Discovering Reduced-Order Dynamical Models From Data

William Qian (bqqian@sas.upenn.edu)

Advisor: Pratik Chaudhari (pratikac@seas.upenn.edu)

April 27, 2022

Abstract

This work explores theoretical and computational principles for data-driven discovery of reduced-order models of physical phenomena. We begin by describing the theoretical underpinnings of multi-parameter models through the lens of information geometry. We then explore the behavior of paradigmatic models in statistical physics, including the diffusion equation and the Ising model. In particular, we explore how coarse-graining a system affects the local and global geometry of a "model manifold" which is the set of all models that could be fit using data from the system. We emphasize connections of this idea to ideas in machine learning. Finally, we employ coarse-graining techniques to discover partial differential equations from data. We extend the approach to modeling ensembles of microscopic observables, and attempt to learn the macroscopic dynamics underlying such systems. We conclude each analysis with a computational exploration of how the geometry of learned model manifolds changes as the observed data undergoes different levels of coarse-graining.

Contents

1 Introduction **3**

2 Overview of Information Geometry **4**

3 Information Geometry of Sloppy Models **7**

4 Coarse-Graining in Statistical Physics Models **9**
4.1 Coarse-graining the Ising Model 9
4.2 Coarse-Graining the Diffusion Model 11

5 System Identification: An Overview **16**
5.1 Sparse Regression 16
5.2 Physics-Informed Neural Networks 17

6 PDE System Identification **19**
6.1 General Methods 19
6.2 System identification of Partial Differential Equations 20
6.2.1 Burgers' equation 20
6.2.2 The Korteweg–De Vries equation 22
6.2.3 Perturbing PDEs with a diffusive term 23
6.3 Identifying macroscopic models from ensembles of microscopic observables 26
6.3.1 Reduced order Modeling of Van der Pol Ensembles 27
6.3.2 Reduced order Modelling of Lorenz Ensembles 30
6.3.3 Limitations of the advection-based approach 31

7 Conclusion **34**

1 Introduction

One problem that has garnered significant attention in recent years is that of predicting the dynamical equations that govern real-world data. While techniques for solving systems of differential equations numerically are well-established, this "inverse" problem of discovering the equations underlying data is significantly more challenging. Typical techniques for tackling this problem include symbolic regression, or neural network-based approaches[1, 2]. For example, previous work by Brunton et al. used sparse regression techniques to reconstruct the dynamical equations underlying chaotic systems such as the fluid flow of vortex shedding behind an obstacle [3].

However, instead of precisely learning the exact model of the dynamics of complex systems—which might be difficult—it may be sufficient to use lower-order approximations. As an example, suppose a scientist wants to understand if the salinity of the water near a coast affects the temperature of the ocean 10 kilometers away. Instead of modeling the dynamics of the ocean currents and salt concentration, a more macroscopic and coarse-grained model of ocean dynamics may be sufficient to answer such a question. Such a reduced-order model would also be more practical; we may need fewer data to fit or computation resources to build such a model.

Complex models of the dynamics that underlie real-world data are ubiquitous across biology, physics, and engineering. Often times, these models can be described as sloppy—namely, that only a few combinations of model parameters significantly affect model behavior [4, 5]. In such situations, complex multi-parameter models can be greatly reduced in size and still capture the dominant dynamics of the system. This work will demonstrate that the sloppiness of real-world systems is something that can be exploited to effectively construct reduced-order models of systems from data. To that end, we first reproduce the findings of Machta et al. and demonstrate that paradigmatic models from statistical physics can be compressed into smaller effective theories upon coarse-graining observable data [5]. Then, we apply concepts from information geometry to analyze how modern techniques for system identification of partial differential equations behave under coarse-graining. Finally, we develop techniques for learning reduced order models of large ensembles of microscopic observables.

2 Overview of Information Geometry

To motivate the methods that will be employed for the discovery of reduced order models from data, we begin by introducing ideas from information geometry. Information geometry has proven to be a powerful framework for analyzing the behavior of multi-parameter models [4, 5]. From models of complex biological systems with thousands of parameters to parameter-dense neural networks in machine learning, multi-parameter models across various domains, it is often of interest to compress multi-parameter models into smaller and more interpretable effective theories. Fortunately, for many such models, such compression of models is possible with little cost to overall model expressivity and accuracy [6]. These models are referred to as sloppy, and can be characterized by a wide hierarchy of model sensitivities to parameter combinations, whereby the least important parameter combinations can be varied drastically without significantly affecting model behavior. Information geometry offers a lens through which to understand the behavior of sloppy models, and how they might be reducible into simpler effective theories.

To illustrate the principles of information geometry in the context of multi-parameter models, it is useful to introduce the concept of a model manifold. Suppose a model $y_\theta(t)$ is parameterized by D parameters given by $\theta = \{\theta^\mu\}$, $1 \leq \mu \leq D$, and is evaluated at points t_i, $1 \leq i \leq M$. We can define a model manifold as

$$\mathcal{Y} = \{[y_\theta(t_1), y_\theta(t_2), \dots y_\theta(t_M)]|\theta\}$$

Each dimension corresponds to a prediction made by the model. In many cases, the number of model evaluations M is much larger than the number of parameters of the model. Therefore, the model manifold can often be thought of as embedding a manifold with a low intrinsic dimensionality (at most D) into a higher dimensional prediction space [4].

This framework yields new interpretations of classical problems such as fitting parameters to data under nonlinear least squares optimization. Suppose that we would like to find the parameters θ of a model y_θ such that the likelihood of observations $\{x_i\}_{i=1}^M$ being produced by the model is maximized. Assuming Gaussian measurement uncertainties $\{\sigma_i\}_{i=1}^M$, we have that

$$x_i = y_\theta(t_i) + \epsilon_i \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_i)$ is a normally distributed random variable. As described in [7], we can then define residual random variables $R_\theta(t_i)$ of the form

$$R_\theta(t_i) = \frac{y_\theta(t_i) - x_i}{\sigma_i} \tag{2}$$

These residuals are now standard normal random variables (zero mean, unit variance). Consequently, the likelihood of observing residuals $\vec{R}$ can be given by the multivariate Gaussian

$$P(\vec{R}|\theta) = \frac{1}{2\pi^{M/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{M} R_\theta(t_i)^2\right) \tag{3}$$

Maximizing the log-likelihood in turn yields the least squares cost function

$$C(\theta) = \sum_{i=1}^{M} R_\theta(t_i)^2 \tag{4}$$

to be minimized. Alternatively, Eqn. (4) could have been arrived at by considering the observations $\{x_i\}_{i=1}^M$ as a point $[x_1, x_2, \dots x_M]$ embedded in the same prediction space as the model manifold $\mathcal{Y}$, and finding the parameters θ that minimize the "distance" between this point and the model predictions $[y_\theta(t_1), y_\theta(t_2), \dots y_\theta(t_M)]$. In general, the precise way in which this distance should be defined depends on the likelihood distribution of observable data $p(\vec{x}|\theta)$. As demonstrated by this example, even models which are not explicitly stochastic in nature can often be framed as arising from a probabilistic model of the likelihood of observations under a given set of model parameters. This motivates the need to define a distance metric between different probabilistic models.

For a notion of distance to be useful in comparing two models with different sets of parameters, it would ideally be large between models that produce significantly different predictions, and small between models that produce near identical predictions. To that end, the Kullback-Liebler (KL) Divergence has commonly been used as a measure of distinguishibility between distributions [8]. Given two models with parameters θ and $\tilde{\theta}$, one can define the KL Divergence between their data likelihood distributions as follows:

$$D_{KL}[p(\vec{x}|\theta)||p(\vec{x}|\tilde{\theta})] = \sum_{\vec{x}} p(\vec{x}|\theta) \log\left(\frac{p(\vec{x}|\theta)}{p(\vec{x}|\tilde{\theta})}\right) \tag{5}$$

While the KL Divergence fails to be a proper distance metric due to its lack of exchange symmetry ($D_{KL}[p||q] \neq D_{KL}[q||p]$), the symmetric requirement can be recovered to first order when considering the KL Divergence produced by an infinitesimal change to the parameters $\tilde{\theta} = \theta + \delta\theta$, yielding [4]:

$$ds^2 = \sum_{\mu,\nu=1}^{D} g_{\mu\nu} d\theta^\mu d\theta^\nu \tag{6}$$

where

$$g_{\mu\nu} = -\sum_{\text{observables } \vec{x}} p(\vec{x}|\theta) \frac{\partial^2 p(\vec{x}|\theta)}{\partial\theta^\mu \partial\theta^\nu} \tag{7}$$

is the Fisher Information Matrix (FIM). This matrix can be viewed as describing the local geometry at a particular point on the model manifold.

In practice, $g_{\mu\nu}$ can be difficult to compute directly using Eqn. (7), as it requires estimating second-order derivatives of the data likelihood function with respect to model parameters. However, for many models, $g_{\mu\nu}$ can be expressed in terms of the Jacobian of model observations with respect to parameters. In particular, for nonlinear least squares models, where we assume Gaussian uncertainty in model estimates, this Jacobian is given by

$$J_{i\mu} = \frac{\partial}{\partial\theta^\mu} \frac{y_\theta(t_i)}{\sigma_i} \tag{8}$$

where θ_i represents the measurement uncertainty of the i^{th} observation.

Then, assuming diagonal measurement errors and that measurement uncertainties are parameter-independent, the Fisher Information Matrix can be expressed in terms of the Jacobian as [7]

$$g_{\mu\nu} = \sum_i J_{i\mu} J_{i\nu} = J^\top J \tag{9}$$

Indeed, modern packages for computing the Fisher Information Matrix of complex multi-parameter models such as neural networks often do so via first computing the model Jacobian and then applying Eqn. (9) [9].

3 Information Geometry of Sloppy Models

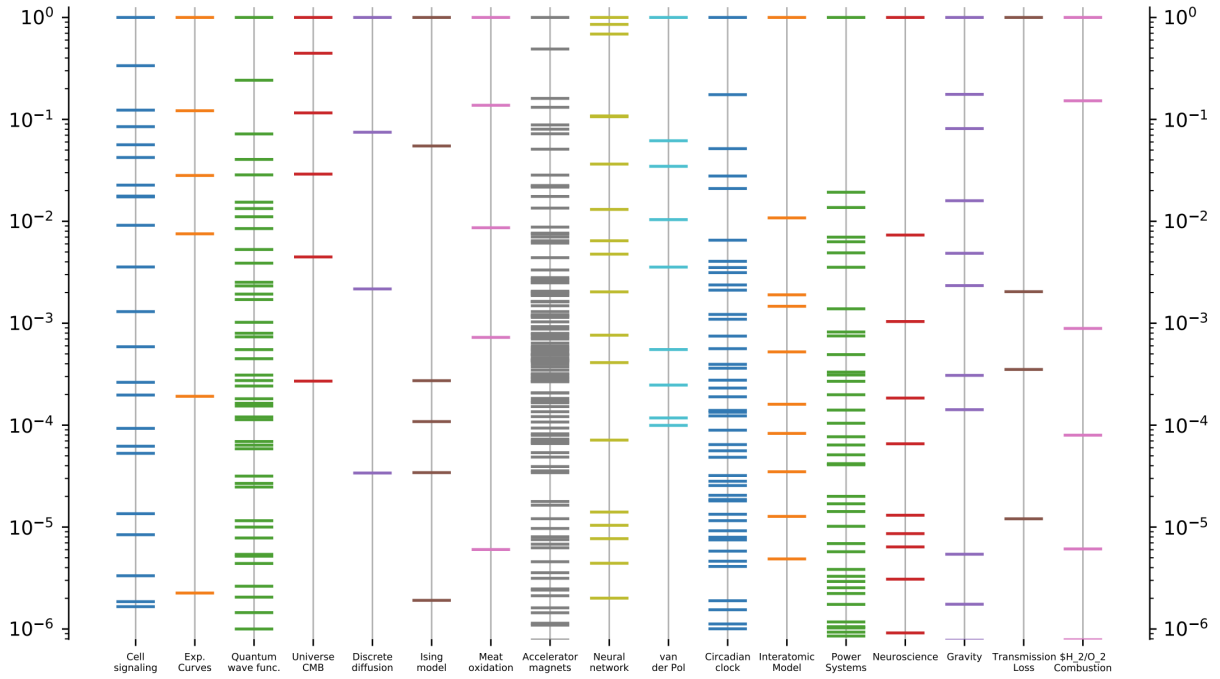

Figure 1: The eigenvalues of the Fisher Information Matrix for various multi-parameter models, plotted on a logarithmic scale. Figure adapted from [4].

While an individual entry $g_{\mu\nu}$ of the FIM can be insightful for understanding how sensitive a model is to co-varying a particular combination of parameters θ^μ and θ^ν, it is often useful to consider the most salient parameter combinations in general. To do this, one can analyze the eigenspectrum of an FIM matrix. FIM eigenvectors paired with large eigenvalues correspond to parameter combinations that the model is extremely sensitive to. Consequently, the distribution of eigenvalues of the FIM can be insightful as to whether or not a large multi-parameter model can be reduced to a few particularly relevant parameter combinations. Indeed, it has been found empirically that many multi-parameter models, including systems biology models, statistical physics models, and artificial neural networks, display an FIM eigenvalue spectrum spanning many orders of magnitude [7, 10, 4]. In particular, such models, which are referred to as "sloppy", display an approximately geometric series of FIM eigenvalues (see Fig. 1). Sloppy FIM eigenspectra are in turn indicative of a particular geometric structure of the model manifold that is often referred to as a hyper-ribbon. Such hyper-ribbons are characterized by successively decreasing manifold widths along different dimensions. For such models, FIM eigenvectors paired with negligibly small eigenvalues can be considered sloppy parameter directions [10](see Fig. 2).

Since model behavior is largely unaffected by the projection of their parameters onto sloppy directions, two models that seem quite distant in parameter space may produce very similar predictions if they are only separated along a sloppy direction. From these insights, a natural way of performing model order reduction of sloppy models arises [11, 7]: 1) Compute the FIM of a model and its eigenspectrum. 2) Consider the smallest FIM eigenvalue and its corresponding eigenvector, and

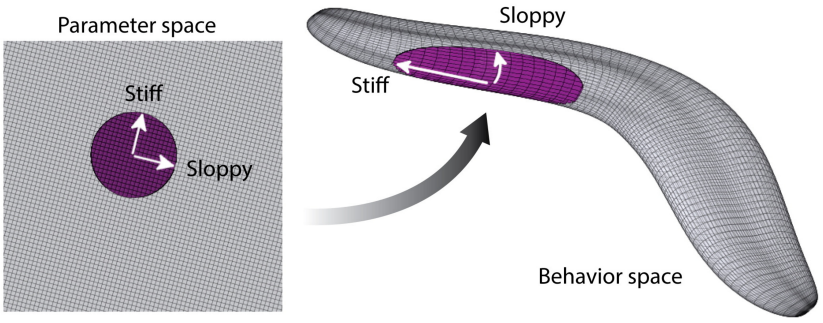

Figure 2: A section of a sloppy model's parameter space (purple) projected onto its model manifold. Figure adapted from [10].

walk along the direction in parameter space pointed by the eigenvector. Eventually, a boundary of the model manifold will be hit. 3) Apply the approximation suggested by the boundary (often involving sending parameter combinations to 0 or ∞). 4) Re-fit the model with reduced parameters, and then repeat the process until the model is sufficiently reduced.

4 Coarse-Graining in Statistical Physics Models

An insightful way of demonstrating the properties of sloppy models is through studying paradigmatic multi-parameter models from statistical physics. Recent work has demonstrated that, in multiple classic models of statistical physics, model sloppiness can arise as an emergent phenomenon from coarse-graining observable data [5]. Thus, the fitting of models to coarse-grained data may prove to be a useful way of discovering reduced order models of real-world systems. To demonstrate the power of this approach, we first reproduce the computational experiments and analyses performed in [5]. In particular, we show that both the Ising model and a discrete 1D model of particle diffusion display emergent sloppiness under coarse-graining. Furthermore, we demonstrate an alternative method not explored in [5] for demonstrating the sloppiness of the diffusion model under coarse-graining.

4.1 Coarse-graining the Ising Model

One of the most well-studied models of statistical physics is the Ising model of ferromagnetism. This system describes a system of discrete spins that each exist as either spin-up or spin-down. Each configuration of spins has an associated energetic cost that depends on whether adjacent spins are aligned, as well as what fraction of spins are aligned with the direction of the global external magnetic field. In this manner, spin configurations that are more energetically costly are less likely to arise. Despite its simplicity, multidimensional versions of the Ising model have been shown to exhibit phase transitions, analogous to phase transitions observed in basic states of matter [12].

In the analysis that follows, we consider the Ising model on a 2D lattice. The standard Ising model on an $L \times L$ lattice can be described as a system of spins $s_{i,j} \in \{-1, +1\}$. In this computational analysis, a set of spin configurations $\vec{x} = \{s_{i,j}\}$ is considered to be an observable, and occurs with a probability given by $P_\theta(\vec{x}) = e^{-\mathcal{H}_\theta(\vec{x})/Z}$, where $\mathcal{H}_\theta(\vec{x})$ is the model Hamiltonian and Z is the partition function. The standard 2D Ising model has a Hamiltonian of the form

$$\mathcal{H}_\theta(\vec{x}) = \theta^0 \sum_{i,j} s_{i,j} + \theta^{01} \sum_{i,j} s_{i,j} s_{i,j+1} + \theta^{10} \sum_{i,j} s_{i,j} s_{i+1,j}$$

where the three terms correspond to the external magnetic field, up-down neighbor coupling, and left-right neighbor coupling.

However, such a model has only three parameters, and thus it would be difficult to meaningfully analyze the spectrum of its Fisher Information Matrix. Likely for this reason, [5] considers an expanded version of the Ising model, where the Hamiltonian is given by

$$\mathcal{H}_\theta(\vec{x}) = \theta^\mu \Phi_\mu(\vec{x}),$$

where $\Phi_0(\vec{x}) = \sum_{i,j} s_{i,j}$, and $\Phi_{\alpha\beta}(\vec{x}) = \sum_{i,j} s_{i+\alpha} s_{j+\beta}$. In particular, the model considered has 13 parameters: the global coupling θ^0, as well as 12 other spin couplings $\theta^{\alpha\beta}$.

However, existing techniques and algorithms for efficiently simulating the Ising model (Monte-Carlo, Wolff [13]) do not cleanly apply to this expanded Ising model. Consequently, [5] limits their simulations to Ising configurations with $\theta^0 = 0$, $\theta^{10} = \theta^{01}$, and $\theta^{\alpha\beta} = 0$ (the regular Ising model with no external magnetic field).

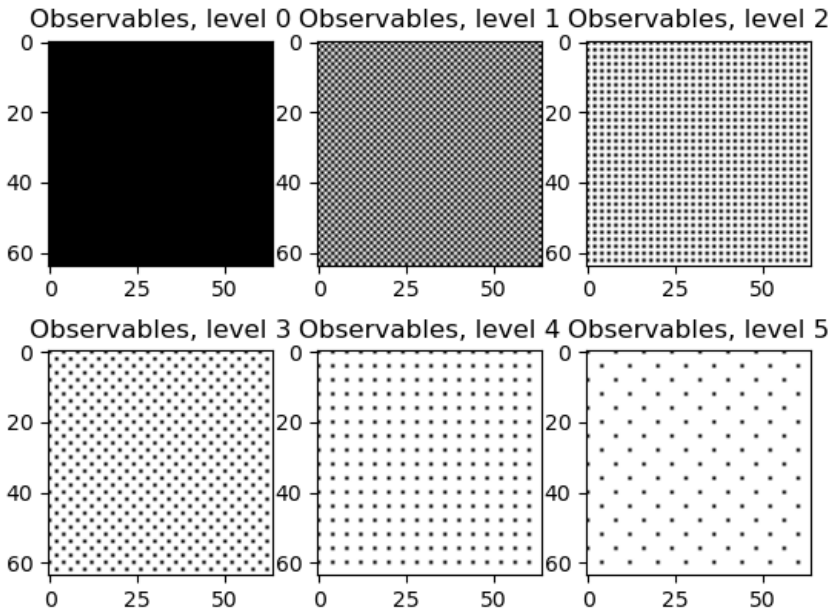

Figure 3: A schematic depicting observable spin sites on the Ising lattice under the observation coarsening scheme analyzed in [5]. Each black site represents an observed spin at a particular level.

The Fisher Information Metric of this expanded Ising model can be cleanly written in terms of the Φ spin correlation functions, as follows:

$$g_{\mu\nu} = \langle \Phi_\mu \Phi_\nu \rangle - \langle \Phi_\mu \rangle \langle \Phi_\nu \rangle \tag{10}$$

To coarsen the Ising model, [5] considers an approach where increasingly smaller subsets of the spins are considered as observables, as opposed to the more common block-spin procedure (see Fig. 3). This approach turns out to be computationally more tractable.

Computing the Fisher metric at level 0 (no coarsening) was done by simulating the Ising model with 1000 different initial conditions, each for 1000 Wolff-algorithm steps. Then, the quantity described in Eqn. (1) was numerically estimated for every pair of parameters μ, ν.

After coarsening, computing the metric becomes more complicated. [5] gives the post-coarsening metric as

$$g^n_{\mu\nu} = \left\langle \{\Phi_\mu(x)\}_{C^n(x)} \{\Phi_\nu(x)\}_{C^n(x)} \right\rangle - \left\langle \{\Phi_\mu(x)\}_{C^n(x)} \right\rangle \left\langle \{\Phi_\nu(x)\}_{C^n(x)} \right\rangle \tag{11}$$

where $g^n_{\mu\nu}$ represents the metric at coarsening level n, and $\{Q\}_{C^n(x)}$ is a notation that represents the expected value of the quantity Q, conditioned on a system coarsening to x^n (x^n representing the observed spins of configuration x at level n).

To estimate the coarsened spin correlation functions shown in Eqn. (2) at some level n, it was necessary to first simulate an ensemble of Ising model configurations ($M = 200$ was chosen as the size of the ensemble). Then, for each member of the ensemble, further simulation of the Ising configuration is run, but with the observables at level n fixed. For each member of the overall ensemble, a sub-ensemble of $M' = 200$ configurations were generated in this manner. However, equilibriation of each of the sub-ensemble configurations was quite slow, and further tricks were needed to speed up the simulations. These tricks are described in detail in the supplement of [5]. The main idea was to first integrate out half of the spins (specifically, the white spins at level 1 shown in Fig. 1).

The results are shown in Fig. 4*B*. The qualitative findings closely match those found in [5] (see. Fig. 4*A*), but there is one key difference to note. Specifically, the eigenvalues found in this analysis spanned a range around 10^2 larger than those of the original work. However, these results remain consistent with the behavior near the critical temperature/coupling combination shown in [5] (see. Fig. 4*C*).

4.2 Coarse-Graining the Diffusion Model

Diffusion processes underlie a wide variety of phenomena, including fluid dynamics and heat conduction[14].

The diffusion model considered in these experiments is the discrete 1D diffusion model. This model is characterized by $2N + 1$ parameters θ^μ that describe the probability of a particle hopping to a location μ steps away, where $-N \leq \mu \leq N$. We take the distribution of particles across all locations j at time t as the observables, given by $\vec{x} = \rho_t(j)$. Note that this model can be viewed as a discretization of the 1D version of the continuous diffusion equation given by

$$\partial_t \rho(r,t) = D\nabla^2 \rho - \vec{v} \cdot \nabla \rho + R\rho \tag{12}$$

where D, $\vec{v}$, and R represent the diffusion constant, drift, and particle creation rate, respectively.

For simplicity, we consider simulations where all particles start at the origin, giving the distribution $\rho_0(j) = \delta_{j,0}$. With these initial conditions, after one time-step has elapsed, the distribution of particles mirrors the hopping probabilities:

$$\rho_1(j) = \begin{cases} \theta^j & -N \leq j \leq N \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

To compute the Fisher Information Matrix after t timesteps, we can appeal to the assumptions made in [5], where the authors assume that uncertainty in measurement at each site is Gaussian with a width of $\sigma = 1$. With this assumption, it can be shown that the FIM is given by $g = J^\top J$, where J is the Jacobian matrix defined by

$$J_{i\mu} = \frac{\partial}{\partial \theta^\mu} \frac{\rho_t(i)}{\sigma} = \partial_\mu \rho_t(i) \tag{14}$$

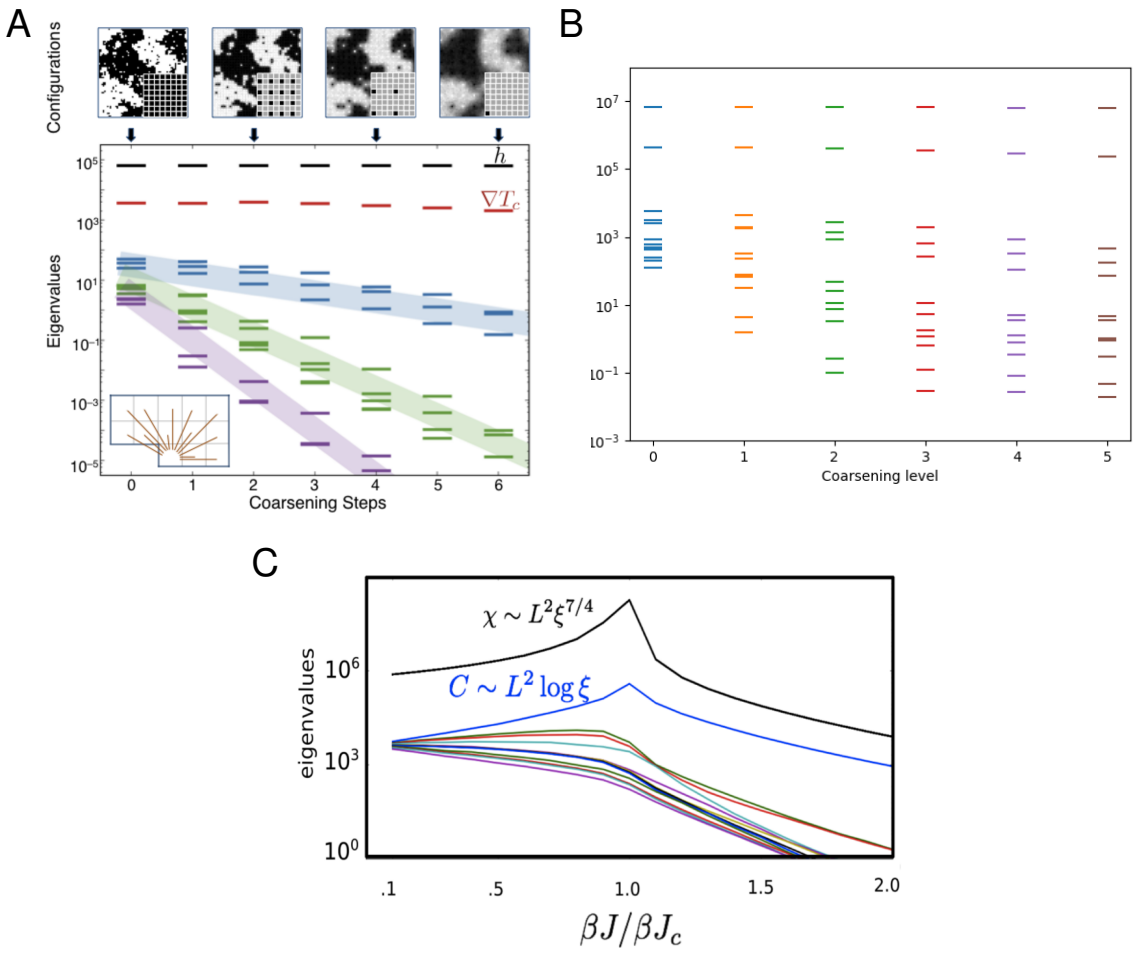

Figure 4: Spectrum of the Ising model FIM at different levels of coarsening, produced via Monte Carlo simulations. **(A)** Adapted figure of the original Ising model FIM eigenspectrums computed in [5]. **(B)** The reproduced Ising Model FIM eigenspectrums, computed in this work. **(C)** Behavior of Ising FIM eigenvalues versus temperature, adapted from [5].

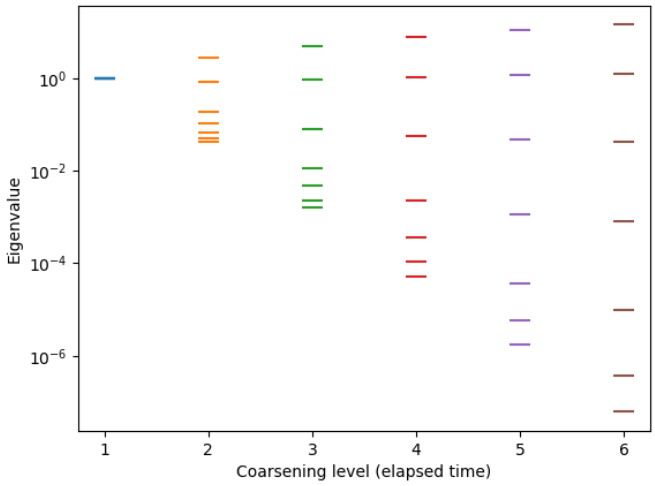

Figure 5: Spectrum of the diffusion model, reproduced analytically in this work for $N = 3$, $\theta^\mu = 1/7$ for all μ.

From Eqn. (2), we can observe that after just 1 timestep, $J_{i\mu} = \delta_{i\mu}$, and consequently the FIM is given by the identity matrix $g^1_{\mu\nu} = \delta_{\mu\nu}$.

To coarsen the diffusion model, we again follow [5] and do this by just observing the diffusion model at later times. However, analytically computing the FIM at timesteps later than 1 is more complicated, but is done in the supplement of [5], yielding

$$g^t_{\mu\nu} = \frac{t^2}{2\pi} \int_{-\pi}^{\pi} dk e^{ik(\mu-\nu)} (\tilde{\theta}^k)^{t-1} (\tilde{\theta}^{-k})^{t-1} \tag{15}$$

where $\tilde{\theta}^k = \sum_{\mu=-N}^{N} e^{-ik\mu}\theta^\mu$ represents the Fourier transform of the model parameters.

Using the analytic result of Eqn. 4, we computed the FIM for a diffusion model with $N = 3$, $\theta^\mu = 1/7$ for all μ. The corresponding spectrum obtained is shown in Fig. 5, which closely resembles the result of [5] (see Fig. 7*A*). As a sanity check that these results are qualitatively similar regardless of N or the model parameters, this analysis was also performed with $N = 10$ and random hopping rates θ^μ (See Fig. 6).

To go a step further, we wanted to see if the same results could be reproduced purely through observing the outcomes of diffusion model simulations, as often times the model form is not as privy to analyses that give a precise equation for the FIM such as Eqn. (15). Note that this was not done in the original paper [5]. Our strategy was to simulate the diffusion model with a finite but large number of particles (M = 20000 was chosen) for the appropriate number of timesteps, and then compute the Jacobian $J_{i\mu} = \partial_\mu \rho_t(i)$ empirically via finite differences. With the Jacobian, we could then appeal to the result that $g = J^\top J$.

However, computing the Jacobian empirically in this manner proved challenging. To demonstrate

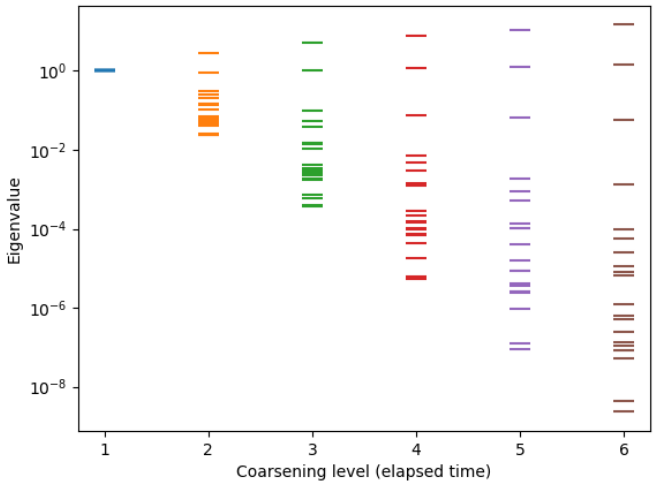

Figure 6: Spectrum of the diffusion model, computed analytically in this work for $N = 10$ and randomly chosen hopping parameters θ^μ.

why, consider the diffusion model $N = 3$ with uniform hopping parameters $\theta^\mu = 1/7$. To compute $J_{i\mu} = \partial_\mu \rho_t(i)$, one would like to consider a small deviation $\delta\theta^\mu$ from the parameters, simulate the diffusion model with parameters $\theta^\mu + \delta\theta^\mu$, and then estimate the Jacobian as

$$J_{i\mu} = \partial_\mu \rho_t(i) \approx \frac{\rho_t(i)|_{\theta^\mu + \delta\theta^\mu} - \rho_t(i)|_{\theta^\mu}}{\delta\theta^\mu} \tag{16}$$

However, the constraint that $\sum_\mu \theta^\mu = 1$ complicates this kind of analysis. Specifically, if θ^0 was increased by $\delta = .001$, then, to maintain normalization, on average, other hopping parameters θ^{-2}, θ^{-1}, θ^1, etc. would decrease. As a result, after 1 timestep, computing derivatives via finite differences would yield erroneous results such as $J_{i\mu} = -0.4$ where $i \neq \mu$, despite the fact that all off-diagonal elements should be 0. This proved to be non-negligible, and caused the estimated Jacobian after just 1 timestep to be completely off (very significantly different from the identity matrix).

Our solution to this was to consider small perturbations of model parameters $\delta\theta^\mu$ without normalization. In particular, to compute $J_{i\mu}$, Eqn. (16) was used with a small perturbation $\delta\theta^\mu = 0.001$, while all other hopping rates were kept constant. Note that since $\sum_\mu \theta^\mu + \delta\theta^\mu$ is now slightly greater than 1, a small number of particles had to be allowed to hop to two locations in one time-step, causing the total number of particles in the simulation to increase. While an unorthodox method, this solved the aforementioned problem with the finite differences computation, and completely recovered the result that $J^1_{i\mu} = \delta_{i\mu}$. The FIM was computed via simulations in this manner at later timesteps as well, with the corresponding spectrums shown in Fig. 7*B*. The spectrums obtained through simulations in this manner appear qualitatively similar to the original analytical spectrums found in [5] (see Fig. 7*A*), confirming the validity of the approach.

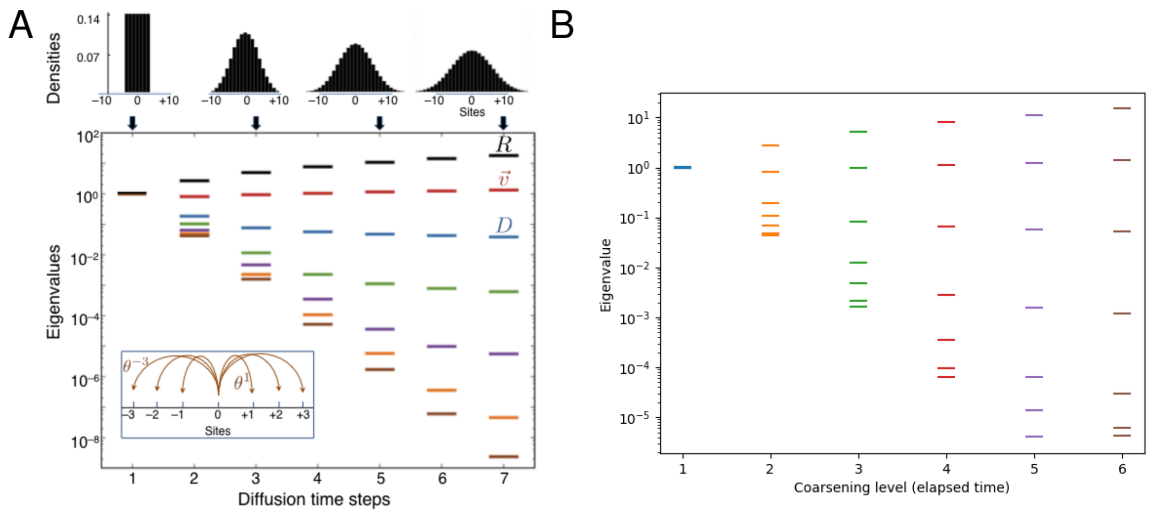

Figure 7: A comparison of computed FIM eigenspectrums for the diffusion model with parameters $N = 3$, $\theta^{\mu} = 1/7$ for all μ. **(A)** The original diffusion model spectrum computed analytically, adapted from [7]. **(B)** The same diffusion model spectrum, reproduced numerically in this work through particle-level simulations.

5 System Identification: An Overview

This section explores some of the most prominent techniques for identifying the dynamics that underlie observed data. In particular, we focus on system identification techniques for systems whose dynamics can be formulated in terms of differential equations.

5.1 Sparse Regression

One approach for system identification that has proven quite successful in certain system identification problems is to apply sparse regression techniques to a library of nonlinear functions of observed input data. This approach has proven useful for discovery of both ordinary differential equations (ODEs) and partial differential equations (PDEs) [3, 15]. Specifically, these works employ finite differences or polynomial-interpolation based methods to compute approximate spatial and temporal derivatives of some observed state variable of interest. For the goal of discovering dynamics underyling data generated from an ODE, this is particularly useful because ODEs can often be expressed in the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{17}$$

Furthermore, it can often be assumed that $\mathbf{f}$ is composed of no more than a few terms involving $\mathbf{x}$ and its components. Consequently, discovering the underlying ODE amounts to solving a sparse regression problem that takes in a library of combinations of numerically approximated derivatives and nonlinearities of the observed state variable $\mathbf{x}(t)$ and predicts a numerically approximated form of $\frac{d}{dt}\mathbf{x}(t)$. Note that this approach can also be applied to models that require second derivatives in time of the state variable (e.g. Newtonian forces). In particular, one can re-parameterize the system of interest in terms of a new state variable $\tilde{\mathbf{x}} = [\mathbf{x}, \frac{d}{dt}\mathbf{x}]$ to recover the desired form. Using these methods, Brunton et al. demonstrated that the precise equations governing chaotic ODE systems including the Lorenz system and mean-field vortex shedding dynamics can be recovered from data to a high degree of accuracy [3].

Similarly, PDEs can often be expressed in the form

$$u_t = f(u, u_x, u_{xx}, \dots u^2 u_x \dots) \tag{18}$$

allowing for the discovery of the the terms that compose f in an analogous manner. Along this line of reasoning, Brunton et al. also extended the sparse regression approach to PDEs, and successfully applied the approach to identify chaotic PDEs such as the Navier-Stokes equations from sparsely sampled observations [15]. The methodology of this approach is summarized in Fig. 8.

However, these methods certainly have limitations. One major limitation is the accuracy of numerical differentiation methods. Especially at higher order derivatives, significant numerical error is often unavoidable in the computation of higher order derivatives via finite differences or polynomial approximation schemes [16, 15]. Indeed, the sparse aggression approach combined with classical numerical differentiation methods failed to accurately identify the coefficients of PDEs involving higher order derivatives, such as the fourth-order Kuramoto-Sivashinsky equation [15]. Despite these limitations, sparse-regression methods remain well-suited for the task of model order reduction. In particular, a learned model with many nonzero coefficients can be successively reduced by

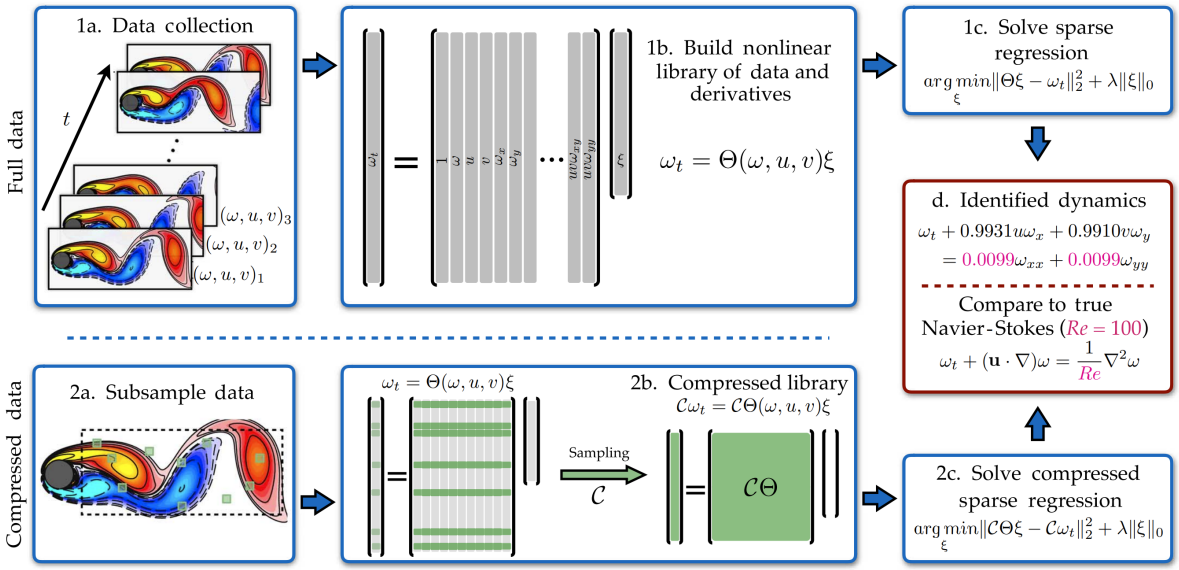

Figure 8: A schematic of the PDE-FIND method for identifying PDEs from data via sparse regression, adapted from [15].

dropping terms that explain the least amount of the variance of observed data. Moreover, learned coefficients can easily be refit following parameter pruning.

5.2 Physics-Informed Neural Networks

Another strategy for data-driven modeling of real-world systems is to take advantage of the expressivity of deep neural networks [17]. In particular, Physics-Informed Neural Networks (PINNs) propose a simple way of augmenting standard neural networks to make them more well-equipped for modelling the dynamics generated by some partial differential equation. Suppose we are trying to predict the dynamics of some state variable of interest as a function of position and time, given by $\mathbf{u}(\mathbf{x}, t)$. This state variable could be a scalar quantity, such as heat or salinity, or could be vector-valued, such as a velocity field. Suppose further that we know the PDE that governs $\mathbf{u}$'s evolution, given by $\mathbf{u}_t = \mathbf{f}(\mathbf{u}, \mathbf{u}^2, \mathbf{u} \circ \mathbf{u}_x, \ldots, \mathbf{x}, t)$. Lastly, there are partial observations of the value of $\mathbf{u}$ at different positions and times, given by a dataset $\mathcal{D} = \{(\mathbf{x}_i, t_i, \mathbf{u}_i)\}_{i=1}^N$. The goal is then to learn a model that best approximates $\mathbf{u}$, which we denote by $\mathbf{u}^\theta(\mathbf{x}, t)$.

There are two main reasons why PINNs are especially suited for this type of problem. Firstly, neural networks are highly expressive and can, in principle, approximate any smooth, continuous function [18]. Secondly, automatic differentiation allows for high-precision estimates of derivatives of interest. While finite difference and polynomial interpolation methods for approximating derivatives are in many situations sufficient, they produce errors that scale with the order of differentiation. This makes classical numerical methods less appropriate for accurate analysis of PDEs containing terms involving second-order derivatives or higher, such as the Laplacian operator. On the other hand, automatic differentiation operates via the chain-rule, analytically decomposing derivatives into elementary functions before any numerical evaluation, and is thus more robust to numerical error. Applying automatic differentiation, one can use a model $\mathbf{u}^\theta$ to compute high precision estimates

of $\mathbf{u}_t$, $\mathbf{u}_x$, $\mathbf{u}_{xx}$, and other spatiotemporal derivatives of the state variable. This in turn allows for computing estimates of the dynamics of $\mathbf{u}$ given by $\mathbf{f}$. From this, a natural loss function for training PINNs arises:

$$\frac{1}{N}\sum_{i=1}^{N}||\mathbf{u}_i^\theta - \mathbf{u}_i||^2 + \frac{1}{N}||\dot{\mathbf{u}}_i^\theta - \mathbf{f}_i^\theta||^2 \tag{19}$$

Here, $\dot{\mathbf{u}}_i^\theta$ represents the approximation of $\mathbf{u}_t|_{(\mathbf{x}_i,t_i)}$ by automatic differentiation, and $\mathbf{f}_i^\theta$ represents the approximate dynamics $\mathbf{f}|_{(\mathbf{x}_i,t_i)}$ at a particular position and time. The first term of this loss function is often referred to as the data loss, and ensures that $\mathbf{u}^\theta$ is a good approximator of $\mathbf{u}$, whereas the second term–which explains the name “physics-informed"–enforces consistency between $\mathbf{u}^\theta$ and the known dynamics $\mathbf{f}$.

However, while PINNs are a useful tool for learning dynamics that arise from PDEs, they are difficult to use for the task of system identification. In particular, they require knowledge of the precise functional form of the PDE that governs the underlying dynamics, which seems to defeat the purpose of their usage for system identification. In the section that follows, we will attempt to circumvent this problem and explore an approach that combines the strategy of PINNs described in [17] with the sparse regression approaches applied in [3, 15].

6 PDE System Identification

In this section, we explore techniques for system identification of PDEs from spatiotemporal data. We begin by introducing an approach that integrates PINNs [17] with sparse regression techniques for PDE discovery, and apply it to two nonlinear PDEs. Then, we investigate methods for discovering reduced order models of large ensembles of microscopic observables. For all methods analyzed, we consider how the behavior of discovered models changes under coarse-graining.

6.1 General Methods

This section describes the general experimental and computational methods used for the following analyses.

One recent approach for discovering PDEs from data integrates the methods used by PDE-FIND [15] with Physics-Informed Neural Networks [17]. In particular, the work of [19] applies the automatic differentiation capabilities of deep neural networks (DNNs) to construct libraries of derivatives and candidate nonlinearities for sparse regression. In this context, automatic differentiation has proven to be a powerful tool; one of the main challenges faced by the approach of [3] were numerical inaccuracies arising from applying classical numerical differentiation methods to estimate higher order derivatives. Automatic differentiation, on the other hand, allows for the computation of spatial and temporal derivatives to much higher precision [19]. During the training of these DNNs, an initial estimate of the sparse regression weights is learned simultaneously, which in turn provides an inductive bias to the DNN that is analogous to the physics-informed prior in PINNs.

We first obtain data $\mathcal{D} = \{(\mathbf{x}_i, t_i, \mathbf{u}_i)\}_{i=1}^N$ that we would like to fit a PDE to. We then use this data to train a DNN to approximate the state variable(s) $\mathbf{u}$, which we denote by $\mathbf{u}^\theta(\mathbf{x}, t) \approx \mathbf{u}(\mathbf{x}, t)$.

We then adapt the approach of [19] by constructing a library $\boldsymbol{\phi}(\mathbf{u}) \in \mathbb{R}^{1\times s}$ of candidate nonlinear functions of the components of the state variable $\mathbf{u}$, such as
$\boldsymbol{\phi} = \{1, \mathbf{u}, \mathbf{u}^2, \mathbf{u}^3, \dots, \mathbf{u}_x, \mathbf{u} \circ \mathbf{u}_x, \dots\}$.

$$\mathbf{u}_t = \boldsymbol{\phi}\boldsymbol{\Lambda} \tag{20}$$

where the matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{s\times n}$ denotes the coefficients of the candidate nonlinearities. To train the DNN to approximate $\mathbf{u}(\mathbf{x}, t)$ and to learn initial estimates of candidate nonlinearity coefficients, we again follow the approach of [19] and construct a loss function with a three main components: a state variable estimation loss, a physics-prior/regression loss, and a regularization penalty to encourage sparsity. The first component, $\mathcal{L}_{data}(\boldsymbol{\theta}; \mathcal{D})$, quantifies the mean squared error between the DNN's approximation of the state variable and its true value across all available measured data, as defined in Eqn. (21):

$$\mathcal{L}_{data}(\boldsymbol{\theta}; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{u}_i^\theta - \mathbf{u}_i||^2 \tag{21}$$

In addition, we penalize the residual between the approximate time derivative of the state variable

and the sparse regression predictions:

$$\mathcal{L}_{physics}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} ||\dot{\mathbf{u}}_i^{\theta} - \boldsymbol{\phi}(\mathbf{u}_i)\boldsymbol{\Lambda}||^2 \tag{22}$$

where $\dot{\mathbf{u}}_i^{\theta}$ denotes the approximation of $\frac{\partial \mathbf{u}}{\partial t}|(\mathbf{x}_i, t_i)$ by automatic differentiation using the DNN.

Finally, we include L_1 regularization of $\boldsymbol{\Lambda}$ to encourage sparsity of learned coefficients. The total loss is then given by

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \mathcal{L}_{data}(\boldsymbol{\theta}; \mathcal{D}) + \alpha \mathcal{L}_{physics}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}) + \beta ||\boldsymbol{\Lambda}||_1 \tag{23}$$

where α and β control the relative importances of the physics prior and the L_1 regularization, respectively.

To ensure higher-order differentiability and smoothness of network predictions, we use the $\tanh$ activation function. For each experiment, the DNN was trained for 200 epochs. To improve on both the performance and interpretability of the approach of [19], we made a few changes to the training process and DNN architecture. First, for simplicity, we reduced the DNN to a three-layer feed-forward network. Additionally, the weight of the physics loss α was set to 0 for the first 100 epochs of training. This was done to allow the DNN to learn to approximate $\mathbf{u}$ well before enforcing the consistency condition of the physics loss, reducing the risk of the DNN getting stuck in a poor local minimum. Finally, while [19] refines the sparse regression coefficient estimates $\boldsymbol{\Lambda}$ after DNN training through an alternating optimization process, for simplicity, we use the Forward-Stagewise regression algorithm [20] for this purpose.

Once trained, we then computed the FIMs of the DNN's approximation of the state space $\mathbf{u}^{\theta}(\mathbf{x}, t)$, the resulting approximation of its time derivative $\dot{\mathbf{u}}^{\theta}(\mathbf{x}, t)$, and the sparse regression predictions $\boldsymbol{\phi}(\mathbf{u}^{\theta})\boldsymbol{\Lambda}$. We then investigated how spatiotemporal coarse-graining of the input data $\mathcal{D}$ used to train these DNNs affected the spectrums of their corresponding FIMs.

To analyze the effects of coarse-graining of observable data, we considered 6 different resolutions of the ground truth data of the state variable $\mathbf{u}(\mathbf{x}, t)$, sampled over a grid spaced regularly over positions and times. At level 1 of coarsening, no coarse-graining is applied. For levels $i > 1$ we employed a convolutional averaging kernel of size $(2^{i-1} + 1) \times (2^{i-1} + 1)$ to the data organized over a position-time meshgrid. The effects of this coarse-graining on spatiotemporal data are visualized in Fig. 12.

6.2 System identification of Partial Differential Equations

To evaluate the effectiveness of our approach, we numerically simulate the dynamics of a few known PDEs and attempt to re-discover their forms from the observed data.

6.2.1 Burgers' equation

We begin by applying this approach to the 1D Burgers' equation, defined below:

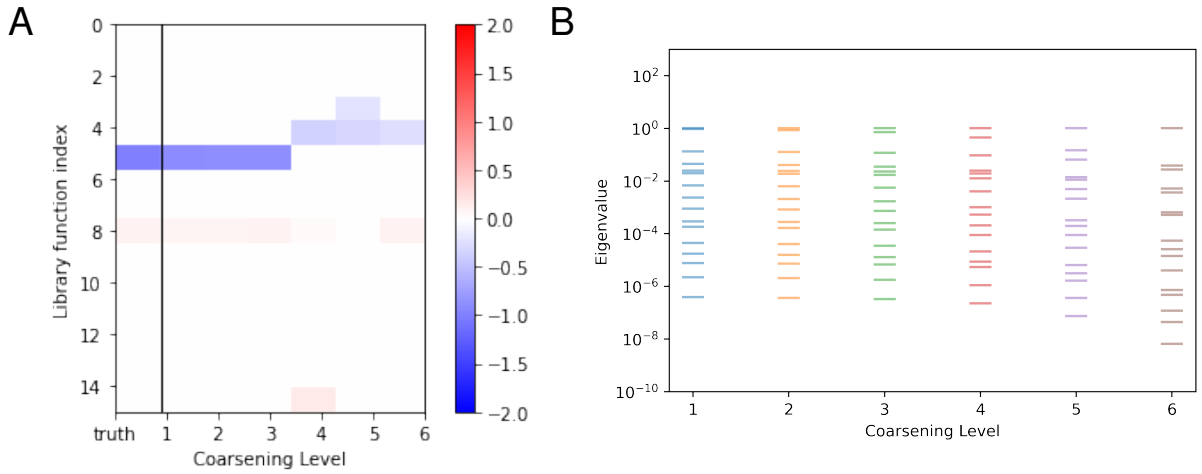

Figure 9: **(A)** Learned coefficients of Burgers' equation after Forward-Stagewise optimization vs level of coarsening. **(B)** FIM spectrum for the regression predictions versus level of coarsening.

$$u_t + uu_x - \nu u_{xx} = 0 \tag{24}$$

Burgers' equation has been used to model phenomena ranging from fluid dynamics to traffic flow [21].

For our simulations, we use $\nu = 0.1$. Since the state variable is one-dimensional, we restrict the library of nonlinearities to terms of degree at most 3, derivatives of order at most 3, and any combinations of the two, yielding the following library of terms:
$\boldsymbol{\phi} = \{1, u, u^2, u^3, u_x, uu_x, u^2u_x, \ldots u_{xxx}, \ldots u^3u_{xxx}\}$

We find that, in the absence of coarse-graining, our model successfully correctly identifies the proper terms in Burgers' equation. Moreover, the estimated coefficients are roughly correct (See Fig. 9*A*). The learned coefficients remain stable under coarse-graining until coarsening level 4, where the model begins confusing uu_x with u_x. We visualize the ground truth state variable u, as well as the predicted state variable $\hat{u}$ obtained from integrating the learned PDEs at coarsening levels 3 and 4 (Fig. 10).

To gain further insight into how coarse-graining affects the behavior of the learned model, we computed the Fisher Information Matrix of three quantities predicted by our model: $\mathbf{u}^\theta(\mathbf{x}, t)$ with respect to parameters θ, $\dot{\mathbf{u}}^\theta(\mathbf{x}, t)$ with respect to parameters θ, and the regression predictions $\boldsymbol{\phi}(\mathbf{u})\boldsymbol{\Lambda}$ with respect to parameters $\boldsymbol{\Lambda}$. As shown in Fig. 9*B*, the eigenspectrum of the FIM of regression predictions can indeed be characterized as sloppy. Furthermore, the spectrum becomes sloppier under coarse-graining, consistent with what was observed in the analysis of statistical physics models (Sec. 4). In particular, the dominant eigenvalue remains stable, but the second largest eigenvalue decays with increased coarsening. Similarly, the FIM spectrums of $\mathbf{u}^\theta(\mathbf{x}, t)$ and $\dot{\mathbf{u}}^\theta(\mathbf{x}, t)$ also span many orders of magnitude, but increased emergent sloppiness with more aggressive coarsening is not apparent (Fig. 11).

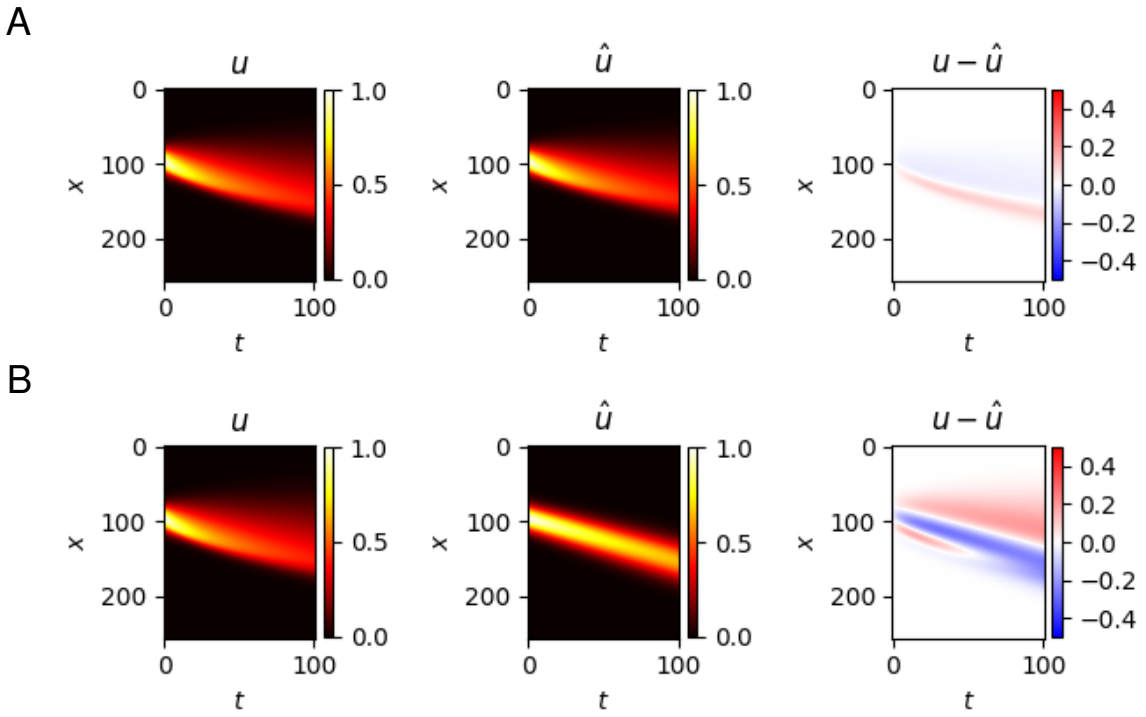

Figure 10: The ground truth state variable u of Burgers' equation, an estimate computed from integrating learned PDE coefficients $\hat{u}$, and error in estimation $u - \hat{u}$ are shown at **(A)** coarsening level 3 and **(B)** coarsening level 4, respectively.

6.2.2 The Korteweg–De Vries equation

To investigate whether this approach remains successful for more complex PDEs, we also apply it to the Korteweg-De Vries (KdV) equation, a third-order PDE that takes the form

$$u_t + 6uu_x - u_{xxx} = 0 \tag{25}$$

This PDE supports solutions in the form of travelling waves, and allows for superposition of multiple waves travelling at different speeds (Fig. 12). We use the same library of nonlinearities as before: $\phi = \{1, u, u^2, u^3, u_x, uu_x, u^2u_x, \ldots u_{xxx}, \ldots u^3u_{xxx}\}$. Unlike what was found for Burgers' equation, our model was unable to identify the correct terms in the KdV equation. In particular, even in the absence of coarse-graining, it incorrectly identified u_x as the dominant term instead of uu_x(Fig. 13*A*). Furthermore, it failed to identify u_{xxx}. At higher levels of coarse-graining, the model begins to identify the correct components, but also introduces extraneous terms. A comparison of the ground truth state variable u and the predicted state variable $\hat{u}$ obtained from integrated learned PDE coefficients is shown at coarsening levels 1 and 6 (Fig. 14). Notably, at level 1 coarse-graining, the learned model successfully produces one of the traveling waves, but is unable to simultaneously propagate another traveling wave with a different speed (Fig. 14*A*). One possible explanation for the model's inability to correctly identify the correct terms of the KdV equation is that, during training, the network becomes stuck in a local minimum where it accurately reproduces the faster traveling wave, but neglects the lower-amplitude, slower traveling wave.

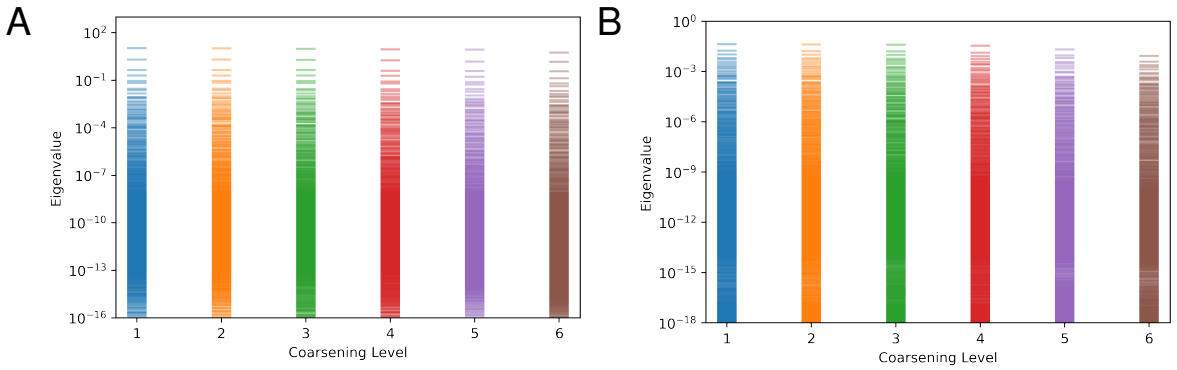

Figure 11: **(A)** FIM spectrum for $\mathbf{u}^\theta$ learned for Burgers' equation vs the level of coarse-graining. **(B)** FIM spectrum for $\dot{\mathbf{u}}^\theta$ vs the level of coarse-graining.

Analogous to our analysis of Burgers' equation, We also analyze the eigenspectra of the FIMs of the approximated state variable $\mathbf{u}^\theta(\mathbf{x}, t)$, its approximated time derivative $\dot{\mathbf{u}}^\theta(\mathbf{x}, t)$, and the regression predictions $\boldsymbol{\phi}(\mathbf{u})\boldsymbol{\Lambda}$. We once again find that all spectra span a wide range of magnitudes and therefore can be described as sloppy (Fig. 13*B*, Fig. 15). Interestingly however, we find that the regression predictions' FIM eigenspectrum does not necessarily become sloppier with greater coarsening of the training data (Fig. 13*B*). Moreover, there does not seem to be any dominant eigenvalues that remain stable under coarsening.

6.2.3 Perturbing PDEs with a diffusive term

As demonstrated with our analysis of Burgers' equation and the Korteweg-De Vries equation, the described method for discovering PDEs from data can be effective for simple PDEs, but struggles for PDEs involving higher order terms. Here we consider an alternative strategy involving analyzing how the eigenspectrum of the model's regression prediction FIM behaves under perturbing the ground truth PDE to be learned. Ideally, for every term in a library of candidate functions (e.g. u_x, u_{xx}, u^2), adding a small perturbation to a PDE consisting of that term would yield a unique effect on the FIM eigenspectrum. If this were the case, then analyzing the FIM spectrum of learned models for signatures of Candidate PDE terms would yield a fruitful way for discovering PDEs using insights from information geometry.

However, there are a few practical limitations to this strategy. Firstly, not all PDEs result in nice solutions for the state variable, as randomly assembled PDEs were almost always found to diverge upon integration. As a concrete example, Burgers' equation with a negative diffusion constant ν leads to a solution that diverges.

Considering these limitations, we restrict our analysis to perturbing PDEs with the diffusion term u_{xx}, and ensure that all perturbations analyzed result in non-negative diffusion coefficients in the final PDE. We apply this approach to the previously studied Burgers' and KdV equations.

The Burgers' and KdV equations are perturbed as follows:

$$u_t + uu_x - \nu u_{xx} = \delta u_{xx} \tag{26}$$

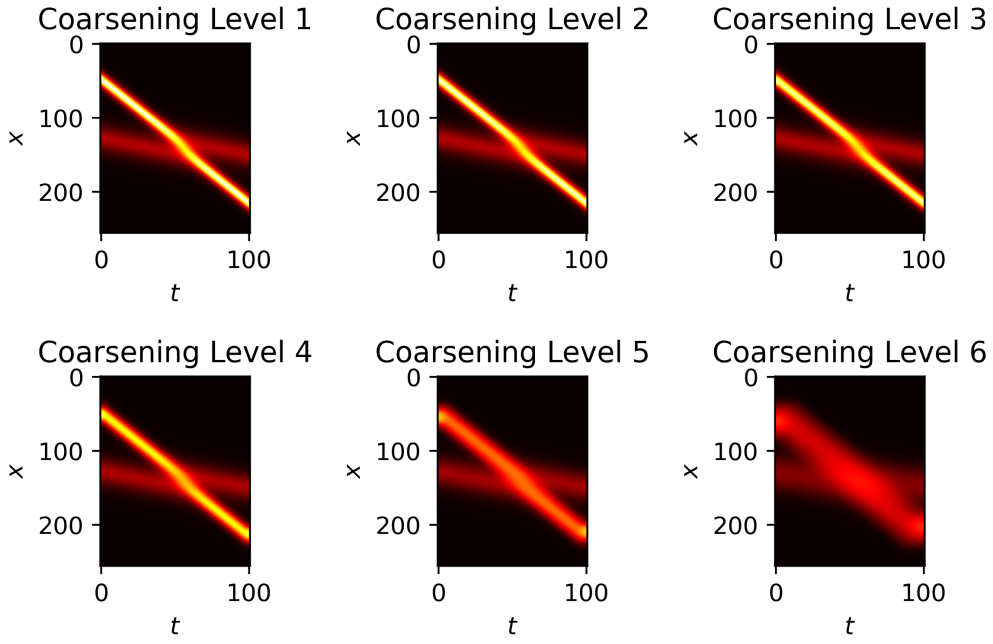

Figure 12: The ground truth form of the state variable $\mathbf{u}(\mathbf{x}, t)$ for the Korteweg–De Vries equation at different levels of coarse graining.

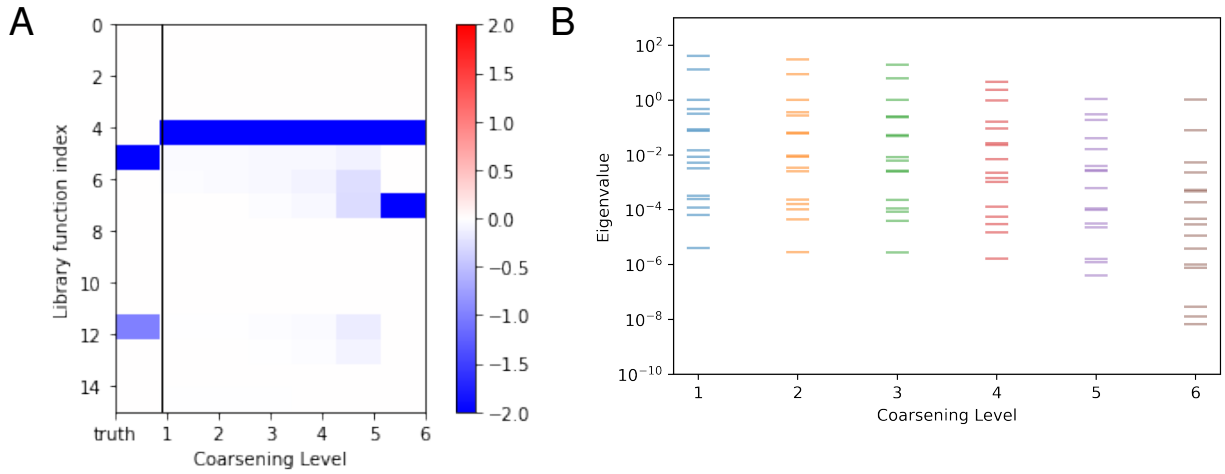

Figure 13: **(A)** Learned coefficients for the Korteweg-De Vries equation after Forward-Stagewise optimization versus level of coarsening. **(B)** FIM spectrum for the regression predictions versus level of coarsening.

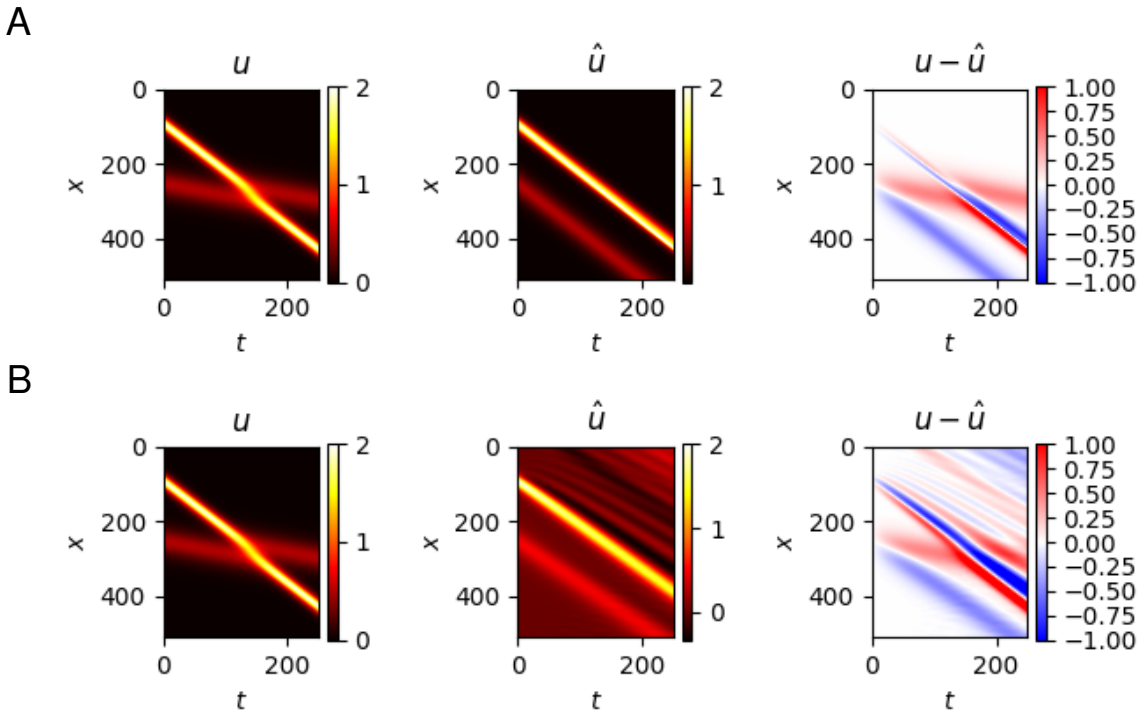

Figure 14: The ground truth state variable u of the Korteweg-De Vries equation, an estimate computed from integrating learned PDE coefficients $\hat{u}$, and error in estimation $u - \hat{u}$ are shown at **(A)** coarsening level 1 and **(B)** coarsening level 6, respectively.

$$u_t + 6uu_x - u_{xxx} = \delta u_{xx} \tag{27}$$

where we choose perturbations $\delta \in [-0.08, 0.08]$ and $\delta \in [0, 0.4]$ for the Burgers' and KdV equations, respectively. We then trained models as before on data generated from these perturbed PDEs, and analyzed how perturbations affect relevant FIM eigenspectra. For Burgers' equation, we observe clear effects of the perturbation process on the regression predictions' FIM eigenspectra (Fig. 16*A*). In particular, decreasing the diffusivity of the equation via supplying a negative perturbation leads to large dominant eigenvalues and a sloppier spectra. This result is sensible considering that Burgers' equation exhibits a shock wave when the overall diffusion coefficient is small [22]. On the other hand, steadily increasing diffusivity also appears to increase overall spectra sloppiness, but causes a different eigenvalue mode to dominate. On the other hand, perturbing the KdV equation via increasing diffusivity appears to cause decay of the first two eigenvalues, but has a stabilizing effect on the third eigenvalue (Fig. 17*A*). However, it is difficult to discern evidence of a unique signature of u_{xx} perturbation in the FIM spectra that is present for both equations. Future analysis along this line of reasoning should investigate how the dominant FIM eigenvectors corresponding to directions stiff directions of parameter space are perturbed as a consequence of PDE perturbation.

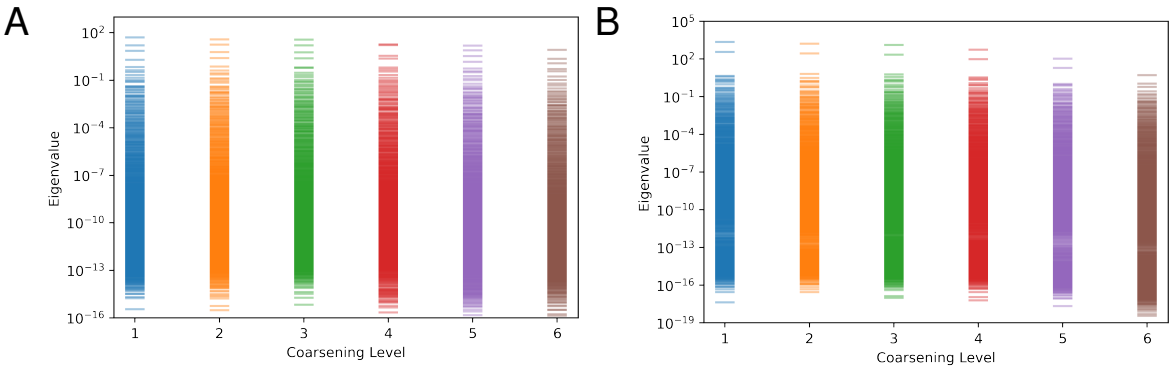

Figure 15: **(A)** FIM spectrum for $\mathbf{u}^\theta$ learned for the Korteweg-De Vries equation vs the level of coarse-graining. **(B)** FIM spectrum for $\dot{\mathbf{u}}^\theta$ vs the level of coarse-graining.

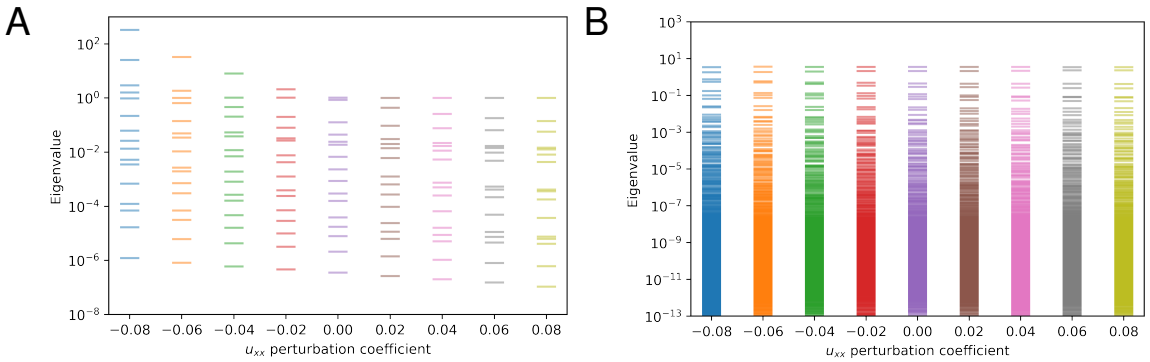

Figure 16: **(A)** FIM spectrum of regression predictions versus the coefficient of u_{xx} perturbation applied to Burgers' equation. **(B)** FIM spectrum for $\mathbf{u}^\theta$ vs the coefficient of u_{xx} perturbation.

6.3 Identifying macroscopic models from ensembles of microscopic observables

One area where reduced order models are particularly useful is for understanding the emergent dynamics that govern ensembles of microscopic observables. For example, in a fluid system containing calcium ions, one might be able to roughly model calcium ion concentrations as a function of position and time. However, actually tracking each of the trajectories of N individual ions in three spatial dimensions would require $3N$ state variables. For real-world scenarios, where a mole of ions would yield $N \sim 10^{24}$ trajectories to follow, such a high-dimensional description of the system would be both highly impractical and different to interpret. Moreover, for such a system, it is unlikely that each individual ion is governed by unique forces, but rather that all ions roughly share some form of underlying dynamics. One would then expect that an accurate macroscopic description of the ions' collective behavior should be feasible.

We can frame this problem more generally as follows: Suppose we have a system of ODEs governed by

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}); \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{28}$$

Each choice of initial condition $\mathbf{x}_0$ gives rise to a unique trajectory $\mathbf{x}(t)$. Now, suppose that there

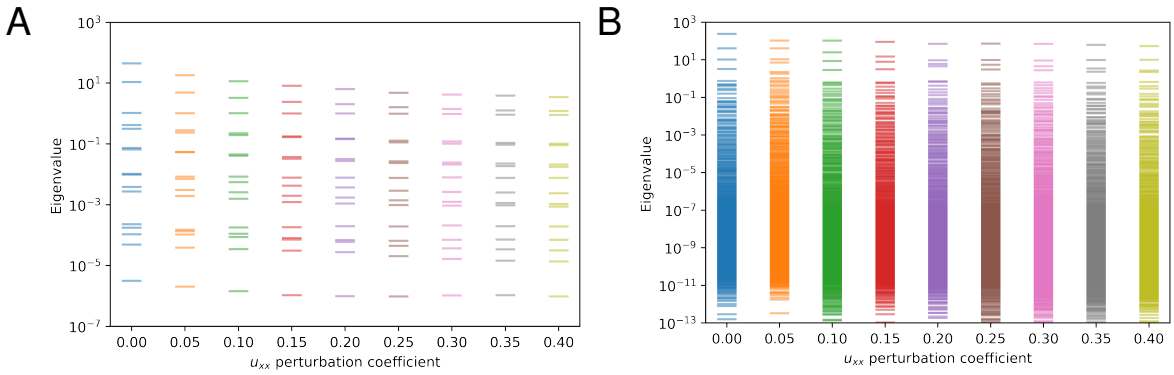

Figure 17: **(A)** FIM spectrum of regression predictions versus the coefficient of the u_{xx} perturbation applied to the Korteweg-De Vries equation. **(B)** FIM spectrum for $\mathbf{u}^\theta$ vs the coefficient of u_{xx} perturbation.

are N independent trajectories that obey the same dynamics $\mathbf{f}$, each with their own initial conditions given by $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$. For any such system of trajectories obeying the same underlying dynamics, a macroscopic description of their collective dynamics exists in the form of a PDE. This PDE is given by the advection equation, and in general describes the conservation of a scalar quantity as it is being advected by a velocity field:

$$\rho_t + \nabla \cdot (\mathbf{f}\rho) = 0; \quad \rho(\mathbf{x}, 0) = \rho_0(\mathbf{x}); \quad \rho|_{\Gamma_i} = 0 \tag{29}$$

Here, $\nabla \cdot (\dots)$ denotes the divergence operator, and $\rho_0(\mathbf{x})$ and $\rho|_{\Gamma_i}$ represent initial and boundary conditions for the trajectory densities, respectively. In this case, the underlying microscopic dynamics of ODE trajectories $\mathbf{f}$ serves as the velocity field for this PDE.

A question then arises: given ODE trajectory densities evaluated at different positions and times, estimated from many observations of individual trajectories, could one learn the form of the PDE given in Eqn. (29)? We investigate this problem from two angles: 1) learning Eqn. (29) from scratch by learning coefficients of libraries of derivatives and nonlinearities of position and state variables (analogous to what was done for Burgers' equation and the Korteweg-De Vries equation), and 2) directly learning the coefficients of the components of $\mathbf{f}$ from a library of nonlinearities of position variables. The first approach attempts to learn the entirety of the trajectory density PDE from scratch, whereas the second approach adds additional structure to the task, taking advantage of prior knowledge that trajectory densities can be modelled by an advection equation. We refer to these strategies as the standard approach and advection-based approach, respectively.

6.3.1 Reduced order Modeling of Van der Pol Ensembles

To explore these methods, we first consider the Van der Pol oscillator, whose dynamics are governed by a set of coupled ODEs given as follows:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= \mu(1 - x^2)y - x \end{aligned} \tag{30}$$

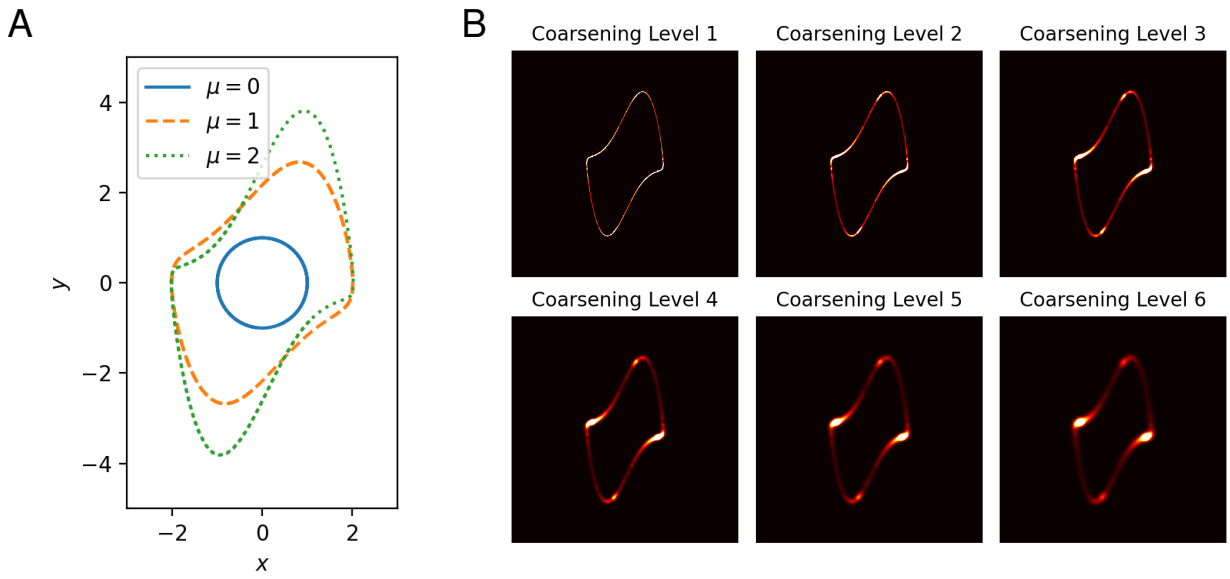

Figure 18: **(A)** Limit cycles of the Van der Pol system for different values of μ. **(B)** Snapshots of the estimated Van der Pol trajectory densities at $t = 5.0$ for different levels of spatial coarse graining. Density estimates were produced from $N = 300,000$ Van der Pol trajectories with initial conditions sampled uniformly from a circle of radius 3 centered at the origin.

Importantly, each individual Van der Pol trajectory will eventually converge onto a limit cycle, whose shape is governed by the parameter μ.

We can recast the ODE dynamics of Van der Pol oscillators to the form of Eqn. (28) as follows:

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}); \quad \mathbf{f} = \begin{bmatrix} y & \mu(1-x^2)y - x \end{bmatrix}^\top \tag{31}$$

We simulated $N = 300,000$ trajectories of the Van der Pol system with $\mu = 2$, using initial conditions sampled uniformly from a circle of radius 3 centered at the origin. Then, the density of trajectories at each position and time was estimated by binning the number of observed trajectories over time in each unit of a discretized x-y grid.

We first attempt to discover the PDE governing Van der Pol trajectory density via the standard approach. Since the ground truth form of the PDE depends on not only the state variable ρ (trajectory density), but also position variables x and y, the library of functions for which to learn coefficients must be expanded. In particular, we consider powers of ρ given by $\mathcal{U} = \{1, \rho, \rho^2, \rho^3\}$, powers of position variables $\mathcal{P} = \{1, x, x^2, x^3, y, y^2, y^3, xy, x^2y, xy^2\}$, and derivatives of ρ up to second order $\mathcal{D} = \{1, \rho_x, \rho_{xx}, \rho_y, \rho_{yy}, \rho_{xy}\}$. The final library of nonlinearities is then given by $\boldsymbol{\phi} = \{upd | u \in \mathcal{U}, p \in \mathcal{P}, d \in \mathcal{D}\} \in \mathbb{R}^{1\times 240}$.

Upon expanding the divergence operator of Eqn. (29), we find that there are 6 terms in $\boldsymbol{\phi}$ that appear in the ground truth expression for the dynamics of Van der Pol trajectory density: ρ, $x^2\rho$, $y\rho_x$, $y\rho_y$, $x\rho_y$, and $x^2y\rho_y$.

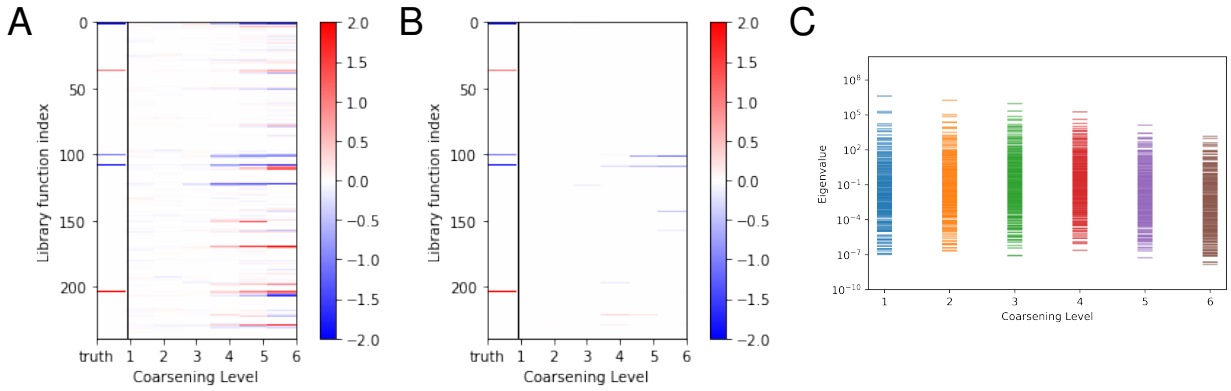

Figure 19: **(A)** Learned coefficients for the Van der Pol density (direct approach) before Forward-Stagewise optimization versus level of coarsening. **(B)** Learned coefficients for the Van der Pol density (direct approach) after Forward-Stagewise optimization versus level of coarsening. **(C)** The FIM spectrum of the regression predictions versus level of coarsening.

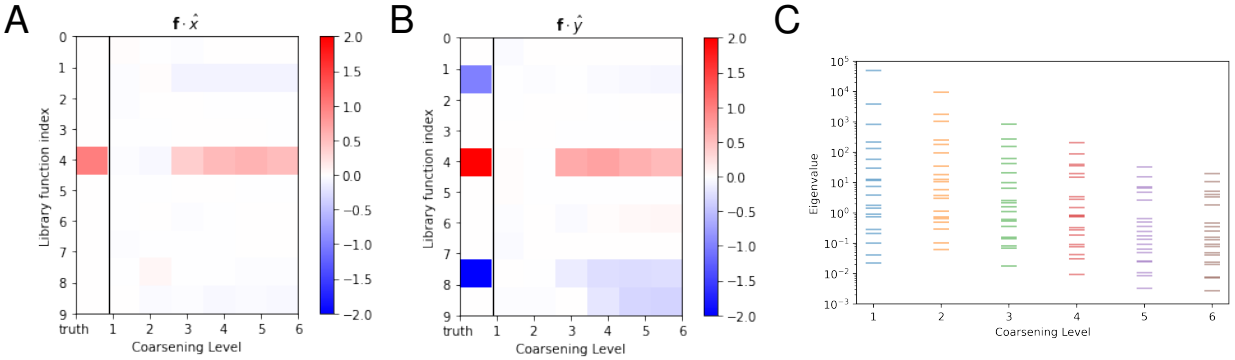

Figure 20: Learned coefficients for the Van der Pol density velocity field (advection-based approach) versus level of coarsening, shown for **(A)** the x-component $\mathbf{f} \cdot \hat{x}$ and **(B)** the y-component $\mathbf{f} \cdot \hat{y}$. **(C)** The FIM spectrum of the advection predictions versus level of coarsening.

Using this approach, our model was unable to identify any of the proper coefficients in the absence of coarse-graining. However, upon coarse-graining the observable data, the correct terms begin to be identifiable, albeit accompanied by the emergence of extraneous terms (Fig. 19 *A,B*). Pruning coefficients via the Forward-Stagewise optimization aids in dropping unwanted terms, but also prunes some terms that actually contribute to the ground truth dynamics (Fig. 19*B*).

Next, we apply the advection-based approach. As before, we approximate the trajectory density via a DNN parameterized by θ, denoted by $\rho^\theta(\mathbf{x}, t)$. Given that each component of the dynamics $\mathbf{f}$ must be a function of position, we can construct a candidate library consisting solely of nonlinearities of position variables, given by $\boldsymbol{\phi}^{\text{spatial}} = \{1, x, x^2, x^3, y, y^2, y^3, xy, x^2y, xy^2\} \in \mathbb{R}^{1\times 10}$. Using this library, we learn the coefficients of terms of $\boldsymbol{\phi}^{\text{spatial}}$ corresponding to $\mathbf{f}\cdot\hat{x}$ and $\mathbf{f}\cdot\hat{y}$, given by $\boldsymbol{\Lambda} \in \mathbb{R}^{10\times 2}$. In particular, estimates of the underlying ODE dynamics are given by $\mathbf{f}^\Lambda = \boldsymbol{\phi}^{\text{spatial}}\boldsymbol{\Lambda}$. Then, taking advantage of automatic differentiation, we can approximate $\nabla\cdot(\rho\mathbf{f})$. The overall network is then trained with the loss function in Eqn. (23), but with the physics loss component modified as follows:

$$\mathcal{L}_{physics}(\boldsymbol{\theta}, \boldsymbol{\Lambda}; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} ||\dot{\rho}_i^\theta + \nabla^\theta\cdot(\rho_i^\theta \boldsymbol{\phi}^{\text{spatial}}(\mathbf{x}_i)\boldsymbol{\Lambda})||^2 \tag{32}$$

Applying this approach, we observe that, without coarse-graining the trajectory density data, the model is unable to identify any of the correct coefficients for either velocity field component (Fig. 20 *A,B*). Interestingly however, upon coarsening the data, the correct terms begin to emerge around coarsening level 3. These findings are consistent with the previous approach, which also could not learn any proper coefficients without coarsening the observable data. This suggests that, for learning the macroscopic dynamics underlying an ensemble of trajectories, coarse-graining the observable density data may actually sharpen signals in the data corresponding to the correct terms. A possible explanation for this finding is that the process of coarse-graining smooths the otherwise discontinuous density data, thereby making the data more suited for approximation via neural networks. Visualizations of the effects of coarse-graining on temporal snapshots of Van der Pol trajectory density estimates are shown in Fig. 18*B*.

We also evaluate the eigenspectrum corresponding to the FIM of the output of the divergence operator
$\nabla^\theta\cdot[\rho_i^\theta\boldsymbol{\phi}^{\text{spatial}}(\mathbf{x}_i)\boldsymbol{\Lambda}]$ (the analogue of the regression predictions of the standard approach) with respect to the learned velocity field coefficients $\boldsymbol{\Lambda}$. Notably, none of the dominant eigenvalues appear to be stable under coarse-graining. This finding suggests that no combination of terms in $\boldsymbol{\phi}^{\text{spatial}}$ remain dominant in explaining model behavior as the data is coarsened.

6.3.2 Reduced order Modelling of Lorenz Ensembles

Next, we apply the advection-based approach to the Lorenz system, a dynamical system known for chaotic behavior [23]. The Lorenz system is given by the system of coupled ODEs

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}); \quad \mathbf{f} = \begin{bmatrix}\sigma(y-x) & x(\gamma - kz) - y & kxy - \beta z\end{bmatrix}^\top \tag{33}$$

where σ γ, and β are parameters that control the shape of the attractor, and k is a scale factor used to re-scale the position coordinates. We choose $\sigma = 10$, $\gamma = 28$ and $\beta = 8/3$, and shrink the position coordinates by a factor of $k = 10$. Unlike the Van der Pol system, the Lorenz system does not have a limit cycle, and instead exhibits two fixed point attractors (Fig. 21*A*).

To obtain trajectory densities of the Lorenz system, we first simulated $N = 300,000$ trajectories of the Lorenz system with initial conditions sampled uniformly from the surface of a sphere of radius 2 at the center of the spatial domain considered (see Table 1). We then estimate trajectory densities over time on a discretized x-y-z grid. Visualization of temporal snapshots of estimate trajectory density before and after coarse-graining are shown in Figs. 20*B, C*.

Preliminary attempts at discovering the reduced order Lorenz system via the advection-based approach were unsuccessful, and yielded significantly incorrect coefficients. A possible reason for this is the complex shape of the Lorenz attractor. While the Van der Pol oscillator's limiting behavior can at least be crudely modeled by an ellipse, capturing the dynamics that give rise to the Lorenz attractor's complex shape is a more challenging task.

To make the task of discovering a reduced order model of the Lorenz trajectory density easier, we prune all coefficients in $\mathbf{\Lambda}$ that do not correspond to terms in the ground truth dynamics $\mathbf{f}$. After pruning irrelevant coefficients, the advection-based approach becomes somewhat successful at identifying the correct coeffficients (Fig. 22). Furthermore, the accuracy of learned coefficients improves through coarse-graining, consistent with what was found with the Van der Pol system.

6.3.3 Limitations of the advection-based approach

One drawback of the advection-based approach is that it is difficult to refine learned coefficients $\mathbf{\Lambda}$ after training. While the standard approach of directly learning coefficients of nonlinearities of the state variable allows for post-training coefficient tuning via techniques such as the Forward-Stagewise algorithm, these methods are inapplicable for the advection-based approach. Specifically, the advection-based approach does not learn linear regression coefficients directly. Instead, it learns parameters that must first undergo differentiation (specifically, the divergence operator) before model outputs can be produced.

One simple alternative for refining parameter estimates is to threshold learned coefficients so that all parameter values below a certain threshold are set to 0. While this does indeed improve sparsity, it is an imperfect method as it is possible for terms with tiny coefficients to be dropped that would otherwise contribute greatly to overall dynamics. A concrete example of this phenomena involves the Kuramoto model of coupled oscillators. While the original model contains only first-order derivatives in time of the state variables, perturbing the model with a term involving a second-order time derivative drastically changes the model's behavior [24]. In particular, it has been proven analytically that even a vanishingly small second-order term can fundamentally change the nature of the Kuramoto model's attractors, causing a supercritical bifurcation to become a subcritical bifurcation [25].

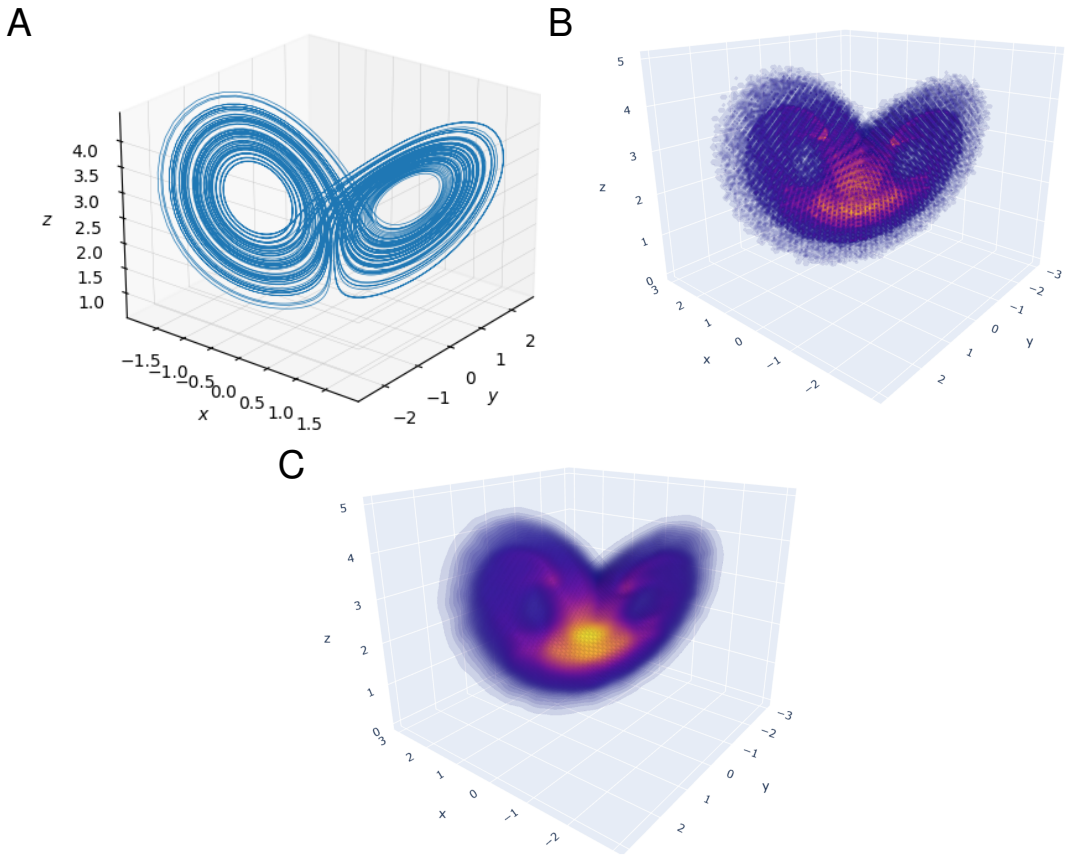

Figure 21: **(A)** The attractor of the re-scaled Lorenz system, generated with the initial condition $\mathbf{x}(0) = (0, 1, 1.05)$. **(B)** A snapshot of estimated trajectory density of the Lorenz system without coarse-graining at $t = 5$. **(C)** The estimated trajectory density of the Lorenz system at level 3 coarsening (spatially isotropic Gaussian blurring with $\sigma = 2.0$). Density estimates were produced from $N = 300,000$ Lorenz trajectories with initial conditions sampled uniformly from the surface of a sphere of radius 2 centered at $(0, 0, 2.5)$.

PDE (Approach)	Spatial Domain	Temporal Domain	# Coeffs.
Burgers'	$x \in [-8, 8]_{n_x=256}$	$t \in [0, 10]_{n_t=101}$	16
Korteweg-De Vries	$x \in [-16, 16]_{n_x=256}$	$t \in [0, 5]_{n_t=101}$	16
VDP Ensemble (Standard)	$x, y \in [-6, 6]_{n_x=n_y=512}$	$t \in [0, 10]_{n_t=201}$	240
VDP Ensemble (Advection)	$x, y \in [-6, 6]_{n_x=n_y=512}$	$t \in [0, 10]_{n_t=201}$	10×2
Lorenz Ensemble (Advection)	$x, y \in [-3, 3]_{n_x=n_y=100}$, $z \in [0, 5]_{n_z=100}$	$t \in [0, 10]_{n_t=1001}$	20×3

Table 1: A table summarizing all PDEs analyzed in this work. For each PDE learned, we report the spatial and temporal domains considered, as well as the number of coefficients learned, corresponding to the size of the candidate function library.

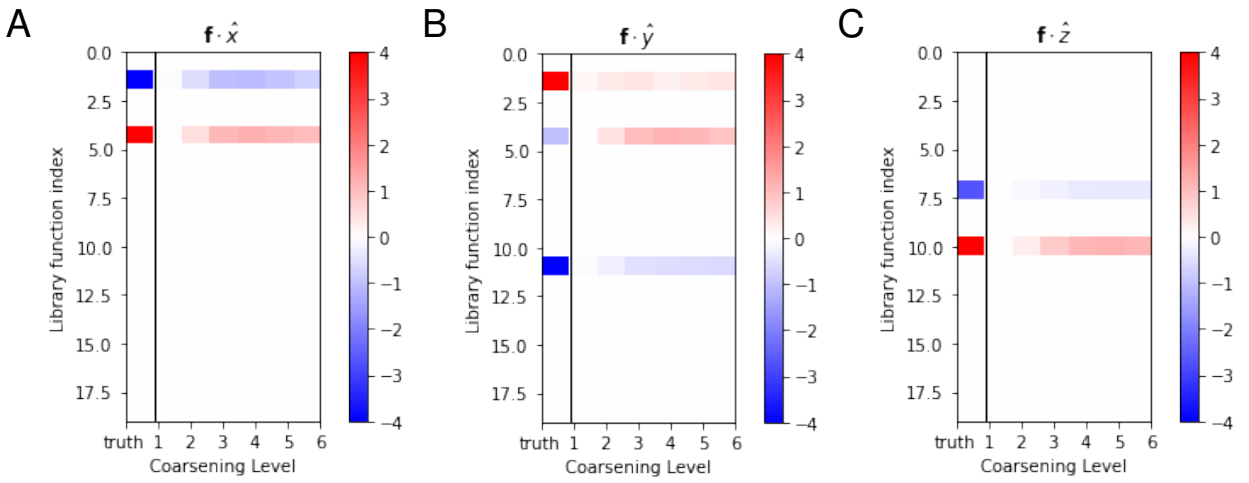

Figure 22: Learned coefficients for the Lorenz density velocity field versus level of coarsening, shown for **(A)** the x-component $\mathbf{f} \cdot \hat{x}$, **(B)** the y-component $\mathbf{f} \cdot \hat{y}$, and **(C)** the z-component $\mathbf{f} \cdot \hat{z}$. During training, coefficients for terms not present in the ground truth solution are set to 0.

7 Conclusion

In this work, we explored techniques for constructing reduced-order models of phenomena from data. By investigating the geometric properties of models from statistical physics, we demonstrate that coarse-graining observable data can amplify model sloppiness, thereby eliminating parameter combinations that do not contribute to macroscopic model behavior. In addition, we study approaches for discovering the functional forms of partial differential equations from data, and how such approaches perform under coarse-graining. Importantly, we find that some, but not all learned models of discovered PDEs exhibit FIM eigenvalues that stay stable under coarse-graining. This suggests that there are a subclass of PDE systems that are well-suited for model order reduction techniques. Lastly, we investigate techniques for identifying macroscopic dynamics underlying large ensembles of individual trajectories. To that end, we extend the approach of physics-informed neural networks and introduce a method for directly learning the coefficients of an advection PDE's underlying velocity field. Notably, we demonstrate that discoverability of a system's underlying macroscopic dynamics can actually improve upon coarse-graining observable data.

Our findings suggest a number of directions for future study. Firstly, while we have studied PDE system identification techniques via observing FIM eigenvalues of learned models, future work should also investigate how the properties of dominant FIM eigenvectors may prove fruitful for model order reduction. In particular, it is possible that projecting learned coefficients along directions of stiff FIM eigenvectors may yield useful information about the dominant or stable modes that underlie a system's dynamics. In addition, future work should investigate whether the advection-based approach introduced in this work remains effective for learning reduced order models of ensembles of microscopic observables from real-world data, such as the flow of ions in a fluid.

Code Link

Code used to generate the plots in this work can be found at https://github.com/wqian0/CIS498Thesis.

References

[1] Zhixin Lu, Brian R. Hunt, and Edward Ott. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, Jun 2018.

[2] Markus Quade, Markus Abel, Kamran Shafi, Robert K. Niven, and Bernd R. Noack. Prediction of dynamical systems by symbolic regression. *Phys. Rev. E*, 94:012214, Jul 2016.

[3] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[4] Katherine N. Quinn, Michael C. Abbott, Mark K. Transtrum, Benjamin B. Machta, and James P. Sethna. Information geometry for multiparameter models: New perspectives on the origin of simplicity, 2021.

[5] Benjamin B. Machta, Ricky Chachra, Mark K. Transtrum, and James P. Sethna. Parameter space compression underlies emergent theories and predictive models. *Science*, 342(6158):604–607, Nov 2013.

[6] Mark K. Transtrum, Benjamin Machta, Kevin Brown, Bryan C. Daniels, Christopher R. Myers, and James P. Sethna. Sloppiness and emergent theories in physics, biology, and beyond, 2015.

[7] Mark K. Transtrum, Benjamin B. Machta, and James P. Sethna. Geometry of nonlinear least squares with applications to sloppy models and optimization. *Physical Review E*, 83(3), Mar 2011.

[8] Tim van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[9] Thomas George. NNGeometry: Easy and Fast Fisher Information Matrices and Neural Tangent Kernels in PyTorch, February 2021.

[10] James P. Sethna, Matthew K. Bierbaum, Karin A. Dahmen, Carl P. Goodrich, Julia R. Greer, Lorien X. Hayden, Jaron P. Kent-Dobias, Edward D. Lee, Danilo B. Liarte, Xiaoyue Ni, Katherine N. Quinn, Archishman Raju, D. Zeb Rocklin, Ashivni Shekhawat, and Stefano Zapperi. Deformation of crystals: Connections with statistical physics. *Annual Review of Materials Research*, 47(1):217–246, Jul 2017.

[11] Philip E. Paré, David Grimsman, Alma T. Wilson, Mark K. Transtrum, and Sean Warnick. Model boundary approximation method as a unifying framework for balanced truncation and singular perturbation approximation, 2019.

[12] Barry A Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 94(10):937–959, 1987.

[13] Ulli Wolff. Comparison between cluster monte carlo algorithms in the ising model. *Physics Letters B*, 228(3):379–382, 1989.

[14] Johannes Martinus Burgers. *The nonlinear diffusion equation: asymptotic solutions and statistical problems*. Springer Science & Business Media, 2013.

[15] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[16] Karsten Ahnert and Markus Abel. Numerical differentiation of experimental data: local versus global methods. *Computer Physics Communications*, 177(10):764–774, Nov 2007.

[17] M. Raissi, 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, Feb 2019.

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[19] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12(1), Oct 2021.

[20] Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

[21] Mayur P. Bonkile, Ashish Awasthi, C. Lakshmi, Vijitha Mukundan, and V S Aswin. A systematic literature review of burgers' equation with recent advances. *Pramana*, 90:1–21, 2018.

[22] Jacques GL Laforgue and Robert E O'Malley, Jr. Shock layer movement for burgers' equation. *SIAM Journal on Applied Mathematics*, 55(2):332–347, 1995.

[23] Shyi-Kae Yang, Chieh-Li Chen, and Her-Terng Yau. Control of chaos in lorenz system. *Chaos, Solitons & Fractals*, 13(4):767–780, 2002.

[24] Simona Olmi. Chimera states in coupled kuramoto oscillators with inertia. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(12):123125, 2015.

[25] J. Barré and D. Métivier. Bifurcations and singularities for coupled oscillators with inertia and frustration. *Phys. Rev. Lett.*, 117:214102, Nov 2016.