Emergent Spaces for Coupled Oscillators

Thomas N. Thiem[1], Mahdi Kooshkbaghi[2], Tom Bertalan[3], Carlo R. Laing[4] and Ioannis G. Kevrekidis[5]*

[1] Department of Chemical and Biological Engineering, Princeton University, Princeton, NJ, United States, [2] Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, United States, [3] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States, [4] School of Natural and Computational Sciences, Massey University, Auckland, New Zealand, [5] Chemical and Biomolecular Engineering and Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, United States

Systems of coupled dynamical units (e.g., oscillators or neurons) are known to exhibit complex, emergent behaviors that may be simplified through coarse-graining: a process in which one discovers coarse variables and derives equations for their evolution. Such coarse-graining procedures often require extensive experience and/or a deep understanding of the system dynamics. In this paper we present a systematic, data-driven approach to discovering "bespoke" coarse variables based on manifold learning algorithms. We illustrate this methodology with the classic Kuramoto phase oscillator model, and demonstrate how our manifold learning technique can successfully identify a coarse variable that is one-to-one with the established Kuramoto order parameter. We then introduce an extension of our coarse-graining methodology which enables us to learn evolution equations for the discovered coarse variables via an artificial neural network architecture templated on numerical time integrators (initial value solvers). This approach allows us to learn accurate approximations of time derivatives of state variables from sparse flow data, and hence discover useful approximate differential equation descriptions of their dynamic behavior. We demonstrate this capability by learning ODEs that agree with the known analytical expression for the Kuramoto order parameter dynamics at the continuum limit. We then show how this approach can also be used to learn the dynamics of coarse variables discovered through our manifold learning methodology. In both of these examples, we compare the results of our neural network based method to typical finite differences complemented with geometric harmonics. Finally, we present a series of computational examples illustrating how a variation of our manifold learning methodology can be used to discover sets of "effective" parameters, reduced parameter combinations, for multi-parameter models with complex coupling. We conclude with a discussion of possible extensions of this approach, including the possibility of obtaining data-driven effective partial differential equations for coarse-grained neuronal network behavior, as illustrated by the synchronization dynamics of Hodgkin–Huxley type neurons with a Chung-Lu network. Thus, we build an integrated suite of tools for obtaining data-driven coarse variables, data-driven effective parameters, and data-driven coarse-grained equations from detailed observations of networks of oscillators.

Keywords: diffusion maps, manifold learning, geometric harmonics, neural networks, Kuramoto oscillators, coupled systems, complex networks

1. INTRODUCTION

We study coupled systems comprised of many individual units that are able to interact to produce new, often complex, emergent types of dynamical behavior. The units themselves (motivated by the modeling of large, complex neuronal networks) may be simple phase oscillators, or may be much more sophisticated, with heterogeneities and parameter dependence contributing to the emerging behavior complexity. Each unit is typically described by a system of ordinary differential equations (ODEs) (1), where $\mathbf{f}$ is a function of the system state $\mathbf{x}$, time t, and the system parameters $\mathbf{p}$,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t; \mathbf{p}). \tag{1}$$

Examples of such systems from across the scientific domains range from coupled reactor networks (Mankin and Hudson, 1986) and molecular dynamics simulations (Haile, 1997), to the modeling of synchronization and swarming among oscillators (O'Keeffe et al., 2017), and even to the oscillations of the millennium bridge (Strogatz et al., 2005). For large coupled oscillator ensembles, numerically evolving the system state can be computationally expensive; yet several such systems of practical interest feature an underlying structure that allows them to be described through a small collection of *coarse variables* or *order parameters* whose dynamic evolution may be described in simpler, reduced terms. It is also observed that the dynamics of these coarse variables may only practically depend on a reduced set of parameters, the "effective" parameters of the system, which themselves are specific, possibly non-linear, combinations of the original, detailed system parameters. In general, uncovering such coarse, descriptive variables and parameters, and approximating their effective dynamic evolution is a significant undertaking, typically requiring extensive experience, deep insight, and painstaking theoretical and/or computational effort.

Here, we present and illustrate an alternative, automated, data-driven approach to finding emergent coarse variables and modeling their dynamical behavior. As part of our methodology we make use of manifold learning techniques, specifically diffusion maps (DMAPs) (Coifman and Lafon, 2006a), to systematically infer tailor-made descriptive variables directly from detailed dynamical data itself, without any prior knowledge of the underlying models/equations. This stands in stark contrast to established, intuition-based or equation-based coarse-graining methods, which rely on a combination of system knowledge, experience and strongly model-dependent mathematical techniques to invent or derive the coarse variables/equations.

We begin our presentation with a brief outline of our key manifold learning tool, the diffusion maps technique, followed by an abridged introduction to the associated geometric harmonics function extension method (Coifman and Lafon, 2006b). Following these introductions, we present a demonstration of our data-driven methodology for coarse variable identification. After this, we consider the time dependent behavior of coarse variables, and show how their corresponding coarse dynamic evolution equations can be learned from time series of dynamical observation data by virtue of an artificial neural network architecture templated on numerical time integration schemes. We compare this approach to typical finite difference based methods complemented with geometric harmonics. Finally, we illustrate how manifold learning techniques can be used to find "effective," reduced sets of parameters in multi-parameter coupled systems.

Motivated by the intended applications of our methodology to neurological systems, we choose to illustrate our techniques with one of the simplest neurobiologically salient models available, the classic Kuramoto coupled phase oscillator model (Kuramoto, 1975, 1984). The Kuramoto model has become a popular choice to study synchronization (Schmidt et al., 2014) and network topology in neuroscience (Rodrigues et al., 2016). At first glance the scalar phase dynamics of the model may seem to be insufficient or highly restrictive. However, the phase reduction approach has become a standard technique in computational neuroscience (Ermentrout and Kopell, 1986, 1990; Brown et al., 2004; Tass, 2007; Guckenheimer and Holmes, 2013) providing a link between computational models of neurons and models of weakly coupled phase oscillators. Indeed, a classical example of this approach is in the characterization of a simple (class I) spiking neuron as a one dimensional phase oscillator (Ermentrout and Kopell, 1986; Ermentrout, 1996). Recently, phase based measures of synchrony have become a typical feature in the characterization of large-scale experimental neuroscience signals (Tass et al., 1998; Varela et al., 2001; Breakspear, 2002; Stam et al., 2007; Penny et al., 2009). Advances in the field of neural mass models have furnished standard approcahes to describe the interaction between excitatory and inhibitory neurons, such as the Wilson–Cowan model (Wilson and Cowan, 1973). Weakly-coupled Wilson–Cowan oscillators have subsequently been shown to exhibit similar interaction dynamics to weakly-coupled Kuramoto oscillators (Izhikevich and Hoppensteadt, 1997). The main deficiency of the Kuramoto model is the lack of a spatial embedding for the oscillators; however numerous modifications of the interaction term of the model have been proposed to circumvent this difficulty, such as time delays, distance-dependent transmission delays, finite-support wavelet-like spatial kernels, and second order phase interaction curves (Breakspear et al., 2010). These modifications yield rich spatiotemporal behaviors, such as traveling rolls and concentric rings (Jeong et al., 2002), which are similar to those observed *in vivo* (Freemann, 1975; Prechtl et al., 1997; Lam et al., 2000; Du et al., 2005; Rubino et al., 2006). Furthermore, these modified models have been used to understand dynamical behavior on cortical-like sheets (Honey et al., 2007; Deco et al., 2009), simulate the BOLD signal (Cabral et al., 2011), and study hypersynchronous neural activity (Schmidt et al., 2014).

We make use of the classic Kuramoto coupled phase oscillator model and its variations throughout this paper as prototypical examples to showcase our methodology. In our concluding discussion and future work section we also mention the possibility of discovering effective emergent *partial differential equation* descriptions of heterogeneous network dynamics,

illustrated through the synchronization of Hodgkin–Huxley type neurons on a Chung-Lu type network.

2. DIFFUSION MAPS

Introduced by Coifman and Lafon (2006a), the diffusion maps (DMAPs) method and associated algorithms form part of a class of dimensionality reduction approaches, techniques which are used to find the intrinsic dimensionality of high dimensional data. Specifically, the diffusion map (DMAP) algorithm is a manifold learning technique that seeks to address the problem of parameterizing d-dimensional manifolds embedded in $\mathbb{R}^n$ based on data, with $d < n$. It accomplishes this by constructing a (discretized) Laplace operator on the data, such that the operator's eigenfunctions define embedding coordinates for the manifold. An algorithm for numerically constructing this operator is provided by Coifman and Lafon (2006a).

At the heart of the Laplace operator construction lies the kernel, k. The kernel describes the affinity/similarity between data points and serves to define the local geometry of the underlying manifold. This information is frequently stored in a kernel matrix K, where K_{ij} is the kernel evaluated on data points i and j. As the specific parametrization intimately depends on the data geometry, it is essential to choose a kernel that captures the relevant properties of the data. An example of a typical Gaussian kernel with the Euclidean distance is shown in (2),

$$K_{ij} = k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\epsilon^2}\right) \quad i, j = 1, \dots, n, \quad (2)$$

where ϵ is a tunable kernel bandwidth parameter and $\| \cdot \|_2$ is the Euclidean norm. By selecting different kernel bandwidth parameters it is possible to examine features of the data geometry at different length scales. Several heuristics are available to select the value of the kernel bandwidth parameter (Dsilva et al., 2018); here we select the median of the pairwise distances between the data points x_i as a starting point for our ϵ tuning.

After defining a kernel, the Laplace operator can be approximated as follows. First, one forms the diagonal matrix D with $D_{ii} = \sum_j K_{ij}$, which is then used to construct the matrix $\widetilde{K} = D^{-\alpha} K D^{-\alpha}$, where α is an algorithm parameter; the choice of α determines the form of the operator that is approximated, which in turn dictates the influence of the sampling density of the data on the parametrization of the underlying manifold. Common choices are: (a) $\alpha = 0$, the normalized graph Laplacian (the one most influenced by the sampling density and useful for uniformly sampled manifolds), (b) $\alpha = 0.5$, Fokker–Planck diffusion, and (c) $\alpha = 1$, the Laplace–Beltrami operator (which removes the influence of the sampling density). Next, one constructs the diagonal matrix $\widetilde{D}$ with $\widetilde{D}_{ii} = \sum_j \widetilde{K}_{ij}$ and uses this to form the matrix $A = \widetilde{D}^{-1}\widetilde{K}$. The eigenvectors of A provide new *intrinsic* coordinates for the data points on the manifold.

A challenge of using the eigenfunctions of a Laplace operator to parameterize the manifold on which the data lie is the existence of "repeated" coordinates, that is, coordinates which

parameterize an already discovered direction along the manifold, referred to as *harmonics* (Dsilva et al., 2018). These harmonics provide no additional dimensional information, and should be ignored in an effort to discover a minimal embedding. A systematic computational method has been developed for automatically identifying such harmonics via a local linear regression on the eigenfunctions (Dsilva et al., 2018).

Before the local linear regression method can be applied, the eigenfunctions of the Laplace operator must be sorted by their corresponding eigenvalues, all of which are real, from greatest to smallest. The main idea behind the local linear regression method is that the repeated eigenfunctions, the harmonics, can be represented as functions of the fundamental eigenfunctions, which correspond to unique directions. The local linear regression method checks for this functional relationship by computing a local linear fit (3) of a given eigenfunction $\phi_k \in \mathbb{R}^d$ (where d is the number of data points) as a function of the previous eigenfunctions $\Phi_{k-1} = [\phi_1 \cdots \phi_{k-1}]^T$ (those previously discovered) for each data index i.

$$\phi_k(i) \approx \alpha_k(i) + \beta_k^T(i)\Phi_{k-1}(i) \quad (3)$$

The coefficients of the fit, $\alpha_k(i) \in \mathbb{R}$ and $\beta_k(i) \in \mathbb{R}^{k-1}$, vary over the domain (indexed by the data points i) and are found by solving the following optimization problem (4).

$$\alpha_k(i), \beta_k(i) = \underset{\alpha, \beta}{\operatorname{argmin}} \sum_{j \neq i} K(\Phi_{k-1}(i), \Phi_{k-1}(j))(\phi_k(j) \\ - (\alpha + \beta^T \Phi_{k-1}(j)))^2 \quad (4)$$

The weighting kernel K for the local linear regression (4) is responsible for the domain dependence of the coefficients and is typically Gaussian (Dsilva et al., 2018), such as in (5). For example

$$K(\Phi_{k-1}(i), \Phi_{k-1}(j)) = \exp\left(-\frac{\|\Phi_{k-1}(i) - \Phi_{k-1}(j)\|_2^2}{\epsilon_{reg}^2}\right), \quad (5)$$

where ϵ_{reg} is a tunable scale for the kernel of the regression. Choosing ϵ_{reg} to be one third of the median of the pairwise distances between the eigenfunctions $\Phi_{k-1}(i)$ is recommended as a starting point (Dsilva et al., 2018).

The quality of the fit is assessed by the normalized leave-one-out cross-validation error (6) or the residual r_k of the fit. Values of the residual near zero indicate an accurate approximation of ϕ_k from Φ_{k-1}, indicative of a harmonic eigenfunction, while values near one correspond to a poor fit, and are therefore suggestive of a significant, informative eigenfunction corresponding to a new direction in the data. Thus, by computing residual values for a collection of the computed eigenfunctions, it is possible to identify the fundamentals and the harmonics in a systematic way,

$$r_k = \sqrt{\frac{\sum_{i=1}^n (\phi_k(i) - (\alpha_k(i) + \beta_k(i)^T \Phi_{k-1}(i)))^2}{\sum_{i=1}^n (\phi_k(i))^2}}. \quad (6)$$

Throughout this paper we employ the DMAPs algorithm to identify and parameterize manifolds as well as the local linear regression method to identify the significant, fundamental (non-harmonic) eigenfunctions. In all cases, we use $\alpha = 1$ to remove the influence of the sampling density of the data, and the Gaussian kernel with the Euclidean distance defined in (2) as our similarity measure. For the local linear regression, we utilize a Gaussian kernel (5) and select ϵ_{reg} as one third of the median of the pairwise distances between the Φ_{k-1} as our regression kernel scale, in accordance with Dsilva et al. (2018).

3. GEOMETRIC HARMONICS

Consider a set X sampled from a manifold $\bar{X} \subset \mathbb{R}^N$ with finite measure $\mu(X) < \infty$ for some measure μ. Let us assume that we have a real valued function $F : X \to \mathbb{R}$ defined on X. Geometric harmonics is a tool for extending F to points on the original manifold $\bar{X}$, namely, it provides a way to find a new function $f : \bar{X} \to \mathbb{R}$, which can be considered as an extension of F to $\bar{X}$ (Coifman and Lafon, 2006b).

The first step of computing the geometric harmonics extension is to define a kernel $k : \bar{X} \times \bar{X} \to \mathbb{R}$, where k need not be the same as in (2). The kernel must be symmetric and bounded on the data set. An example of such a kernel is the Gaussian kernel with the Euclidean distance defined in (2). The main features of this kernel that are required are: (a) the ability to easily evaluate it on out-of-sample data points, and (b) that its eigenfunctions form a basis of a function space. By utilizing these features it is possible to represent a function in terms of the eigenfunctions of this kernel (as approximated by the data) and then extend it to out-of-sample data points.

Consider n data points at which we know the values of the function of interest, F. We want to approximate F at some m other, new points by d geometric harmonics. We can construct a matrix $K \in \mathbb{R}^{m \times n}$ with entries $K_{ij} = k(x_i, x_j)$, where x_i is an unknown point, x_j is a sampled point with known value of $F(x_j)$, and k is the kernel. The l-th geometric harmonic is defined by

$$\Phi_i^{(l)} = \Phi^{(l)}(x_i) = \sum_{j=1}^{n} K_{ij} \psi_j^{(l)} \lambda_l^{-1}, \tag{7}$$

where $\psi^{(l)}$ is the l-th column of the matrix $\Psi \in \mathbb{R}^{n \times d}$, the first d eigenfunctions of the kernel matrix, and $\lambda_l \in \Lambda = diag(\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^{d \times d}$ is the corresponding eigenvalue. The extension f of the known function F to the set of new points is then computed as (Coifman and Lafon, 2006b)

$$f = K \Psi \Lambda^{-1} \Psi^T F. \tag{8}$$

Any observable of the data (e.g., the phase or the amplitude or any state variable for each particular oscillator in our network) is a function on the low-dimensional manifold, and can therefore be extended out-of-sample through geometric harmonics. This provides us a systematic approach for translating back-and-forth between physical observations and coarse-grained "effective" or "latent space" descriptions of the emergent dynamics.

4. IDENTIFYING COARSE VARIABLES (ORDER PARAMETERS)

Here we illustrate how manifold learning techniques, in this case diffusion maps, can be used to identify coarse variables for coupled oscillator systems directly from time series data. We consider the simple Kuramoto model with sinusoidal coupling as a test case, to demonstrate how we can learn a variable that is equivalent to the typical Kuramoto order parameter, R.

4.1. The Kuramoto Model

The Kuramoto model is a classical example of limit cycle oscillators that can exhibit synchronizing behavior. The basic version of the model consists of a number of heterogeneous oscillators $(i = 1, \ldots, N)$, each with their own natural frequency ω_i, selected from a distribution $g(\omega)$, that interact through a coupling term f. The coupling term is typically sinusoidal and can be expressed as a function of the difference in phases between oscillators $f(\theta_i - \theta_j)$.

The strength and presence of coupling among the oscillators is expressed by the coupling matrix A. Each element of the matrix A_{ij} $(i, j = 1, \ldots, N)$ is a pairwise strength quantifying the influence of oscillator j on oscillator i. A general Kuramoto model with a sinusoidal coupling function can be written as (9)

$$\frac{d\theta_i}{dt} = \omega_i + \frac{1}{N} \sum_{j=1}^{N} A_{ij} \sin(\theta_j - \theta_i) \quad \text{for } i = 1, \ldots, N. \tag{9}$$

Originally, Kuramoto considered a mean-field coupling approximation (Strogatz, 2000), $A_{ij} = K > 0$ where K is a global coupling constant, which results in the following simplified all-to-all coupled model
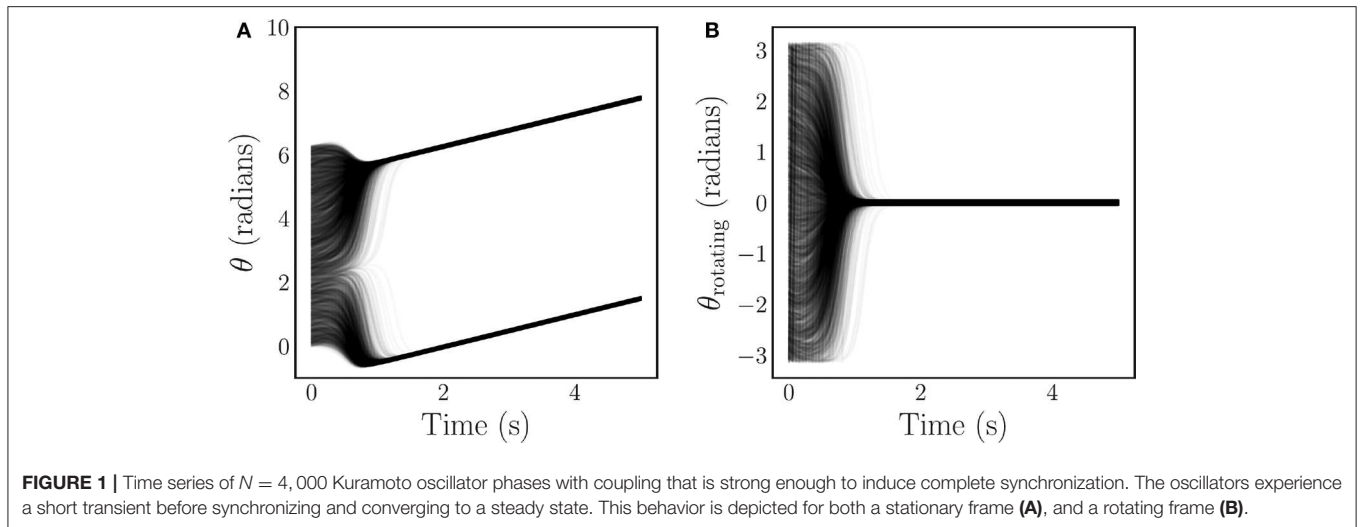
$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_i) \quad \text{for } i = 1, \ldots, N. \tag{10}$$

The degree of phase synchronization of the oscillators can be expressed in terms of the complex-valued order parameter (a coarse variable) introduced by Kuramoto as follows (Kuramoto, 1984)

$$R(t) e^{i\psi(t)} = \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j(t)}, \tag{11}$$

where $0 \leq R(t) \leq 1$ is the time-dependent phase coherence, and $\psi(t)$ is the average phase. Values of the phase coherence near one correspond to phase synchronization of the oscillators, while values near zero imply disorder. An example of the typical synchronizing behavior of the Kuramoto model is illustrated for both a stationary reference frame **Figure 1A**, and a rotating reference frame **Figure 1B**. Note that in the rotating frame the frequency synchronization induces a steady state.

In general, if the coupling is chosen such that the system exhibits complete synchronization (characterized by the lack

FIGURE 1 | Time series of $N = 4{,}000$ Kuramoto oscillator phases with coupling that is strong enough to induce complete synchronization. The oscillators experience a short transient before synchronizing and converging to a steady state. This behavior is depicted for both a stationary frame **(A)**, and a rotating frame **(B)**.

of "rogue" or unbound oscillators), then a steady state exists in a rotating reference frame. Typically, the rotating frame transformation is taken to be $\theta_i(t) \mapsto \theta_i(t) - \psi(t)$, in which $\psi(t)$ is the average phase. This transformation results in a system for which the new average phase is zero, yielding a real-valued order parameter. The order parameter in this rotating frame is given by

$$R(t) = \frac{1}{N} \sum_{j=1}^{N} e^{i(\theta_j(t) - \psi(t))}. \quad (12)$$

Throughout the remainder of this paper we employ the simple all-to-all coupled Kuramoto model and its variations, some with more complicated couplings, to illustrate our methodology. We emphasize the synchronizing behavior of the Kuramoto model as it facilitates coarse-graining, and repeatedly make use of this property to simplify our calculations throughout the rest of the paper.

4.2. Order Parameter Identification

For this section we consider the simple Kuramoto model with all-to-all mean-field coupling (10). Our goal is to: (a) first identify a coarse variable from time series of phase data; and then (b) demonstrate that this discovered variable is equivalent to the established Kuramoto order parameter. Throughout, we consider a rotating frame in which only the magnitude of the order parameter R varies in time. This in turn enables us to neglect the angle ψ, as R suffices to capture the time-dependent behavior.

We begin our analysis by simulating 8,000 Kuramoto oscillators ($N = 8{,}000$) with a coupling constant $K = 2$, and frequencies drawn from a Cauchy distribution (13) with $\gamma = 0.5$.
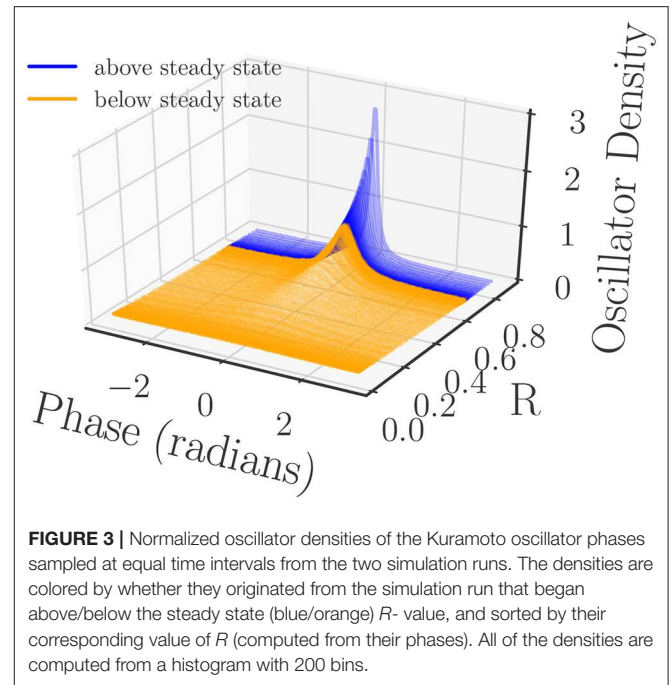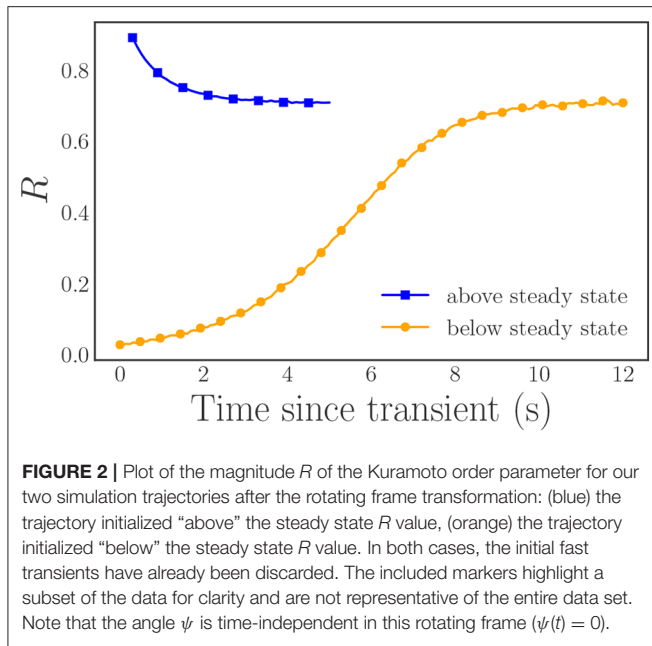
$$g(\omega; \gamma) = \frac{\gamma/\pi}{\omega^2 + \gamma^2} \quad (13)$$

In order to reduce the finite sample noise in R, we use inverse transform sampling, with equally spaced values over the interval $[0 + \epsilon, 1 - \epsilon]$ with $\epsilon \approx 2.5 \times 10^{-4}$, to generate our frequencies

in a systematic and symmetric manner. We select ϵ to place a cap on the maximum absolute value of the frequencies in order to facilitate numerical simulation. For these parameter values, R exhibits an attracting, stable steady state with $R_{\infty} \approx 0.71$. Thus, in order to sample the entire range of potential R-values, we select two different sets of initial conditions for our simulations, with one "above" the steady state synchronization R value, and the other "below" it. For the first case, $R = 1$, we set all of the initial oscillator phases to π and for the second case, $R \approx 0$, we use equally spaced initial phases over $[0, 2\pi]$. In both cases, we integrate our system of oscillator ODEs with SciPy's Runge–Kutta integrator (*solve_ivp* with the RK45 integrator) with the absolute and relative tolerances set to 10^{-7} and 10^{-4}, respectively. After discarding the initial transients, we transform the phase data into a rotating frame and then sample it at discrete, equidistant time steps to form our time series data (**Figure 2**).

Before applying the DMAPs algorithm to these time series of phase data to identify a coarse variable, we need to select a suitable kernel. It is crucial to select a kernel that will compare the relevant features of the data in order to produce a meaningful measure of the similarity between data points. In this case, it is important to choose a kernel that measures the degree of clustering or equivalently the phase synchronization of the oscillators *in a way that is invariant to permutations of the oscillator identity*, as a relabeling of the oscillators should correspond to the same degree of synchronization.

Since the synchronization of the oscillators is related to how the oscillator phases group together, it is a natural choice to consider the phase density as a meaningful observable. The oscillator phase density captures phase clustering while being permutation independent, and should thus provide a meaningful similarity measure between system snapshots. Therefore, we first pre-process the phase data by computing the density of the oscillators over the interval $[-\pi, \pi]$ before defining the kernel. We approximate this density with a binning process (histogram) that uses 200 equally spaced bins over the interval $[-\pi, \pi]$. As part of the binning process we are careful to reduce the phases of

FIGURE 2 | Plot of the magnitude R of the Kuramoto order parameter for our two simulation trajectories after the rotating frame transformation: (blue) the trajectory initialized "above" the steady state R value, (orange) the trajectory initialized "below" the steady state R value. In both cases, the initial fast transients have already been discarded. The included markers highlight a subset of the data for clarity and are not representative of the entire data set. Note that the angle ψ is time-independent in this rotating frame ($\psi(t) = 0$).

FIGURE 3 | Normalized oscillator densities of the Kuramoto oscillator phases sampled at equal time intervals from the two simulation runs. The densities are colored by whether they originated from the simulation run that began above/below the steady state (blue/orange) R- value, and sorted by their corresponding value of R (computed from their phases). All of the densities are computed from a histogram with 200 bins.

the oscillators mod 2π to ensure that all of them are captured in the $[-\pi, \pi]$ interval (**Figure 3**).

Using time series of density data instead of individual phase data simplifies the comparison of different snapshots, and thus facilitates the selection of the kernel in the DMAPs algorithm. For this we select the standard Gaussian kernel with the Euclidean metric for the comparison of our oscillator density vectors ($d \in \mathbb{R}^{200}$) with diffusion map tuning parameters of $\alpha = 1$ selecting the Laplace-Beltrami operator, and $\epsilon \approx 1.54$ as the kernel bandwidth parameter. This results in a single diffusion map coordinate (eigenfunction) that is deemed significant by the local linear regression method, see **Figure 4A**.

Comparing this diffusion map coordinate ϕ_1 to the Kuramoto order parameter R reveals there is a one-to-one correspondence between the two, as **Figure 4B** clearly illustrates: the plot is monotonic, and its derivative remains bounded away from zero and from infinity. This means that the diffusion map coordinate contains the same information as the established Kuramoto order parameter, and therefore constitutes a suitable coarse variable for this system. This is particularly interesting as we managed to identify this variable directly through manifold learning on the minimally pre-processed phase data, without referencing Kuramoto's order parameter. In this way, we have demonstrated a process that "learns" bespoke coarse variables, circumventing the traditional discovery/invention process.

While this process is both systematic and automated, we must point out that the key step in this process, the choice of the relevant features of the data, still requires a modicum of insight. Different choices of the relevant features of the data, either through pre-processing and/or selection of the kernel, will produce different coarse variables, that may well be "reconcilable," i.e., one-to-one with each other and with R.

5. LEARNING THE DYNAMIC BEHAVIOR OF DATA-DRIVEN COARSE VARIABLES WITH NEURAL NETWORKS

Once descriptive coarse variables have been identified, it is desirable to find descriptions of their behavior (evolution laws, typically in the form of ordinary or partial differential equations, ODEs or PDEs). However, finding analytical expressions for these descriptions ("laws") is often extremely challenging, if not practically infeasible, and may rely on *ad hoc* methods. The process used to learn the analytical equations that describe the behavior of the Kuramoto order parameter in the continuum (infinite oscillator) limit exemplifies these types of difficulties (Ott and Antonsen, 2008). As a result of this process, Ott and Antonsen showed that the continuum limit of the all-to-all coupled Kuramoto model (10) admits an attracting, invariant manifold. For Cauchy distributed frequencies, the conservation PDE governing the oscillator density can be analytically transformed into a pair of ODEs that describe the time evolution of the Kuramoto order parameter along this manifold,

$$
\begin{aligned}
\frac{dR}{dt} &= -\gamma R + \frac{KR}{2}(1 - R^2) \\
\frac{d\psi}{dt} &= 0,
\end{aligned}
\tag{14}
$$

where R, ψ, K, and γ are the phase coherence, average phase, coupling constant, and Cauchy distribution parameter, respectively, with t as time. As this manifold is invariant and attracting, the dynamics of the order parameter can be accurately described by these equations after a short initial transient.
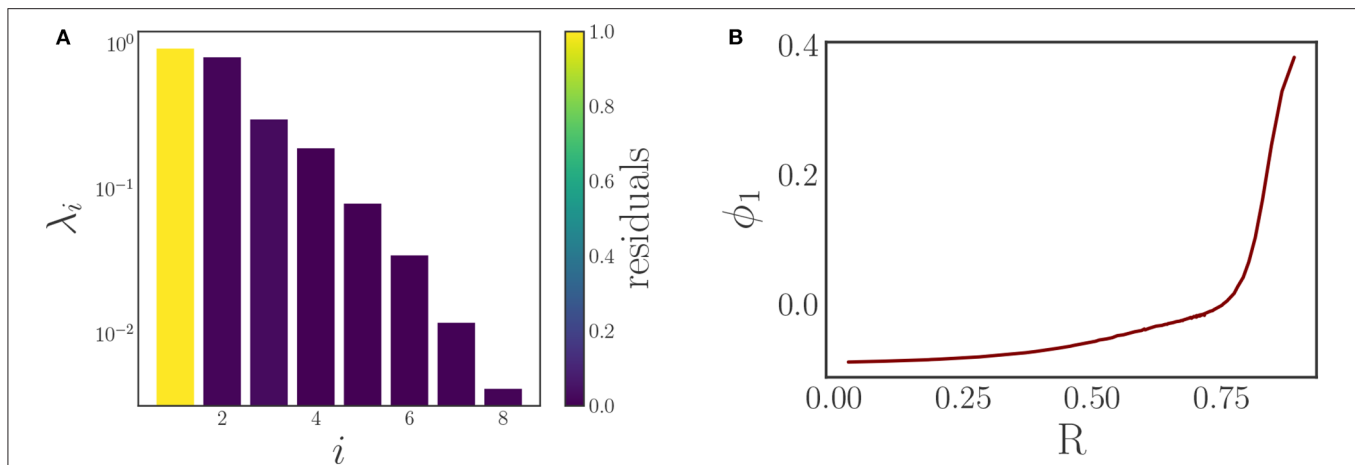
FIGURE 4 | (A) Eigenvalues of the diffusion map operator on the oscillator density dynamic data, sorted in decreasing order and colored by the corresponding residual value produced by the local linear regression algorithm. The first eigenfunction ϕ_1 suffices to parameterize the behavior. **(B)** A plot demonstrating the correspondence between the discovered diffusion map coordinate ϕ_1 and the magnitude of the typical Kuramoto order parameter R. It clearly shows the one-to-one correspondence between the two, verifying that the diffusion map coordinate is an "equivalent" coarse variable to R for this system.

In the following sections, we present an alternative, data-driven approach to learning the behavior of coarse variables directly from time series of observational data. As part of this approach we make use of a recurrent neural network architecture "templated" on numerical time integration schemes, which allows us to learn the time derivatives of state variables from flow data in a general and systematic way. We illustrate this approach through an example in which we learn the aforementioned ODEs (14) that govern the behavior of the Kuramoto order parameter in the continuum limit from data. We then compare the result of our neural network based approach to standard finite differences complemented with geometric harmonics. Throughout this example we only observe the magnitude of the order parameter R and neglect the angle ψ, as the magnitude captures the relevant synchronization dynamics of this model.

5.1. A Neural Network Based on a Numerical Integration Scheme

Numerical integration algorithms for ODEs rely on knowledge of the time derivative in order to approximate a future state. If an analytical formula for the time derivative is not available or unknown, it can be approximated from time series of observations through, say, the use of finite differences, with known associated accuracy problems, especially when the data is scarce. Here, we discuss an alternative, neural network based approach to learning time derivatives (the "right hand sides" of ODEs) from discrete time observations.

Artificial neural networks have gained prominence for their expressiveness and generality, and especially for their ability to model non-linear behavior. These qualities have led to their widespread adoption and use in areas as diverse as image (Simonyan and Zisserman, 2014) and speech recognition (Graves et al., 2013), financial modeling and prediction (Guresen et al., 2011), and general game playing (Silver et al., 2016; Vinyals et al., 2017). For the approximation of dynamical systems,

neural networks have been used for tasks such as the accurate approximation of functions and their derivatives (Cardaliaguet and Euvrard, 1992), system identification and control (Narendra and Parthasarathy, 1990; Subudhi and Jena, 2011), and system modeling (Chow and Li, 2000). A variety of network architectures have been studied for the analysis of dynamical systems, such as feedforward networks (Rico-Martinez et al., 1992), recurrent networks (Chow and Li, 2000), high-order networks (Kosmatopoulos et al., 1995), and multistep networks (Raissi et al., 2018), along with novel training approaches such as the differential evolution approach (Chow and Li, 2000). Here we focus on a feedforward stepping approach, whose utility for accurately modeling system dynamics has been previously demonstrated (Rico-Martinez et al., 1992; Raissi et al., 2018).

This feedfoward approach is a method for approximating the functional form of the right-hand-side of systems modeled through autonomous ODEs (Rico-Martinez et al., 1992; Rico-Martinez et al., 1995; Rico-Martinez and Kevrekidis, 1993). The crux of the approach is to approximate the time derivative of the system with a feedforward (and here, a recurrent) neural network. This neural network approximation can then be used in place of a first-principles based right-hand-side in any initial value solver, such as the Euler or Runge–Kutta methods, to produce a new, neural network based time-stepper (Rico-Martinez et al., 1995). In other words, a neural network architecture templated on a numerical time-stepping processes can be trained to learn an approximation of the right-hand-side of the system equations from pairs of inputs and outputs, where the input is the state $y(t)$ at time t and the output is $y(t + \Delta t)$ at time $t + \Delta t$. By training such a neural network architecture on pairs of state variable observations, $(y(t), y(t + \Delta t))$, a part of the neural network learns an approximation of the right-hand-side of the evolution equation. This is a surrogate model, which subsequently, using any good integration algorithm, can produce a flow that closely matches the training data. Following

successful training, an estimate of the right-hand-side of the evolution equation for any out-of-sample initial system states can be easily accessed by evaluating the neural network directly for these system states.

In addition to its application to learning the right-hand-sides of systems of ODEs, this type of neural network architecture can also be extended to learn the right hand sides of PDEs discretized as systems of ODEs through a method of lines approach (Gonzalez-Garcia et al., 1998). While any explicit integration algorithm that only requires knowledge of the right-hand-side can be used to devise acceptable neural network architectures for learning unknown evolution equations, we will focus here on the fourth order Runge–Kutta algorithm for our analysis and computations.

A schematic of the feedforward recurrent neural network architecture we construct templated on a fourth order, fixed-step Runge–Kutta algorithm is illustrated in **Figure 5**. An important feature of this method with the Runge–Kutta algorithm is that the "black box" neural network evaluating f (a recurrent component of the overall network architecture) is shared between all of the stages of the algorithm (k_1, k_2, k_3, k_4). That is, there is a single copy of the neural "sub"-network f that is evaluated multiple times per time step with different inputs as required by the Runge–Kutta algorithm in order to produce the output.

In the following section, we use this Runge–Kutta scheme to learn the ODE governing the behavior of the magnitude of the Kuramoto order parameter R in a data-driven way.

5.2. Generating the Flow (Training) Data

As discussed in the previous section, we need to generate pairs of flow data of the coarse variable, $(R(t), R(t + \Delta t))$, in order to train the neural network based approach. We begin by considering 2,000 unique (different initial phases and frequencies) simulations of the simple all-to-all coupled Kuramoto model (10) with 8,000 oscillators ($N = 8,000$) with a coupling constant $K = 2$, and independently sampled frequencies ω_i drawn from a Cauchy distribution with $\gamma = 0.5$.

One of the difficulties of generating flow data for the order parameter is adequately sampling the entire range of $R \in [0, 1]$. As the order parameter is not a quantity that we are directly simulating, and instead depends on the phases of the oscillators in a complex way, it is difficult to consistently initialize flows to arbitrary values of R. We surmount this difficulty by means of an initialization integration approach. This initialization approach consists of initializing each of the unique oscillator simulations to easily expressed order parameter values, either $R \approx 0$ (uniformly distributed random initial phases on $[0, 2\pi]$) or $R = 1$ (identical initial phases), and then integrating them for different amounts of time to produce a set of 2,000 variegated starting points for our R-flow data, $R(t)$. These points are the final points of the black initialization trajectories shown in **Figure 6A** and are marked by orange stars. By judiciously selecting the integration times, it is possible to produce a set of initial points $R(t)$ that covers the entire range of possible R-values nearly uniformly, see **Figure 6B**.

After generating the starting points of the flows $R(t)$ with this initialization approach, we proceed to produce the corresponding end points $R(t + \Delta t)$ through a detailed time integration of the oscillator phases. This process consists of first integrating the oscillator phases corresponding to $R(t)$ forward in time for a chosen time step Δt, and then computing the order parameter for the resulting new phases. We select two different reporting time horizons for this integration, $\Delta t = 0.01$, and $\Delta t = 0.05$, to produce two different sets of flow data to assess the sensitivity of our method to the flow time. For all of these simulations, we use Scipy's Runge–Kutta integrator (*solve_ivp* with the RK45 option) to perform the time integration.
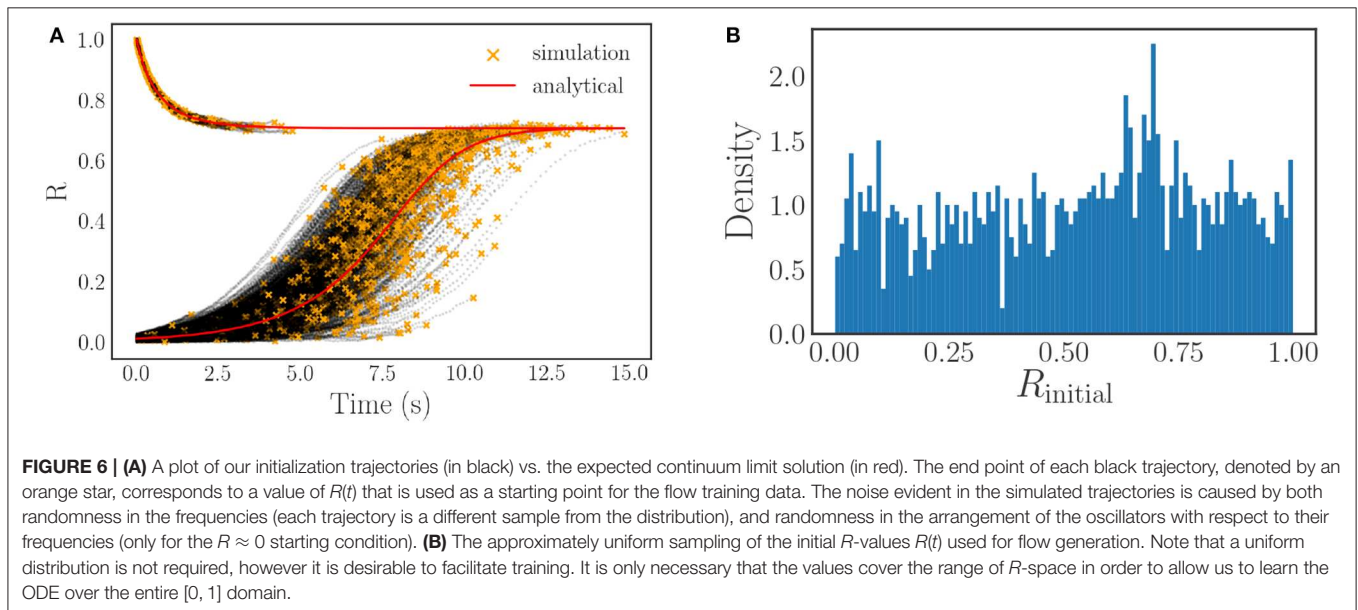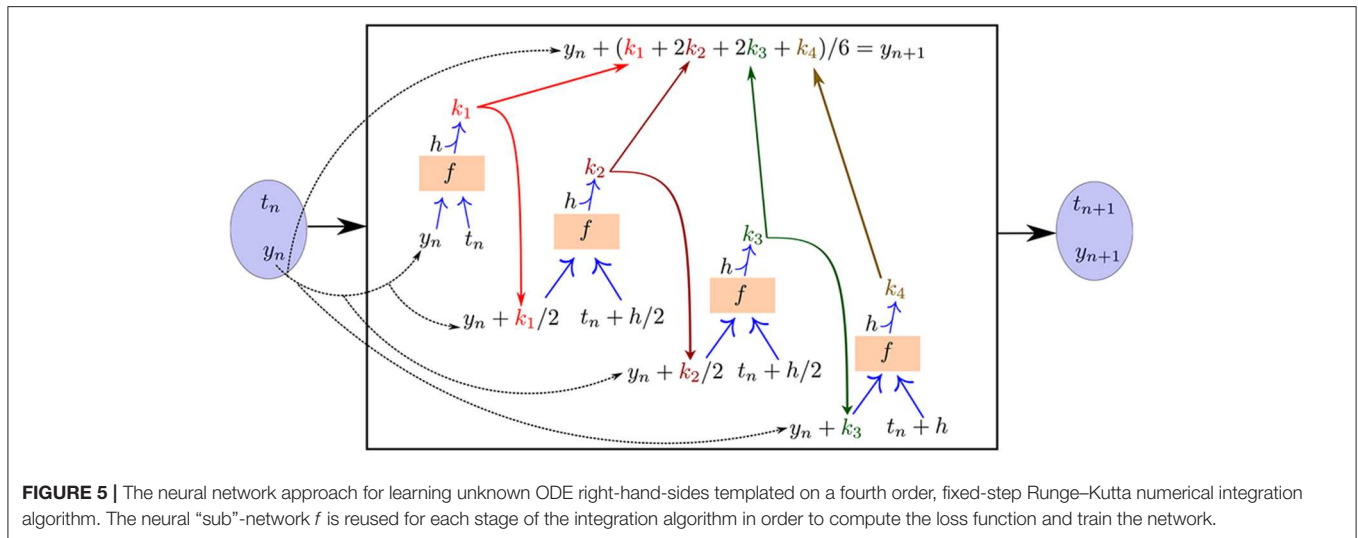
This entire process results in two collections of training data, each consisting of 2,000 pairs of the form $(R(t), R(t + \Delta t))$, with one collection for each choice of Δt. Each of these collections is used individually in the following section to train our neural network to approximate the right-hand-side of the coarse evolution equation for $R(t)$.

5.3. Training the Neural Network Scheme

Our neural network architecture is based on a standard fixed step-size, fourth order Runge–Kutta integration algorithm complemented with a feedforward neural network with 3 hidden layers of 24 neurons each, **Figure 5**. We initialize our network with a uniform Glorot procedure (Glorot and Bengio, 2010) for the kernels, and zeros for the biases. For training, we use TensorFlow's Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3} and a mean squared error loss on the final flow point $R(t + \Delta t)$ with full batch training (all of the training data used in each training epoch).

We train one neural network for each time horizon data set, $\Delta t = 0.01, 0.05$, to investigate the sensitivity of the right-hand-side estimation to the flow time. In each case we use a split of 10% of the data for validation and 90% of the data for training to check for overfitting and proceed to train the network with full batch training (batches of 1,800 data points) until we reach a sufficiently small loss value (10,000 epochs in total), **Figures 7A,B**. As **Figures 7C,D** shows, the different time steps behave similarly with minimal deviation from the analytical values, indicating a low sensitivity to the time step for this range of values. The gray points in **Figure 7C** are forward Euler approximations of the time derivative of the order parameter that are computed from the flow data generated with $\Delta t = 0.01$. These points reveal the scatter in the data, as noted in **Figure 6A**, and illustrate the power of the neural network time-stepping process to regress the data and average out the noise to find what can be thought of as an expected value (over oscillator ensemble realizations) of the time derivative. To provide a comparison between finite difference time-derivative estimates and our neural network based approach, we compute a geometric harmonic interpolation of the Euler approximations with 10 geometric harmonics and include the result in **Figure 7C**.

As an added point of comparison between our neural network right-hand-sides and the analytical expression, we integrate flows for a variety of initial conditions with both of the neural network approximated right-hand-sides and the analytical (theoretical, infinite oscillator limit) right-hand-side. As **Figure 8** illustrates, the neural network equations produce flows that both closely match the analytical solution over the majority of the R-domain and converge to the correct steady state. Thus, the neural network

FIGURE 5 | The neural network approach for learning unknown ODE right-hand-sides templated on a fourth order, fixed-step Runge–Kutta numerical integration algorithm. The neural "sub"-network f is reused for each stage of the integration algorithm in order to compute the loss function and train the network.

FIGURE 6 | (A) A plot of our initialization trajectories (in black) vs. the expected continuum limit solution (in red). The end point of each black trajectory, denoted by an orange star, corresponds to a value of $R(t)$ that is used as a starting point for the flow training data. The noise evident in the simulated trajectories is caused by both randomness in the frequencies (each trajectory is a different sample from the distribution), and randomness in the arrangement of the oscillators with respect to their frequencies (only for the $R \approx 0$ starting condition). **(B)** The approximately uniform sampling of the initial R-values $R(t)$ used for flow generation. Note that a uniform distribution is not required, however it is desirable to facilitate training. It is only necessary that the values cover the range of R-space in order to allow us to learn the ODE over the entire [0, 1] domain.
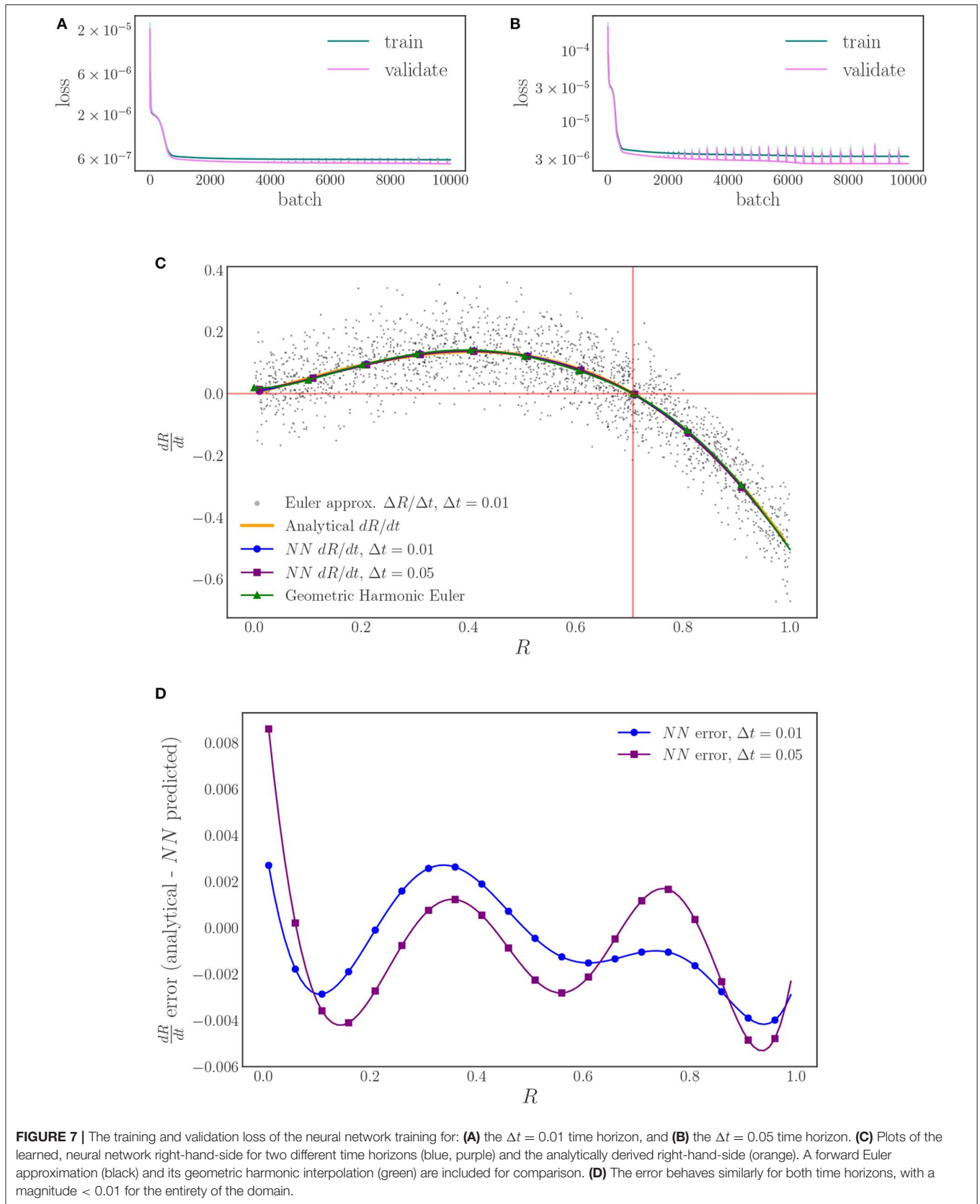
approach successfully learns an accurate approximation of both the behavior and the governing ODEs of the Kuramoto order parameter in the continuum limit.

One of the greatest utilities of this approach is its generality, which enables it to be applied to cases in which analytical approaches have not yet been devised, or may not be practical. Throughout the previous example we assumed knowledge of the coarse variable, the Kuramoto order parameter, and learned its governing ODEs. However, it is important to realize that this same technique can also be applied to our "discovered" coarse variables, that we identified earlier in this paper through manifold learning techniques. In this way, this methodology allows one to **both** identify the descriptive coarse variables **and** learn their behaviors (ODEs) in a general, data-driven way that is tailored to the specific problem by the nature of the technique. We conclude this section with a demonstration of such an application.

Following the same methodology outlined in the "Order Parameter Identification" section, we combine the phase data corresponding to the two $(R(t), R(t + \Delta t))$ data sets used earlier in this section and then bin the oscillator phases for each time point to produce oscillator density data points. As before, we use this oscillator density data as the input to the DMAPs algorithm with a Gaussian kernel with the Euclidean distance and find that it is described by a single significant eigenfunction, as determined by the local linear regression method. **Figure 9** illustrates the nearly one-to-one mapping between this discovered diffusion map coordinate and the analytical order parameter R. Following our discovery of a coarse variable, we then use the neural network process outlined earlier in this section to approximate the evolution law (the right-hand-side of the unknown ODE) for our discovered, diffusion map coarse variable ϕ_1.

By keeping track of the ϕ_1 values corresponding to the R values for each data set, we form two sets of pairs of training data as

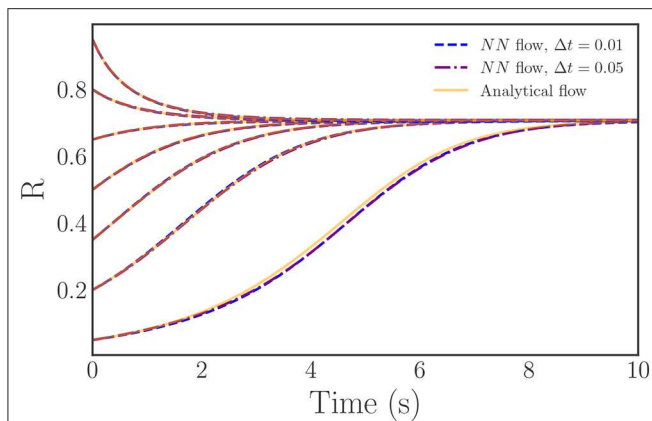FIGURE 7 | The training and validation loss of the neural network training for: **(A)** the $\Delta t = 0.01$ time horizon, and **(B)** the $\Delta t = 0.05$ time horizon. **(C)** Plots of the learned, neural network right-hand-side for two different time horizons (blue, purple) and the analytically derived right-hand-side (orange). A forward Euler approximation (black) and its geometric harmonic interpolation (green) are included for comparison. **(D)** The error behaves similarly for both time horizons, with a magnitude < 0.01 for the entirety of the domain.

FIGURE 8 | Plots of trajectories generated from the learned, neural network, evolution law right-hand-side (blue, purple) vs. the analytically known evolution law (orange). The learned trajectories exhibit nearly identical behavior to the analytical trajectories and, importantly, converge to the correct steady state.

FIGURE 9 | The coarse variable identified by our manifold learning procedure applied to data generated in this section. Only the single coarse coordinate ϕ_1 was deemed significant by the local linear regression method. Even in the presence of noisy data, there is nearly a one-to-one map between the discovered coordinate ϕ_1 and the analytical order parameter R.

before, $(\phi_1(t), \phi_1(t + \Delta t))$, with one for each value of Δt, but now in the manifold learning derived variable ϕ_1. Using an identical procedure to the one used to learn the right-hand-side of the R equation, we define a feedforward neural network with 3 hidden layers of 24 neurons each and use it to approximate the time derivative of ϕ_1 in an architecture templated on a fixed step-size, fourth order Runge–Kutta algorithm. As before, we initialize this neural network integration procedure with a uniform Glorot procedure for the kernels, and zeros for the biases. We define the loss to be the mean squared error on the final point of the training flow and then train the network with full batch training with the Adam optimizer with a learning rate of 10^{-3}. **Figure 10** illustrates the result of this training procedure for each time step. As before, we include the forward Euler approximation of the ϕ_1 time derivative for $\Delta t = 0.01$ and its geometric harmonic interpolation with 10 geometric harmonics as a point of reference.

Our recurrent, integrator-based neural network architecture, appears to successfully learn the evolution equation for ϕ_1 over the domain. However, the fit does not appear to be as accurate as the one found for R itself. Furthermore, this training required 50,000 epochs to reach this level of accuracy compared to the 10,000 used for R. We argue that both the difficulty in training and the lower quality of this fit can be attributed to the bias in sampling introduced by the non-uniform sampling in our diffusion map coordinate, demonstrated by the accumulation of the black data points in **Figure 10**. Earlier we noted that we took great care to produce training data sets that sample the R domain in a nearly uniform fashion, **Figure 6B**. As demonstrated by **Figure 9**, the mapping from R to ϕ_1 is one-to-one, but does not preserve the shape of the sampling density due to its non-linearity, leading to a bias in the sampling of the ϕ_1 domain. However, despite this bias, the neural network procedure manages to learn a time derivative near the visual average of its observed Euler approximation.

To summarize, we have shown how our order parameter identification method can discover a coarse variable that
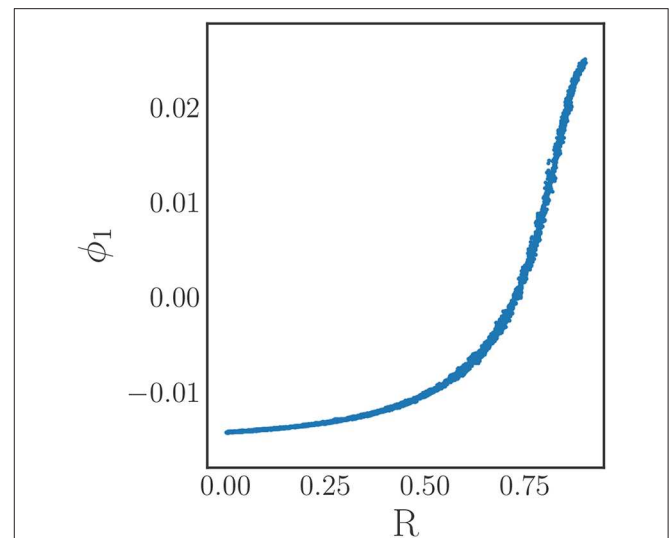
shows good agreement (i.e., is one-to-one) with a known analytical variable, even in the case of non-ideal, noisy data. Furthermore, we have demonstrated that even with biased sampling, our recurrent, integrator templated neural network architecture is capable of successfully learning the right-hand-side of the evolution of our discovered coarse variable, effectively smoothing its Euler finite difference estimates, and certainly closely approximating the correct steady state.

6. IDENTIFYING "EFFECTIVE" PARAMETERS

Up to this point we focused on coarse-graining (reducing the dimension) the system state variables required to formulate an effective dynamic coupled oscillator model. Often, however, complex systems whose dynamics depend explicitly on several parameters can also admit *a reduced set of parameter combinations* (or "effective" parameters) that depend on the full set of parameters in complex, non-linear ways. Finding these reduced sets of parameters often requires insight, along with trial and error. Here, we present a methodology for discovering such *effective parameters* in a data-driven way via a modification of diffusion maps, our manifold learning technique of choice in this paper (Holiday et al., 2019). A strong motivation for this work comes from the determination of explicit dimensionless parameters from the (possibly long) list of dimensional parameters of a physical model.

Throughout this section we investigate variations of the Kuramoto model and show how DMAPs can be used to discover effective parameters that accurately describe the phases of Kuramoto oscillators when synchronized (at steady state in
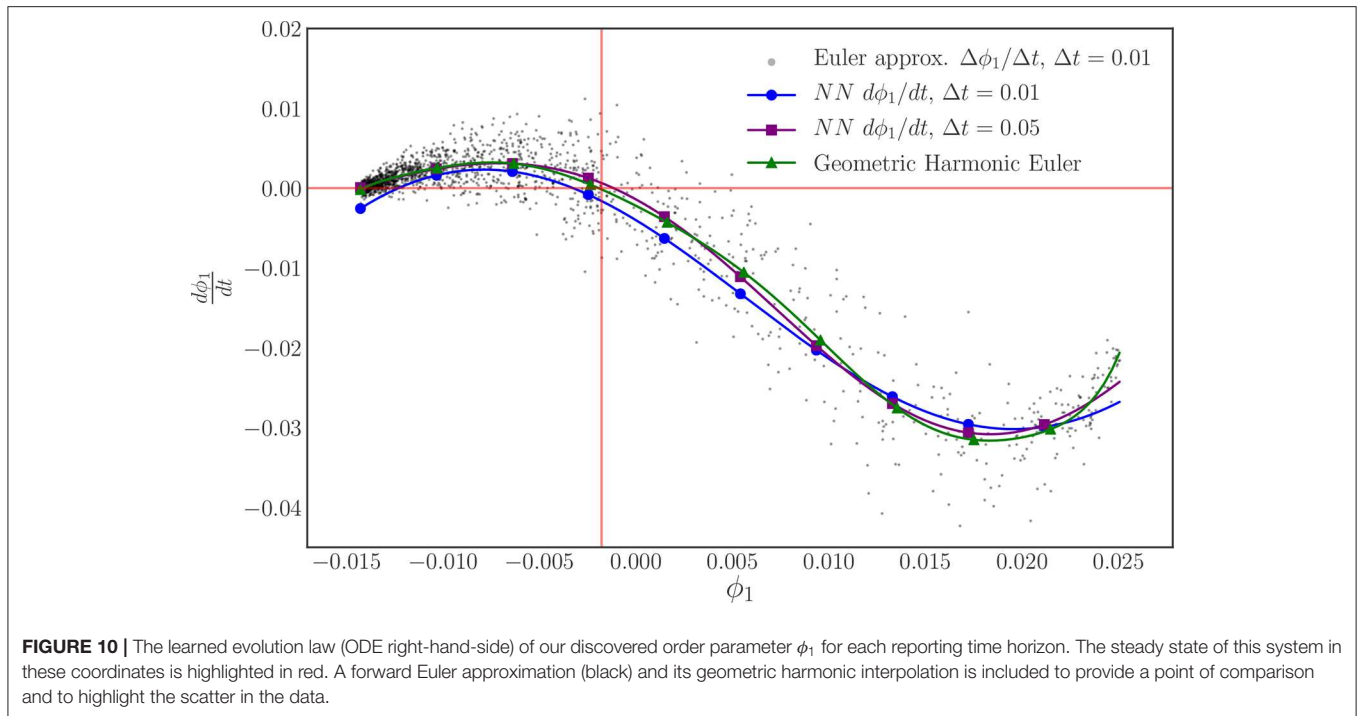
FIGURE 10 | The learned evolution law (ODE right-hand-side) of our discovered order parameter ϕ_1 for each reporting time horizon. The steady state of this system in these coordinates is highlighted in red. A forward Euler approximation (black) and its geometric harmonic interpolation is included to provide a point of comparison and to highlight the scatter in the data.

a rotating frame). The general idea is to use an *output-only informed kernel* for the diffusion map. With this approach, we only consider the outputs of the system, here steady state phase data, as the input to the DMAPs algorithm/kernel computation and neglect the values of the oscillator parameters, which embody the heterogeneity of the oscillator ensemble. This allows us to discover the intrinsic dimensionality of the state space at synchronization, independent of the detailed list of system parameters. The significant eigenfunctions provided by DMAPs, as determined by the local linear regression method, serve as new coordinates for the state space, that is, their values completely determine the values of the state variables at steady state (16).

$$\theta_\infty(\mathbf{p}) \quad \text{Original analysis} \quad (15)$$

$$\theta_\infty(\boldsymbol{\phi}) \quad \text{Diffusion map coordinate(s)} \quad (16)$$

$$\mathbf{p} \mapsto \boldsymbol{\phi} \quad \text{New "effective" parameter(s)} \quad (17)$$

Since these effective coordinates describe the variability of the steady state variables (which depend on the detailed system parameters), the eigenfunctions themselves provide a new, data-driven set of effective parameters for the model. If fewer eigenfunctions are required to describe the steady state space than there are original parameters, then these eigenfunctions furnish an *effective reduced set of parameters*. Once such data-driven effective parameters are found, they can be compared to the original parameters through regression to determine how they are related (17). We illustrate this methodology through a series of examples.

6.1. A Simple Example: The Kuramoto Model With Heterogeneous Coupling Coefficients

We begin by considering a simple variation of the Kuramoto model in which we include the coupling constant K_i as an oscillator heterogeneity in addition to the frequencies ω_i. The governing equations for this model are

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K_i}{N} \sum_{j=1}^{N} \sin(\theta_j - \theta_i), \quad \text{for } i = 1, \ldots, N. \quad (18)$$

Similar to the typical Kuramoto model, the phase synchronization of the oscillators can be described with the usual complex-valued order parameter

$$Re^{i\psi} = \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j}. \quad (19)$$

As with the typical Kuramoto model, this variation admits a steady state in a rotating frame if there is complete synchronization.

Our goal for this model is to show that even though there are two model heterogeneities, (ω_i, K_i), the steady state phases can be described by a single "effective" parameter. That is, we will demonstrate that there is a new parameter that depends on both of the original two parameters, such that the steady state oscillator phases only depend on this single combination of the original parameters.

In order to find this effective parameter, we apply the DMAPs algorithm to the steady state phase data with an *output-only informed kernel*. This means that our observations consist solely of the output of the model, here the steady state phases of the oscillators $\theta_{i,\infty} = \theta_i(t_\infty)$ (where t_∞ is a time after which the oscillators have reached a steady state), and ignore the oscillator heterogeneities (ω_i, K_i). We use the eigenfunctions provided by the DMAPs algorithm to define a change of variables for the parameter space (ω, K). As we show, only a single eigenfunction ϕ_1 is required to describe the phase data. Thus, this change of variables is a many-to-one map and provides a reduction from the two original parameters (ω, K) to the single effective parameter, ϕ_1.

$$(\omega, K) \mapsto \theta_\infty \in \mathbb{R} \quad \text{Original analysis} \tag{20}$$

$$\phi_1(\omega, K) \mapsto \theta_\infty \in \mathbb{R} \quad \text{Diffusion map coordinate} \tag{21}$$

For our simulations we consider 1,500 oscillators ($N = 1,500$) with uniformly randomly distributed initial phases over $[0, 2\pi]$, uniformly randomly distributed frequencies over $[-\pi, \pi]$, and uniformly randomly distributed coupling coefficients over $[10, 100]$. We select these parameter values as they lead to complete synchronization, and hence a steady state in a rotating frame. We integrate the oscillators with Scipy's vode integrator until they achieve complete synchronization, and then transform the phases into a rotating frame.

Next, we apply the DMAPs algorithm to the steady state phases with a Gaussian kernel with the Euclidean distance and parameters of $\alpha = 1$ for the Laplace-Beltrami operator and $\epsilon = 0.5$ for the kernel bandwidth parameter. The kernel is given by

$$K_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\epsilon^2}\right) \quad i, j = 1, \ldots, 1500, \tag{22}$$

where $\|\cdot\|_2$ is the Euclidean norm in the complex plane[1] of 1500 oscillator phases, e.g., $x_j = e^{i\theta_j(t_\infty)}$.

We use the local linear regression method to verify that only a single diffusion map eigenfunction ϕ_1 is required to represent this phase data, resulting in a single, data-driven effective parameter. A coloring of the original, two-dimensional parameter space (ω, K) by this eigenfunction is shown in **Figure 11A**, demonstrating the many-to-one character of the diffusion map coordinate map.

Due to the simplicity of this model it is also possible to find an effective parameter analytically. Multiplying the order parameter

---

[1]It is necessary to map the oscillator phases to the complex plane to avoid the unfortunate occurrence of the phases lying across the branch cut of the multi-valued argument function. Taking the Euclidean norm of the difference between oscillator phases in this situation would result in the DMAPs algorithm incorrectly identifying two different clusters of oscillators split across the branch cut instead of a single group. By first mapping the phases to the complex plane and then computing the distances there, we avoid this possibility and correctly identify a single group of oscillators.

by $e^{-i\theta_i}$, taking the imaginary part, and substituting the result into the model equations yields

$$\frac{d\theta_i}{dt} = \omega_i + (RK_i)\sin(\psi - \theta_i), \tag{23}$$

which under steady state conditions yields an effective parameter

$$\theta_{i,\infty} = \psi + \arcsin\left(\frac{\omega_i}{RK_i}\right). \tag{24}$$

As a validation of our data-driven approach, we compare our data-driven parameter ϕ_1 to the analytical one (24). In **Figure 11** we show the parameter space colored by (a) our data-driven parameter ϕ_1, (b) the analytical parameter, and (c) the steady state phases. As the figure illustrates, the colorings are similar suggesting that our data-driven parameter is indeed an equivalent effective parameter for this model.

This is confirmed by the plot in **Figure 11D**, which clearly illustrates the invertible relationship between the data-driven parameter and the analytically obtainable parameter combination. Thus, our data-driven approach is able to discover an equivalent effective parameter for this model. Combinations of the original parameters that yield the same steady state synchronized phase can be found as level sets of the eigenfunction ϕ_1 in (ω, K) space.

6.2. A Three-Parameter Example: The Kuramoto Model With Firing

We now consider a modification to the Kuramoto model in which we additionally incorporate a "firing term" with coefficient α_i to the coupling strength K_i, and frequency ω_i of each oscillator. This model was originally introduced to model excitable behavior among coupled oscillators. If $|\omega_i/\alpha_i| < 1$ and $K_i = 0$, each oscillator exhibits two steady states, one stable and one unstable. A small perturbation of the stationary, stable solution that exceeds the unstable steady state induces a firing of the oscillator, which appears as a large deviation in phase before a return to the stable state (Tessone et al., 2007, 2008). The model equations for this variation are provided below.
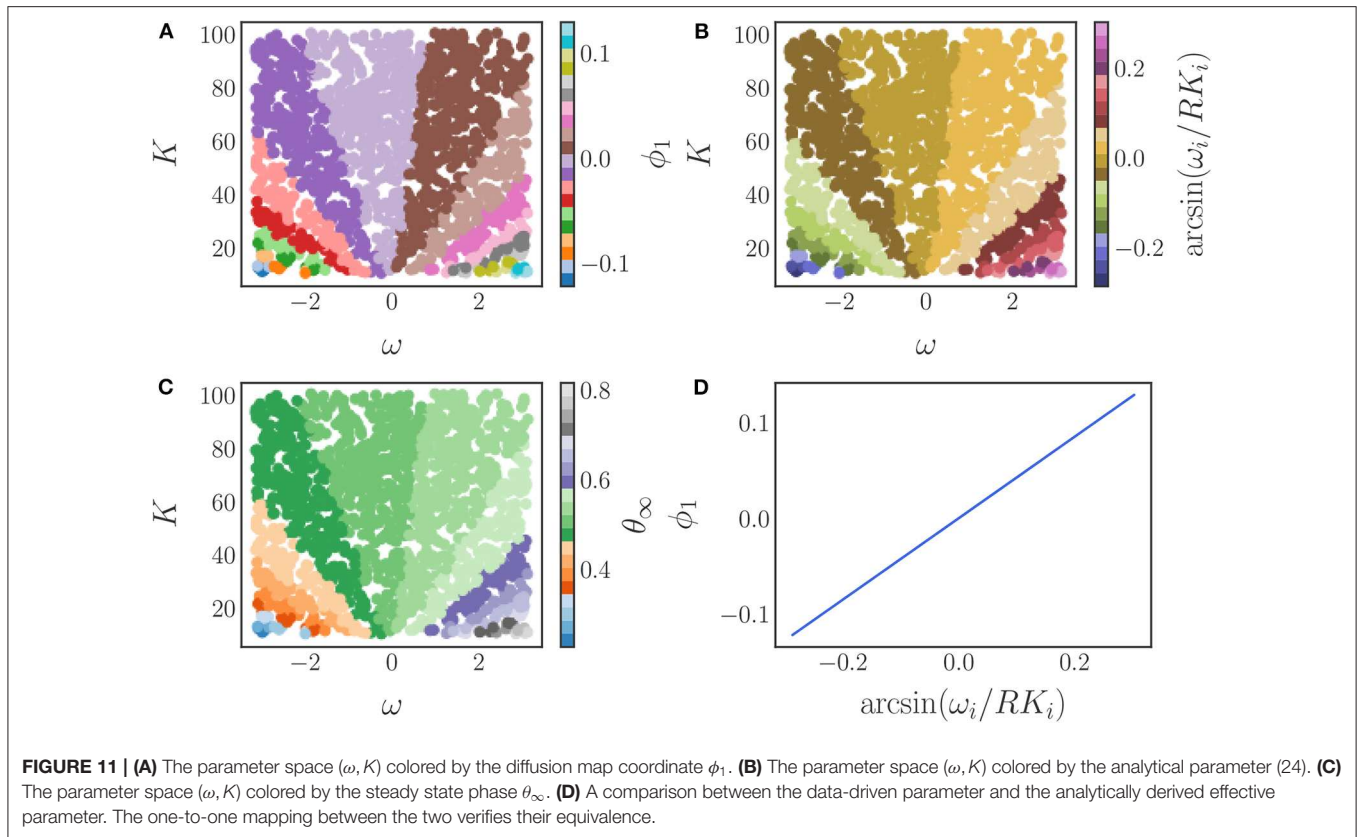
$$\frac{d\theta_i}{dt} = \omega_i + \alpha_i \sin(\theta_i) + \frac{K_i}{N}\sum_{j=1}^{N} \sin(\theta_j - \theta_i) \tag{25}$$

Similar to the typical Kuramoto model, one can express the degree of phase synchronization among the oscillators with the Kuramoto order parameter,

$$Re^{i\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j}. \tag{26}$$

Transforming the model equations in the same way as those of the Kuramoto model with heterogeneous coupling coefficients (18) studied earlier results in

FIGURE 11 | **(A)** The parameter space (ω, K) colored by the diffusion map coordinate ϕ_1. **(B)** The parameter space (ω, K) colored by the analytical parameter (24). **(C)** The parameter space (ω, K) colored by the steady state phase θ_∞. **(D)** A comparison between the data-driven parameter and the analytically derived effective parameter. The one-to-one mapping between the two verifies their equivalence.

$$\frac{d\theta_i}{dt} = \omega_i + \alpha_i \sin(\theta_i) + (RK_i)\sin(\psi - \theta_i), \qquad (27)$$

which under steady state conditions yields

$$\frac{2RK_i}{\alpha_i}\sin(\psi)\sin^2\left(\frac{\theta_i}{2}\right) + \left(\frac{RK_i\cos(\psi)}{\alpha_i} - 1\right)\sin(\theta_i)$$
$$= \frac{RK_i}{\alpha_i}\sin(\psi) + \frac{\omega_i}{\alpha_i}. \qquad (28)$$

Now considering a rotating reference frame, in which ψ is constant, we set $\psi = 0$ for convenience yielding

$$\left(\frac{RK_i}{\alpha_i} - 1\right)\sin(\theta_i) = \frac{\omega_i}{\alpha_i}. \qquad (29)$$

By the above manipulations, it is now clear that the steady state phases $\theta_{i,\infty}$ are a function of K_i/α_i and ω_i/α_i, meaning that this system can be analytically described by two combinations of the original parameters. We now employ our data-driven approach to discover the effective parameter(s).

We begin by simulating this model using Scipy's vode integrator with $N = 1,500$ oscillators, α_i uniformly randomly sampled in $[-2, 2]$, K_i uniformly randomly sampled in $[10, 100]$, and ω_i uniformly randomly sampled in $[-\pi, \pi]$. We select these parameter values to ensure complete synchronization of the

oscillators and hence a steady state in a rotating frame. As with the Kuramoto model with heterogeneous coupling coefficients, we transform the phases into a rotating frame and apply the DMAPs algorithm with an output-only informed kernel to the complex transformed steady state phases. We select diffusion map parameters of $\alpha = 1$, and $\epsilon = 0.9$ and find that a single diffusion map coordinate ϕ_1 is identified by the local linear regression method, giving rise to a single effective parameter.

This parameter is a non-linear combination of all three original parameters (ω, K, α), as we illustrate in **Figure 12**, which shows the three-dimensional parameter space colored by the level sets of the single significant diffusion map coordinate. Thus, this model admits a reduction of the three original parameters to a single effective parameter ϕ_1, which is itself a combination of the three original system parameters (ω, K, α). The relationship between the system parameters and ϕ_1 can be subsequently explored, if desired, through standard regression techniques.

6.3. A More Complicated Example: The Kuramoto Model With Chung-Lu Coupling

Here, we showcase our data-driven parameter discovery process for systems with more complicated couplings. Instead of the all-to-all coupled model considered by Kuramoto (10), we consider a general Kuramoto model (9) with Chung-Lu type coupling between oscillators (Chung and Lu, 2002). The connection probability between oscillators for a Chung-Lu network is given by
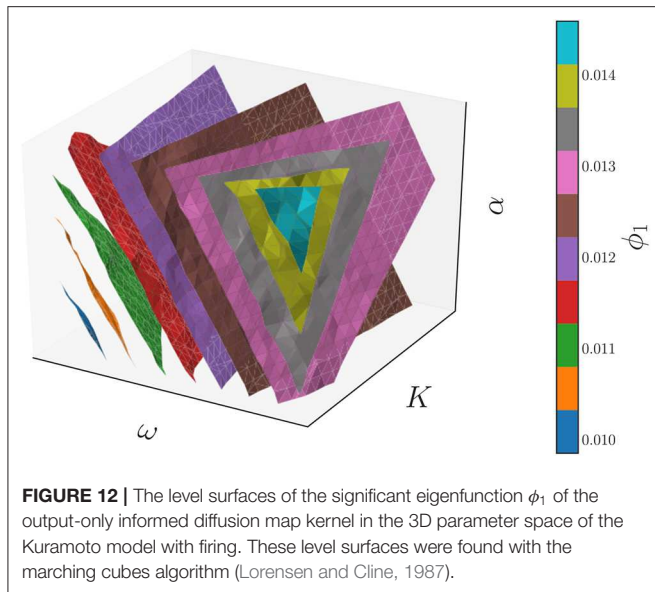
FIGURE 12 | The level surfaces of the significant eigenfunction ϕ_1 of the output-only informed diffusion map kernel in the 3D parameter space of the Kuramoto model with firing. These level surfaces were found with the marching cubes algorithm (Lorensen and Cline, 1987).

$$P_{ij} = P_{ji} = \min\left(\frac{w_i w_j}{\sum_k w_k}, 1\right), \qquad (30)$$

where w_i is a sequence of weights defined by

$$w_i = N_p \left(1 - q(i-1)/N\right)^r \quad i = 1, 2, \ldots, N, \qquad (31)$$

for network parameters p, q, and r. Multiple Chung-Lu networks can be generated from the same parameter values, with a specific network corresponding to generating an adjacency matrix from the connection probabilities P_{ij} defined by the network parameters through the sequence of weights w_i.

One of the special properties of the Kuramoto system with a Chung-Lu coupling is that when a steady state exists in a rotating reference frame, the steady state phases of the oscillators θ_i lie on an invariant manifold (Bertalan et al., 2017), which can be parameterized by two heterogeneities: the frequency of the oscillators ω_i, and a network property called the degree κ_i, as depicted in **Figure 13A**. This property was observed to hold for many Chung-Lu networks generated from a given set of parameters. We now show how the effect of both of these heterogeneities can be succinctly expressed by a single "effective" parameter.

Throughout the remainder of this section we consider a collection of 4,000 oscillators ($N = 4,000$) with uniformly randomly distributed frequencies ω_i over the range $[0, 1]$ that are coupled together in a Chung-Lu network with a coupling constant of $K = 20$ (multiplying the adjacency matrix), and network parameters of $p = 0.5$, $q = 0.9$, and $r = 0.5$. **Figure 13B** shows the degree distribution of a sampling of a Chung-Lu network with these parameter values. We are careful to select our coupling constant large enough to ensure complete synchronization for these parameter values, and hence a steady state in a rotating frame that can be described in terms of the heterogeneities ω, and κ.

After our parameter selection, we integrate the oscillators in time with SciPy's Runge–Kutta integrator (*solve_ivp* with RK45) until they reach a steady state in a rotating reference frame.

Now we show that, although there are two heterogeneity parameters, (ω_i, κ_i), the long term behavior of the Kuramoto oscillators with a Chung-Lu network is intrinsically one dimensional, and can be described by a single effective parameter, which itself is a, possibly non-linear, combination of the original system parameters.

$$
\begin{aligned}
(\omega, \kappa) &\mapsto \theta_\infty \quad \text{Original analysis} & (32)\\
\phi &\mapsto \theta_\infty \quad \text{Diffusion map outcome} & (33)
\end{aligned}
$$

In order to find this effective parameter, we begin by applying the DMAPs algorithm to the complex transformed steady state phase data with an output-only informed kernel. As mentioned before, this means that our observations consist solely of the steady state phases of the oscillators $\theta_{i,\infty}$, and ignore the oscillator heterogeneities (ω_i, κ_i). For our diffusion maps we use $\alpha = 1$ for the Laplace-Beltrami operator, a kernel bandwidth parameter of $\epsilon \approx 2.3 * 10^{-2}$, and the Gaussian kernel with the Euclidean distance
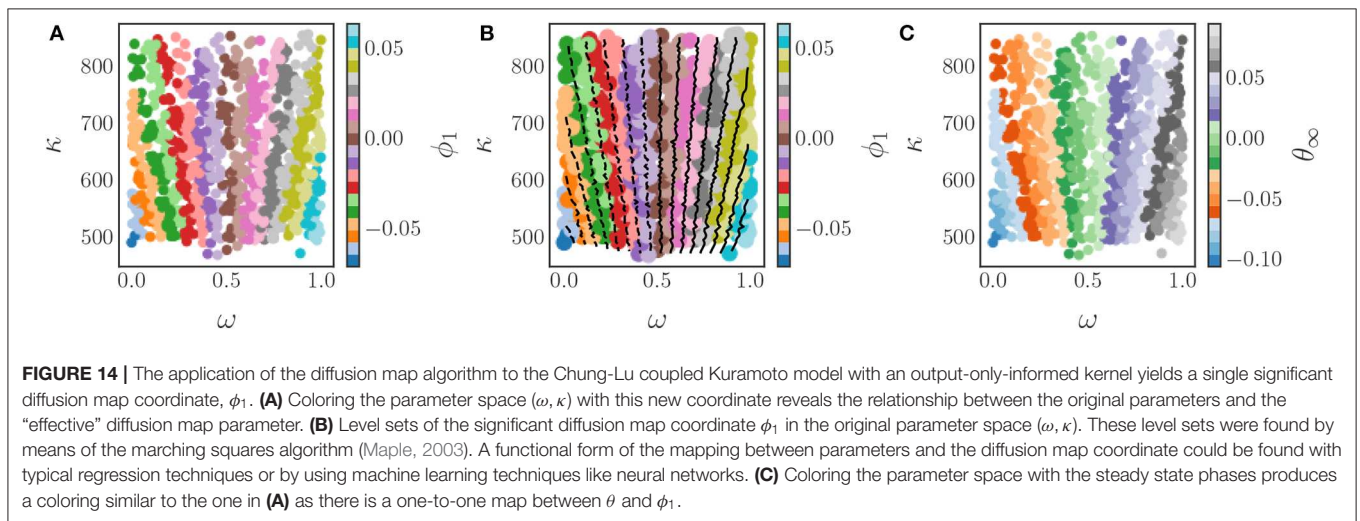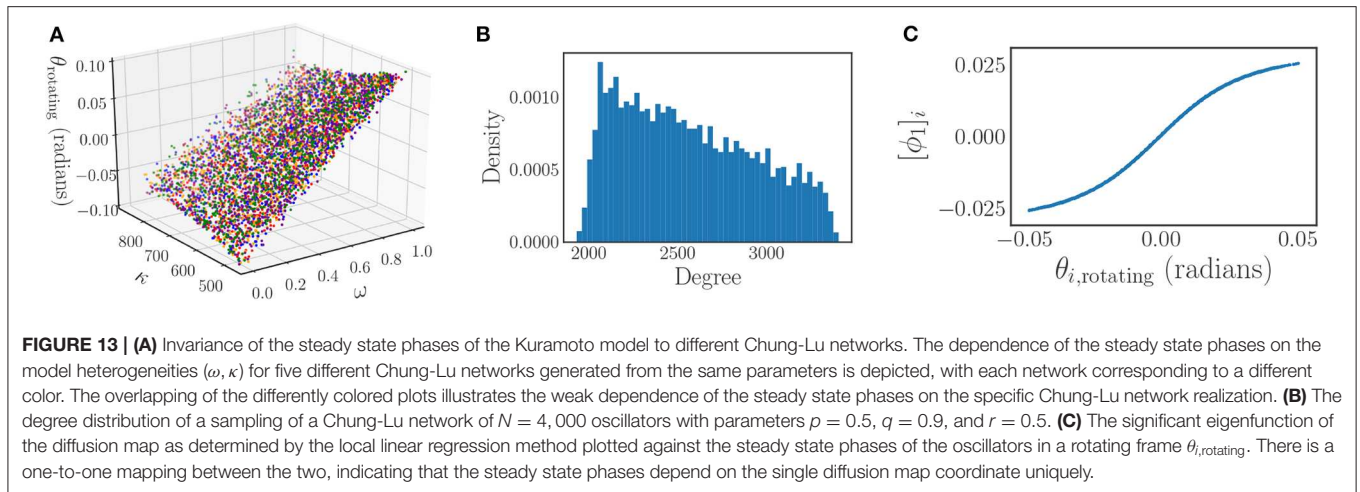
$$K_{ij} = \exp\left(-\frac{||x_i - x_j||_2^2}{\epsilon^2}\right) \quad i, j = 1, \ldots, 4000, \qquad (34)$$

where $|| \cdot ||_2$ is the Euclidean norm in the complex plane of 4,000 oscillator phases, i.e., $x_j = e^{i\theta_j(t_\infty)}$.

Next, we use the local linear regression method to determine that there is a single significant eigenfunction, ϕ_1. **Figure 13C** shows that this eigenfuction is one-to-one with the steady state phases of the oscillators in a rotating frame, and thus provides an equivalent description of the steady state behavior of this system. Therefore, the synchronized phases of this system can be accurately described by the single effective parameter ϕ_1, as claimed.

The relationship between the original parameters and the DMAPs parameter is illustrated by **Figure 14A**, which depicts a coloring of the original parameter space with ϕ_1. In this figure one can observe that there are multiple combinations of the original parameters that correspond to the same value of ϕ_1 and hence the same steady state phase. The key observation is that the level sets of ϕ_1 provide the mapping between the original parameters and the new effective parameter. These level sets are depicted in **Figure 14B**, and can be found with established techniques, such as the marching squares algorithm (Maple, 2003). Thus, by using the DMAPs parameter ϕ_1 we can express the steady state phases in terms of a single combination of the original parameters. If necessary/useful, we can try to express this new effective parameter as a function of the original system parameters through standard regression techniques, or even possibly through neural networks (**Figure 14C**).

To summarize our approach, in each of three examples above we used DMAPs combined with the local linear regression method to determine the intrinsic dimensionality of the output

FIGURE 13 | (A) Invariance of the steady state phases of the Kuramoto model to different Chung-Lu networks. The dependence of the steady state phases on the model heterogeneities (ω, κ) for five different Chung-Lu networks generated from the same parameters is depicted, with each network corresponding to a different color. The overlapping of the differently colored plots illustrates the weak dependence of the steady state phases on the specific Chung-Lu network realization. **(B)** The degree distribution of a sampling of a Chung-Lu network of $N = 4,000$ oscillators with parameters $p = 0.5$, $q = 0.9$, and $r = 0.5$. **(C)** The significant eigenfunction of the diffusion map as determined by the local linear regression method plotted against the steady state phases of the oscillators in a rotating frame $\theta_{i,\text{rotating}}$. There is a one-to-one mapping between the two, indicating that the steady state phases depend on the single diffusion map coordinate uniquely.

FIGURE 14 | The application of the diffusion map algorithm to the Chung-Lu coupled Kuramoto model with an output-only-informed kernel yields a single significant diffusion map coordinate, ϕ_1. **(A)** Coloring the parameter space (ω, κ) with this new coordinate reveals the relationship between the original parameters and the "effective" diffusion map parameter. **(B)** Level sets of the significant diffusion map coordinate ϕ_1 in the original parameter space (ω, κ). These level sets were found by means of the marching squares algorithm (Maple, 2003). A functional form of the mapping between parameters and the diffusion map coordinate could be found with typical regression techniques or by using machine learning techniques like neural networks. **(C)** Coloring the parameter space with the steady state phases produces a coloring similar to the one in **(A)** as there is a one-to-one map between θ and ϕ_1.

space. The significant eigenfunctions that we obtained from this process provided new coordinates for the output space and can be considered as the "effective" parameters of the system. If there are fewer significant DMAPs eigenfunctions than original parameters, then this change of variables also provides a reduction in total necessary parameters.

7. DISCUSSION AND FUTURE WORK

Throughout this paper we have presented a data-driven methodology for discovering coarse variables, learning their dynamic evolution laws, and identifying sets of effective parameters. In each case, we used either an example or a series of examples to demonstrate the efficacy of our techniques compared to the established analytical technique and, in each case, the results of our data-driven approach were in close agreement with the established methodology.

Nevertheless, it is important to consider the interpretability (the "X" in XAI, explainable artificial intelligence) of the data-driven coarse variables discovered in our work. Even when performing model reduction with linear data-driven techniques, like Principal Component Analysis, it is difficult to ascribe a

physical meaning to linear combinations of meaningful system variables (what does a linear combination of, say, a firing rate and an ion concentration "mean"?). The conundrum is resolved by looking for physically meaningful quantities that, on the data, are one-to-one with the discovered data-driven descriptors, and are therefore equally good at parametrizing the observations. One hypothesizes a set of meaningful descriptors and then checks that the Jacobian of the transformation from data-driven to meaningful descriptors never becomes singular *on the data* (see Sonday et al., 2009; Frewen et al., 2011; Kattis et al., 2016; Rajendran et al., 2017; Meila et al., 2018 for further discussion).

With that said, we believe that our techniques offer an approach that is both general and systematic, and we intend to apply it to a variety of coupled oscillator systems. One such problem that we are currently investigating is the possible existence, and data-driven identification, of partial differential equation (PDE) descriptions of coupled oscillator systems. Such an alternative coarse-grained description would confer the typical benefits associated with model reduction, such as accelerated simulation and analysis; it will also present unique challenges: for instance, the selection of appropriate

boundary conditions for such a data-driven PDE model of coupled oscillators.

As we remarked in our discussion of the integrator-based neural network architecture, there is an extension of the ODE neural network approach that allows one to learn PDEs discretized by the method of lines approach (Gonzalez-Garcia et al., 1998). We plan to leverage this capability to discover PDE descriptions of coupled oscillator systems, such as the simple Kuramoto model in the continuum limit as well as of networks of Hodgkin–Huxley oscillators.

Utilizing the properties of the continuous form of the Kuramoto model, one can express the time- and phase-dependent oscillator density $F(\theta, t)$ as an integral of the conditional oscillator density with respect to the frequency.

$$F(\theta, t) = \int_{-\infty}^{\infty} \rho(\theta | \omega; t) g(\omega) d\omega \qquad (35)$$

As we pointed out earlier, Ott and Antonsen discovered an invariant attracting manifold for the simple Kuramoto model in the continuum limit with Cauchy distributed frequencies (Ott and Antonsen, 2008). It has been shown that on this manifold, the oscillator density F satisfies the following equation.

$$F(\theta, t) = \frac{1 - R^2}{2\pi (1 - 2R\cos(\psi - \theta) + R^2)} \qquad (36)$$

Away from this attracting invariant manifold, the full oscillator density $\rho(\theta, \omega, t)$ obeys the continuity equation

$$\frac{\partial \rho(\theta, \omega, t)}{\partial t} + \frac{\partial ([\omega + \frac{K}{2i}(re^{-i\theta} - r^*e^{i\theta})] \rho(\theta, \omega, t))}{\partial \theta} = 0 \qquad (37)$$

We would like to use our neural network based approach to learn equivalent PDEs directly from oscillator density data. As an example of what this would look like for $F(\theta, t)$, one can approximate the partial derivative of F with respect to time

$$\frac{\partial F(\theta, t)}{\partial t} \approx \frac{F(\theta, t + \delta t) - F(\theta, t)}{\delta t}. \qquad (38)$$

Furthermore, one can numerically obtain the partial derivative of F with respect to θ. Plotting $\frac{\partial F}{\partial t}$ vs. both F and F_θ produces a loop, as illustrated in **Figure 15**, where traversing a loop corresponds to varying θ from 0 to 2π. Each of the different loops depicted in **Figure 15** coincides with a different initial value of $R(t)$, ranging from 0 to 0.85. Thus, it appears that along the attracting manifold, we can write

$$\frac{\partial F}{\partial t} = G\left(F, \frac{\partial F}{\partial \theta}\right) \qquad (39)$$

where G is an unknown function of F and $\frac{\partial F}{\partial \theta}$. The unknown function G takes a form that is amenable to being learned with the PDE extension of the neural network integration procedure. In future work we intend to demonstrate this process.
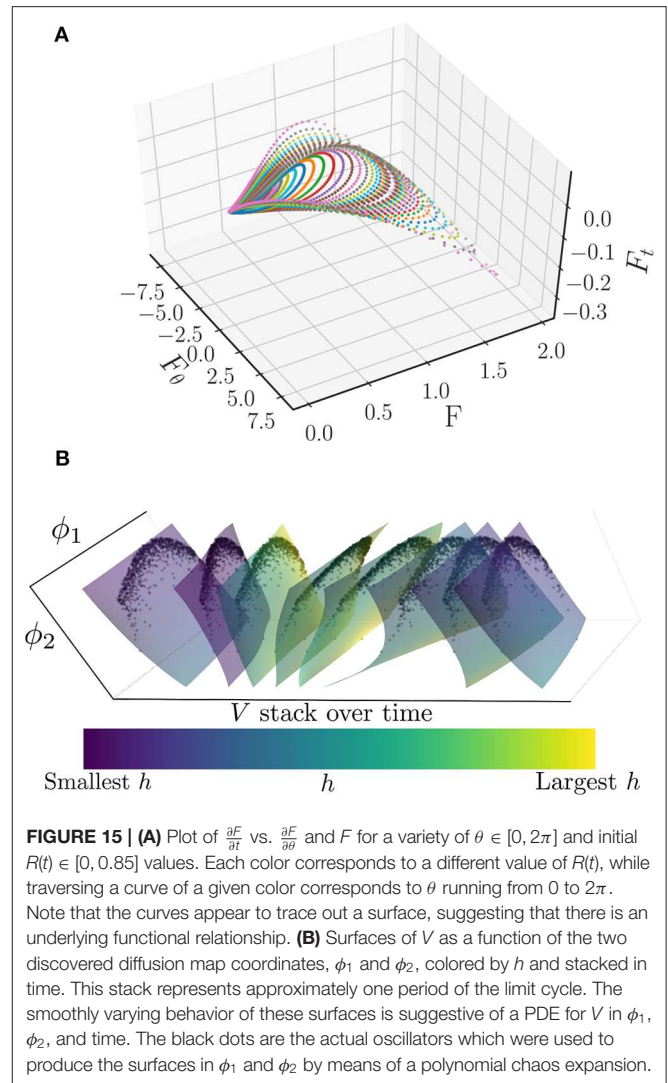
FIGURE 15 | (A) Plot of $\frac{\partial F}{\partial t}$ vs. $\frac{\partial F}{\partial \theta}$ and F for a variety of $\theta \in [0, 2\pi]$ and initial $R(t) \in [0, 0.85]$ values. Each color corresponds to a different value of $R(t)$, while traversing a curve of a given color corresponds to θ running from 0 to 2π. Note that the curves appear to trace out a surface, suggesting that there is an underlying functional relationship. **(B)** Surfaces of V as a function of the two discovered diffusion map coordinates, ϕ_1 and ϕ_2, colored by h and stacked in time. This stack represents approximately one period of the limit cycle. The smoothly varying behavior of these surfaces is suggestive of a PDE for V in ϕ_1, ϕ_2, and time. The black dots are the actual oscillators which were used to produce the surfaces in ϕ_1 and ϕ_2 by means of a polynomial chaos expansion.

Moving away from simple phase oscillators, we consider a model of Hodgkin–Huxley neural oscillators studied in (Choi et al., 2016), which are characterized by two state variables each, a channel state h and a potential V (40),

$$C\frac{dV_i}{dt} = -g_{Na}m(V_i)h_i(V_i - V_{Na}) - g_l(V_i - V_l) + I_{syn}^i + I_{app}^i,$$
$$\frac{dh_i}{dt} = \frac{h_\infty(V_i) - h_i}{\tau(V_i)}, \qquad (40)$$

for $i = 1, \dots, N$. Where the coupling is provided by the synaptic current I_{syn} defined as

$$I_{syn}^i = \frac{g_{syn}(V_{syn} - V_i)}{N} \sum_{j=1}^{N} A_{ij}s(V_j), \qquad (41)$$

with adjacency matrix A_{ij}. The functions $\tau(V)$, $h_\infty(V)$, and $m(V)$ are a standard part of the Hodgkin–Huxley formalism, $s(V)$ is

the synaptic communication function, and g_{Na}, V_{Na}, g_l, and V_l are model parameters.

It has been shown that with a Chung-Lu network (30, 31) these oscillators are drawn to an attractive limit cycle along which their states can be described by two parameters: their applied current I_{app} and their degree, κ. These two heterogeneities can also be described by two diffusion map coordinates, ϕ_1 and ϕ_2 (Kemeth et al., 2018). Plotting the potential, V, for a single period of the limit cycle produces the stack of surfaces shown in **Figure 15B**. The smoothly varying character of these surfaces is suggestive of a PDE description of these oscillators along this limit cycle. We intend to investigate the identification of such a PDE through an extension of the data-driven identification technique for coarse ODEs presented herein.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

REFERENCES

Bertalan, T., Wu, Y., Laing, C., Gear, C. W., and Kevrekidis, I. G. (2017). Coarse-grained descriptions of dynamics for networks with both intrinsic and structural heterogeneities. *Front. Comput. Neurosci.* 11:43. doi: 10.3389/fncom.2017.00043

Breakspear, M. (2002). Nonlinear phase desynchronization in human electroencephalographic data. *Hum. Brain Mapp.* 15, 175–198. doi: 10.1002/hbm.10011

Breakspear, M., Heitmann, S., and Daffertshofer, A. (2010). Generative models of cortical oscillations: neurobiological implications of the kuramoto model. *Front. Hum. Neurosci.* 4:190. doi: 10.3389/fnhum.2010.00190

Brown, E., Moehlis, J., and Holmes, P. (2004). On the phase reduction and response dynamics of neural oscillator populations. *Neural Comput.* 16, 673–715. doi: 10.1162/089976604322860668

Cabral, J., Hugues, E., Sporns, O., and Deco, G. (2011). Role of local network oscillations in resting-state functional connectivity. *Neuroimage* 57, 130–139. doi: 10.1016/j.neuroimage.2011.04.010

Cardaliaguet, P., and Euvrard, G. (1992). Approximation of a function and its derivative with a neural network. *Neural Netw.* 5, 207–220. doi: 10.1016/S0893-6080(05)80020-6

Choi, M., Bertalan, T., Laing, C. R., and Kevrekidis, I. G. (2016). Dimension reduction in heterogeneous neural networks: generalized polynomial chaos (GPC) and analysis-of-variance (ANOVA). *Eur. Phys. J. Spcl Top.* 225, 1165–1180. doi: 10.1140/epjst/e2016-02662-3

Chow, T. W., and Li, X.-D. (2000). Modeling of continuous time dynamical systems with input by recurrent neural networks. *IEEE Trans. Circ. Systems I Fund. Theory Appl.* 47, 575–578. doi: 10.1109/81.841860

Chung, F., and Lu, L. (2002). Connected components in random graphs with given expected degree sequences. *Ann. Combinat.* 6, 125–145. doi: 10.1007/PL00012580

Coifman, R. R., and Lafon, S. (2006a). Diffusion maps. *Appl. Comput. Harm. Anal.* 21, 5–30. doi: 10.1016/j.acha.2006.04.006

Coifman, R. R., and Lafon, S. (2006b). Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comput. Harm. Anal.* 21, 31–52. doi: 10.1016/j.acha.2005.07.005

Deco, G., Jirsa, V., McIntosh, A. R., Sporns, O., and Kötter, R. (2009). Key role of coupling, delay, and noise in resting brain fluctuations. *Proc. Natl. Acad. Sci. U.S.A.* 106, 10302–10307. doi: 10.1073/pnas.0901831106

Dsilva, C. J., Talmon, R., Coifman, R. R., and Kevrekidis, I. G. (2018). Parsimonious representation of nonlinear dynamical systems through manifold learning:

a chemotaxis case study. *Appl. Comput. Harm. Anal.* 44, 759–773. doi: 10.1016/j.acha.2015.06.008

Du, X., Ghosh, B. K., and Ulinski, P. (2005). Encoding and decoding target locations with waves in the turtle visual cortex. *IEEE Trans. Biomed. Eng.* 52, 566–577. doi: 10.1109/TBME.2004.841262

Ermentrout, B. (1996). Type I membranes, phase resetting curves, and synchrony. *Neural Comput.* 8, 979–1001. doi: 10.1162/neco.1996.8.5.979

Ermentrout, G., and Kopell, N. (1990). Oscillator death in systems of coupled neural oscillators. *SIAM J. Appl. Math.* 50, 125–146. doi: 10.1137/0150009

Ermentrout, G. B., and Kopell, N. (1986). Parabolic bursting in an excitable system coupled with a slow oscillation. *SIAM J. Appl. Math.* 46, 233–253. doi: 10.1137/0146017

Freemann, W. (1975). *Mass Action in the Nervous System-Examination of the Neurophysiological Basis of Adaptive Behavior through the EEG.* New York, NY: Academic Press.

Frewenm T. A., Couzin, I. D., Kolpas, A., Moehlis, J., Coifman, R., Kevrekidis, I. G. (2011). "Coarse collective dynamics of animal groups," in *Coping with Complexity: Model Reduction and Data Analysis. Lecture Notes in Computational Science and Engineering*, Vol. 75, eds A. Gorban and D. Roose (Berlin; Heidelberg: Springer).

Glorot, X., and Bengio, Y. (2010). "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, (Sardinia), 249–256.

Gonzalez-Garcia, R., Rico-Martinez, R., and Kevrekidis, I. (1998). Identification of distributed parameter systems: a neural net based approach. *Comput. Chem. Eng.* 22, S965?S968. doi: 10.1016/S0098-1354(98)00191-4

Graves, A., Mohamed, A.-R., and Hinton, G. (2013). "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vancouver, BC: IEEE), 6645–6649. doi: 10.1109/ICASSP.2013.6638947

Guckenheimer, J., and Holmes, P. (2013). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Vol. 42.* New York, NY: Springer Science & Business Media.

Guresen, E., Kayakutlu, G., and Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.* 38, 10389–10397. doi: 10.1016/j.eswa.2011.02.068

Haile, J. M. (1997). *Molecular Dynamics Simulation: Elementary Methods, 1st Edn.* New York, NY: John Wiley & Sons, Inc.

Holiday, A., Kooshkbaghi, M., Bello-Rivas, J. M., Gear, C. W., Zagaris, A., and Kevrekidis, I. G. (2019). Manifold learning for parameter reduction. *J. Comput. Phys.* 392, 419–431. doi: 10.1016/j.jcp.2019.04.015

Honey, C. J., Kötter, R., Breakspear, M., and Sporns, O. (2007). Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proc. Natl. Acad. Sci. U.S.A.* 104, 10240–10245. doi: 10.1073/pnas.0701519104

Izhikevich, E., and Hoppensteadt, F. (1997). *Weakly Connected Neural Networks*. New York, NY: Springer. doi: 10.1007/978-1-4612-1828-9

Jeong, S.-O., Ko, T.-W., and Moon, H.-T. (2002). Time-delayed spatial patterns in a two-dimensional array of coupled oscillators. *Phys. Rev. Lett.* 89:154104. doi: 10.1103/PhysRevLett.89.154104

Kattis, A. A., Holiday, A., Stoica, A.-A., and Kevrekidis, I. G. (2016). Modeling epidemics on adaptively evolving networks: A data-mining perspective. *Virulence* 7, 153–162. doi: 10.1080/21505594.2015.1121357

Kemeth, F. P., Haugland, S. W., Dietrich, F., Bertalan, T., Höhlein, K., Li, Q., et al. (2018). An emergent space for distributed data with hidden internal order through manifold learning. *IEEE Access* 6, 77402–77413. doi: 10.1109/ACCESS.2018.2882777

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv [preprint]* arXiv:1412.6980.

Kosmatopoulos, E. B., Polycarpou, M. M., Christodoulou, M. A., and Ioannou, P. A. (1995). High-order neural network structures for identification of dynamical systems. *IEEE Trans. Neural Netw.* 6, 422–431. doi: 10.1109/72.363477

Kuramoto, Y. (1975). "Self-entrainment of a population of coupled non linear oscillators," in *International Symposium on Mathematical Problems in Theoretical Physics* (Berlin; Heidelberg: Springer), 420–422. doi: 10.1007/BFb0013365

Kuramoto, Y. (1984). *Chemical Oscillations, Waves, and Turbulence*. Berlin; Heidelberg: Springer-Verlag. doi: 10.1007/978-3-642-69689-3

Lam, Y.-W., Cohen, L. B., Wachowiak, M., and Zochowski, M. R. (2000). Odors elicit three different oscillations in the turtle olfactory bulb. *J. Neurosci.* 20, 749–762. doi: 10.1523/JNEUROSCI.20-02-00749.2000

Lorensen, W. E., and Cline, H. E. (1987). Marching cubes: a high resolution 3d surface construction algorithm. *ACM Siggraph Comp. Graph.* 21, 163–169. doi: 10.1145/37402.37422

Mankin, J., and Hudson, J. (1986). The dynamics of coupled nonisothermal continuous stirred tank reactors. *Chem. Eng. Sci.* 41, 2651–2661. doi: 10.1016/0009-2509(86)80053-7

Maple, C. (2003). "Geometric design and space planning using the marching squares and marching cube algorithms," in *2003 International Conference on Geometric Modeling and Graphics, 2003* (London, UK: IEEE), 90–95. doi: 10.1109/GMAG.2003.1219671

Meila, M., Koelle, S., and Zhang, H. (2018). A regression approach for explaining manifold embedding coordinates. *arXiv [preprint]* arXiv:1811.11891.

Narendra, K. S., and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.* 1, 4–27. doi: 10.1109/72.80202

O'Keeffe, K. P., Hong, H., and Strogatz, S. H. (2017). Oscillators that sync and swarm. *Nat. Commun.* 8:1504. doi: 10.1038/s41467-017-01190-3

Ott, E., and Antonsen, T. M. (2008). Low dimensional behavior of large systems of globally coupled oscillators. *Chaos* 18:037113. doi: 10.1063/1.2930766

Penny, W., Litvak, V., Fuentemilla, L., Duzel, E., and Friston, K. (2009). Dynamic causal models for phase coupling. *J. Neurosci. Methods* 183, 19–30. doi: 10.1016/j.jneumeth.2009.06.029

Prechtl, J., Cohen, L., Pesaran, B., Mitra, P., and Kleinfeld, D. (1997). Visual stimuli induce waves of electrical activity in turtle cortex. *Proc. Natl. Acad. Sci. U.S.A.* 94, 7621–7626. doi: 10.1073/pnas.94.14.7621

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2018). Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv [preprint]* arXiv:1801.01236.

Rajendran, K., Kattis, A., Holiday, A., Kondor, R., and Kevrekidis, Y. (2017). *Data Mining When Each Data Point is a Network*. Springer. doi: 10.1007/978-3-319-64173-7_17

Rico-Martınez, R., Kevrekidis, I., and Krischer, K. (1995). "Nonlinear system identification using neural networks: dynamics and instabilities," in *Neural Networks for Chemical Engineers*, ed A. B. Bulsari (Amsterdam: Elsevier Science BV), 409–442.

Rico-Martinez, R., and Kevrekidis, I. G. (1993). "Continuous time modeling of nonlinear systems: A neural network-based approach," in *IEEE International Conference on Neural Networks* (San Francisco, CA: IEEE), 1522–1525. doi: 10.1109/ICNN.1993.298782

Rico-Martinez, R., Krischer, K., Kevrekidis, I., Kube, M., and Hudson, J. (1992). Discrete-vs. continuous-time nonlinear signal processing

of cu electrodissolution data. *Chem. Eng. Commun.* 118, 25–48. doi: 10.1080/00986449208936084

Rodrigues, F. A., Peron, T. K. D., Ji, P., and Kurths, J. (2016). The kuramoto model in complex networks. *Physics Rep.* 610, 1–98. doi: 10.1016/j.physrep.2015.10.008

Rubino, D., Robbins, K. A., and Hatsopoulos, N. G. (2006). Propagating waves mediate information transfer in the motor cortex. *Nat. Neurosci.* 9, 1549–1557. doi: 10.1038/nn1802

Schmidt, H., Petkov, G., Richardson, M. P., and Terry, J. R. (2014). Dynamics on networks: the role of local dynamics and global networks on the emergence of hypersynchronous neural activity. *PLoS Comput. Biol.* 10:e1003947. doi: 10.1371/journal.pcbi.1003947

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature* 529:484. doi: 10.1038/nature16961

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv [preprint]* arXiv:1409.1556.

Sonday, B. E., Haataja, M., and Kevrekidis, I. G. (2009). Coarse-graining the dynamics of a driven interface in the presence of mobile impurities: effective description via diffusion maps. *Phys. Rev. E* 80:031102. doi: 10.1103/PhysRevE.80.031102

Stam, C. J., Nolte, G., and Daffertshofer, A. (2007). Phase lag index: assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources. *Hum. Brain Mapp.* 28, 1178–1193. doi: 10.1002/hbm.20346

Strogatz, S. H. (2000). From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Phys. D Nonlinear Phenomena* 143, 1–20. doi: 10.1016/S0167-2789(00)00094-4

Strogatz, S. H., Abrams, D. M., McRobie, A., Eckhardt, B., and Ott, E. (2005). Theoretical mechanics: crowd synchrony on the millennium bridge. *Nature* 438:43. doi: 10.1038/438043a

Subudhi, B., and Jena, D. (2011). A differential evolution based neural network approach to nonlinear system identification. *Appl. Soft Comput.* 11, 861–871. doi: 10.1016/j.asoc.2010.01.006

Tass, P., Rosenblum, M. G., Weule, J., Kurths, J., Pikovsky, A., et al. (1998). Detection of n: m phase locking from noisy data: application to magnetoencephalography. *Phys. Rev. Lett.* 81:3291. doi: 10.1103/PhysRevLett.81.3291

Tass, P. A. (2007). *Phase Resetting in Medicine and Biology: Stochastic Modelling and Data Analysis*. Berlin; Heidelberg: Springer-Verlag.

Tessone, C. J., Scire, A., Toral, R., and Colet, P. (2007). Theory of collective firing induced by noise or diversity in excitable media. *Phys. Rev. E* 75:016203. doi: 10.1103/PhysRevE.75.016203

Tessone, C. J., Zanette, D. H., and Toral, R. (2008). Global firing induced by network disorder in ensembles of active rotators. *Eur. Phys. J. B* 62, 319–326. doi: 10.1140/epjb/e2008-00162-5

Varela, F., Lachaux, J.-P., Rodriguez, E., and Martinerie, J. (2001). The brainweb: phase synchronization and large-scale integration. *Nat. Rev. Neurosci.* 2, 229–239. doi: 10.1038/35067550

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., et al. (2017). Starcraft II: a new challenge for reinforcement learning. *arXiv [preprint]* arXiv:1708.04782.

Wilson, H. R., and Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13, 55–80. doi: 10.1007/BF00288786