emphasizes accurate prediction *for large-magnitude snapshots*, and that for the small-amplitude data used here, it might be expected that the relative error would then be large. Therefore, the learned estimation of the time-derivative near the steady state is less accurate than estimates for states away from steady state.

Figure 7 shows sample snapshots of a trajectory integrated using the PDEs discovered by the neural nets and their comparison with the corresponding homogenized solution. The neural net solutions closely follow the homogenized solution early in the integration interval (at large solution amplitudes) and their error is comparable to that of the training data (i.e., gap-tooth versus homogenized). However, as the true trajectory approaches the steady state, and the solution amplitude decays, the relative error of the learned models grows—the time derivative is less accurately estimated in that region, and that error accumulates in the trajectory over time (Figure 8(a)). Interestingly, however, *the qualitative state-space behavior of the discovered models is visibly accurately recovered, and close to truth:* Figure 8(b) shows a projection of the solution trajectories from the homogenized solution and from the neural-nets-identified PDE in the solution state space. This projection is realized by observing the six largest coefficients of the Fourier expansion for the solutions. The trajectories from the learned models, regardless of time parametrization, are visibly close to the homogenized solution trajectory, and only start to deviate in a small neighborhood of the origin. We then credibly argue that the overall identification process is successful: the trajectories of the learned models quantitatively shadow the true ones to the neighborhood of the final steady state, and even there they have the right long term/stability characteristics.
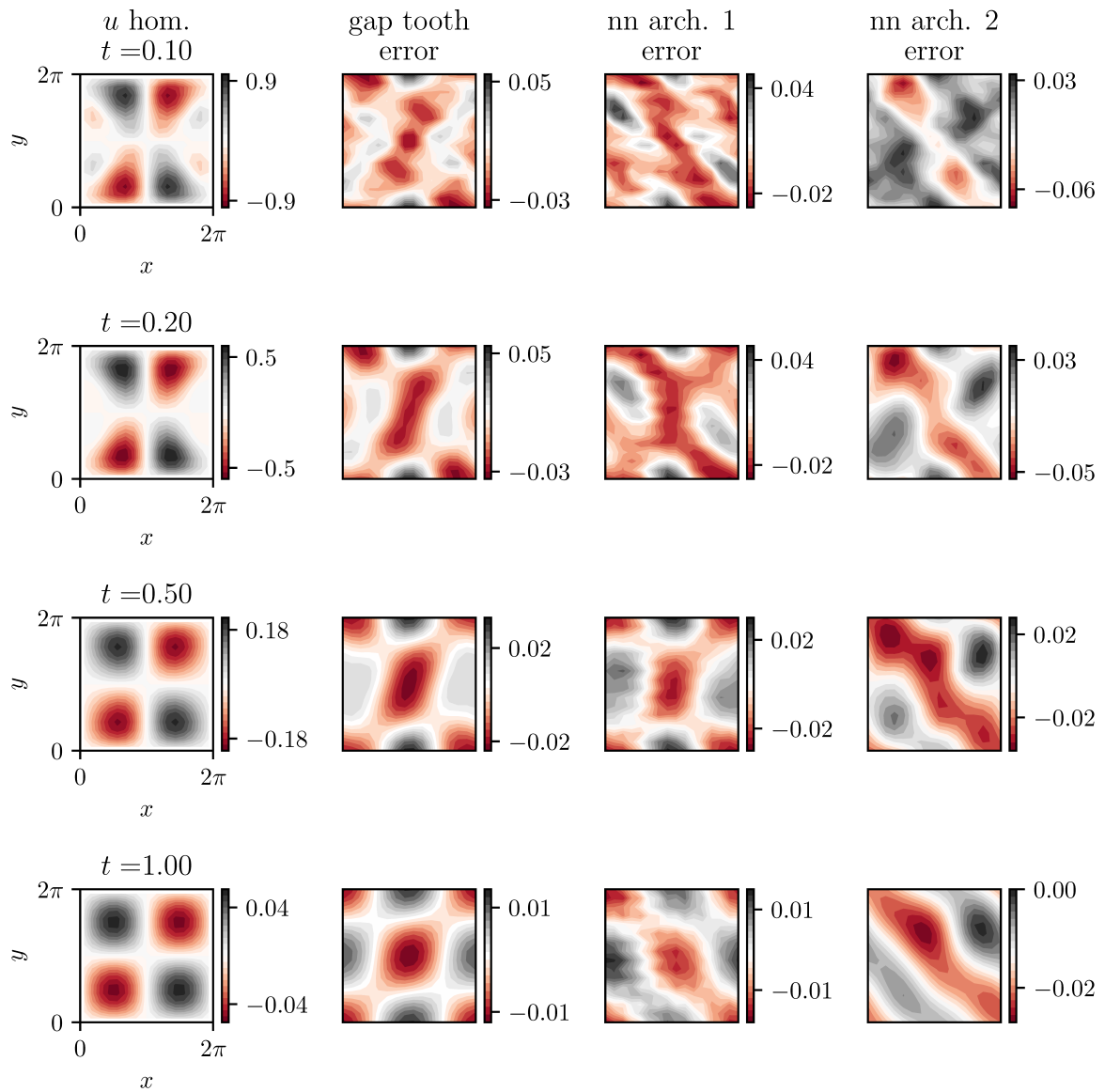
17

Figure 7: **Performance of learned models in integrating a test trajectory.** The neural net models (trained via gap-tooth data) follow the homogenized solution closely at the beginning but yet lag behind the final decay to steady state.
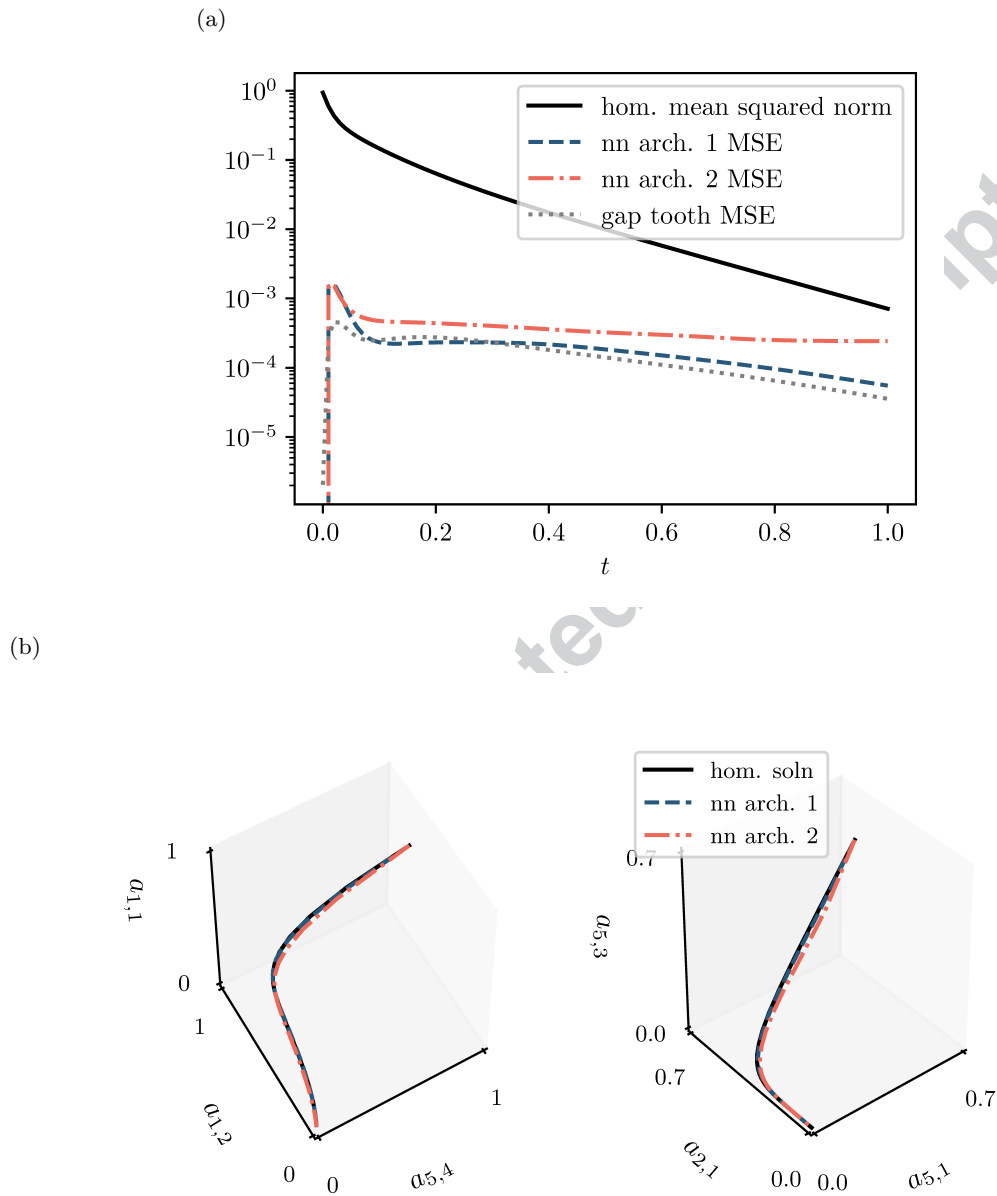
(a)

(b)

Figure 8: **Qualitative behavior of learned models** (a) The mean squared error (MSE) of test trajectories and (b) a state space viewpoint showing the six largest Fourier amplitudes of the solution trajectory.

19

6   Conclusion

Machine learning techniques have proven successful in learning PDEs that approximate the observed evolution of complex systems from spatiotemporal data. Despite this initial success, an outstanding challenge for identifying models of the effective dynamics of multiscale/multiphysics systems is the computational burden of generating the training data from detailed computational models embodying our best understanding of the fine-scale physics. In this work, we proposed linking established multiscale scientific computing algorithms (and, in particular, using the equation-free approach, with modern neural network algorithms for learning effective PDEs. We illustrated the combination of these two "computational technologies" for learning homogenized equations for materials with micro-scale property heterogeneities, enabled by parsimonious training data collection. The advantage of our approach lies precisely in the efficient compilation of training data: the equation-free approach (and other similar approaches [e.g. 42]) requires simulation of the detailed problem in only a small fraction of the space-time domain and dramatically reduces the amount of computational effort for collecting the requisite training data. We demonstrated our approach in the case of heterogeneous diffusion and validated our results using the analytically known homogenized solutions. Our approach can be readily extended to problems with more complex material behavior (non-periodic heterogeneity and drift terms) where the there are no established methods to derive closed-form solutions (assuming such equations exist).

Throughout this paper, we assumed the appropriate macro-scale field variable is known. Indeed, guided by the classical homogenization theory, we took the target macro-scale variable (i.e., homogenized equation field) to be a low-pass filtered version of the micro-scale variable (the fully resolved solution), and used the averaging-over-tooth operation within the equation-free approach to transform fine-scale data to training data for the macroscopic model (what in equation-free terminology is called "the restriction operator"). An important research direction involves the data-driven identification of the right observables, i.e. the variables in terms of which the PDE to be learned should be formulated. Whether through manifold learning techniques, like diffusion maps (e.g., in the "variable free" computational efforts) [43, 44, 45] or through deep-learning tools like (possibly variational) auto-encoders, and Wasserstein GANs [46, 2, 47], this possibility will, we expect, greatly enhance the range of phenomena for which useful data-driven models can be derived. Physically informing such models [e.g. 15] and providing explainable descriptions of their predictions is a complementary, but equally important, challenge. One can even contemplate the case of "emergent PDEs", where not only the dependent variables and the operators, but even the independent variables themselves (the right "space" and even the right "time" for the PDE) are obtained in a data-driven manner [48, 49].

Acknowledgements

Data and source code

A python Implementation of our framework and the data for producing the figures in this paper is available at https://github.com/arbabiha/homogenization_via_ML.

References

[1] Xuan Liu, David Furrer, Jared Kosters, and Jack Holmes. *Vision 2040: A Roadmap for Integrated, Multiscale Modeling and Simulation of Materials and Systems*. Tech. rep. NASA STI Program, Mar. 2018. URL: https://ntrs.nasa.gov/search.jsp?R=20180002010.

[2] R Rico-Martinez, K Krischer, IG Kevrekidis, MC Kube, and JL Hudson. "Discrete-vs. continuous-time nonlinear signal processing of Cu electrodissolution data". *Chemical Engineering Communications* 118.1 (1992), pp. 25–48.

[3] R Rico-Martinez, IG Kevrekidis, and K Krischer. "Nonlinear system identification using neural networks: dynamics and instabilities". *Neural networks for chemical engineers* (1995), pp. 409–442.

[4] R Gonzalez-Garcia, R Rico-Martinez, and IG Kevrekidis. "Identification of distributed parameter systems: A neural net based approach". *Computers & Chemical Engineering* 22 (1998), S965–S968.

[5] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Data-driven discovery of partial differential equations". *Science Advances* 3.4 (2017), e1602614.

[6] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. "Numerical Gaussian processes for time-dependent and nonlinear partial differential equations". *SIAM Journal on Scientific Computing* 40.1 (2018), A172–A198.

[7] Maziar Raissi and George Em Karniadakis. "Hidden physics models: Machine learning of nonlinear partial differential equations". *Journal of Computational Physics* 357 (2018), pp. 125–141.

[8] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach". *Physical review letters* 120.2 (2018), p. 024102.

[9] Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213 (2018), p. 20170844.

[10] Zhenlin Wang, Xun Huan, and Krishna Garikipati. "Variational system identification of the partial differential equations governing the physics of pattern-formation: inference under varying fidelity and noise". *Computer Methods in Applied Mechanics and Engineering* 356 (2019), pp. 44–74.

[11] Dimitris C Psichogios and Lyle H Ungar. "A hybrid neural network-first principles approach to process modeling". *AIChE Journal* 38.10 (1992), pp. 1499–1511.

[12] R Rico-Martinez, JS Anderson, and IG Kevrekidis. "Continuous-time nonlinear signal processing: a neural network based approach for gray box identification". *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*. IEEE. 1994, pp. 596–605.

[13] Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. "Learning data-driven discretizations for partial differential equations". *Proceedings of the National Academy of Sciences* 116.31 (2019), pp. 15344–15349.

[14] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. "Artificial neural networks for solving ordinary and partial differential equations". *IEEE transactions on neural networks* 9.5 (1998), pp. 987–1000.

[15] Maziar Raissi, Paris Perdikaris, and George 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 (2019), pp. 686–707.

[16] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. "Variational physics-informed neural networks for solving partial differential equations". *arXiv preprint arXiv:1912.00873* (2019).

[17] Xuhui Meng, Zhen Li, Dongkun Zhang, and George Em Karniadakis. "PPINN: Parareal physics-informed neural network for time-dependent PDEs". *Computer Methods in Applied Mechanics and Engineering* 370 (2020), p. 113250.

[18] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I Siettos, and Ioannis G Kevrekidis. "Coarse-scale PDEs from fine-scale observations via machine learning". *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.1 (2020), p. 013141.

[19] Alain Bensoussan, Jacques Louis Lions, and George Papanicolaou. *Asymptotic analysis for periodic structures*. Vol. 374. American Mathematical Soc., 2011.

[20] J. E. Bunder and A. J. Roberts. *Nonlinear emergent macroscale PDEs, with error bound, for nonlinear microscale systems*. Tech. rep. [https://arxiv.org/abs/1806.10297], June 2018.

[21] Ioannis G Kevrekidis, C William Gear, James M Hyman, Panagiotis G Kevrekidid, Olof Runborg, Constantinos Theodoropoulos, et al. "Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis". *Communications in Mathematical Sciences* 1.4 (2003), pp. 715–762.

[22] Giovanni Samaey, Anthony J Roberts, and Ioannis G Kevrekidis. "Equation-free computation: an overview of patch dynamics". *Multiscale methods: bridging the scales in science and engineering* (2009), p. 216.

[23] Grigorios A Pavliotis and Andrew Stuart. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.

[24] A. J. Roberts. "Macroscale, slowly varying, models emerge from the microscale dynamics in long thin domains". *IMA Journal of Applied Mathematics* 80.5 (2015), pp. 1492–1518. DOI: 10.1093/imamat/hxv004.

[25] Olof Runborg, Constantinos Theodoropoulos, and Ioannis G Kevrekidis. "Effective bifurcation analysis: a time-stepper-based approach". *Nonlinearity* 15.2 (2002), p. 491.

[26] Giovanni Samaey, Dirk Roose, and Ioannis G Kevrekidis. "The gap-tooth scheme for homogenization problems". *Multiscale Modeling & Simulation* 4.1 (2005), pp. 278–306.

[27] Giovanni Samaey, Ioannis G Kevrekidis, and Dirk Roose. "Patch dynamics with buffers for homogenization problems". *Journal of Computational Physics* 213.1 (2006), pp. 264–287.

[28] Anthony J Roberts and Ioannis G Kevrekidis. "General tooth boundary conditions for equation free modeling". *SIAM Journal on Scientific Computing* 29.4 (2007), pp. 1495–1510.

[29] J. E. Bunder, I. G. Kevrekidis, and A. J. Roberts. *Equation-free patch scheme for efficient computational homogenisation via self-adjoint coupling*. Tech. rep. http://arxiv.org/abs/2007.06815, July 2020.

[30] A.J. Roberts. *Multiscale modelling of microscale heterogeneous systems: analysis supports systematic and efficient macroscale modelling and simulation*. Tech. rep. AMSI Winter School on Computational Modeling of Heterogeneous Media [https://arxiv.org/abs/1907.09685], July 2019.

[31] JE Bunder, Anthony J Roberts, and Ioannis G Kevrekidis. "Good coupling for the multiscale patch scheme on systems with microscale heterogeneity". *Journal of Computational Physics* 337 (2017), pp. 154–174.

[32] C. William Gear and Ioannis G. Kevrekidis. "Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum". *SIAM Journal on Scientific Computing* 24.4 (2003), pp. 1091–1106.

[33] C. William Gear, Ioannis G. Kevrekidis, and Constantinos Theodoropoulos. "'Coarse' integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods". *Computers & Chemical Engineering* 26.7-8 (2002), pp. 941–963.

[34] Tianping Chen and Hong Chen. "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems". *IEEE Transactions on Neural Networks* 6.4 (1995), pp. 911–917.

[35] 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).

[36] Michael S Jolly, IG Kevrekidis, and Edriss S Titi. "Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: analysis and computations". *Physica D: Nonlinear Phenomena* 44.1-2 (1990), pp. 38–60.

[37] Peter Constantin, Ciprian Foias, Basil Nicolaenko, and Roger Temam. *Integral manifolds and inertial manifolds for dissipative partial differential equations*. Vol. 70. Springer Science & Business Media, 2012.

[38] AJ Roberts. "Holistic discretization ensures fidelity to Burgers' equation". *Applied Numerical Mathematics* 37.3 (2001), pp. 371–396.

[39] JE Bunder, Anthony J Roberts, and Ioannis G Kevrekidis. "Accuracy of patch dynamics with mesoscale temporal coupling for efficient massively parallel simulations". *SIAM Journal on Scientific Computing* 38.4 (2016), pp. C335–C371.

[40] Diederik P Kingma and Jimmy Ba. "ADAM: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014).

[41] Lawrence F Shampine and Mark W Reichelt. "The MATLAB ODE suite". *SIAM Journal on Scientific Computing* 18.1 (1997), pp. 1–22.

[42] Weinan E, Bjorn Engquist, Xiantao Li, Weiqing Ren, and Eric Vanden-Eijnden. "Heterogeneous multiscale methods: a review". *Communications in Computational Physics* 2.3 (2007), pp. 367–450.

[43] Radek Erban, Thomas A Frewen, Xiao Wang, Timothy C Elston, Ronald Coifman, Boaz Nadler, and Ioannis G Kevrekidis. "Variable-free exploration of stochastic models: a gene regulatory network example". *The Journal of Chemical Physics* 126.15 (2007), 04B618.

[44] Eliodoro Chiavazzo, Roberto Covino, Ronald R Coifman, C William Gear, Anastasia S Georgiou, Gerhard Hummer, and Ioannis G Kevrekidis. "Intrinsic map dynamics exploration for uncharted effective free-energy landscapes". *Proceedings of the National Academy of Sciences* 114.28 (2017), E5494–E5503.

[45] Ping Liu, CI Siettos, CW Gear, and IG Kevrekidis. "Equation-free Model Reduction in Agent-based Computations: Coarse-grained Bifurcation and Variable-free Rare Event Analysis". *Mathematical Modelling of Natural Phenomena* 10.3 (2015), pp. 71–90.

[46] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". *arXiv preprint arXiv:1701.07875* (2017).

[47] Diederik P Kingma and Max Welling. "Auto-encoding variational Bayes". *arXiv preprint arXiv:1312.6114* (2013).

[48] Felix P Kemeth, Sindre W Haugland, Felix Dietrich, Tom Bertalan, Kevin Hohlein, Qianxiao Li, Erik M Bollt, Ronen Talmon, Katharina Krischer, and Ioannis G Kevrekidis. "An emergent space for distributed data with hidden internal order through manifold learning". *IEEE Access* 6 (2018), pp. 77402–77413.

[49] Thomas N Thiem, Mahdi Kooshkbaghi, Tom Bertalan, Carlo R Laing, and Ioannis G Kevrekidis. "Emergent spaces for coupled oscillators". *Frontiers in Computational Neuroscience* 14 (2020), p. 36.