Learning Dynamics Across Similar Spatiotemporally-Evolving Physical Systems

Joshua Whitman, Girish Chowdhary
Coordinated Science Laboratory (CSL)
University of Illinois at Urbana Champaign
`jewhitm2,girishc@illinois.edu`

Abstract: We present a differentially-constrained machine learning model that can generalize over similar spatiotemporally evolving dynamical systems. It is shown that not only can an E-GP model be used to estimate the latent state of large-scale physical systems of this type, but that a single E-GP model can generalize over multiple physically-similar systems over a range of parameters using only a few training sets. This is demonstrated on computational flow dynamics (CFD) data sets on fluid flowing past a cylinder at different Reynolds numbers. Though these systems are governed by highly nonlinear partial differential equations (the Navier-Stokes equations), we show that their major dynamical modes can be captured by a linear dynamical layer over the temporal evolution of the weights of stationary kernels. Furthermore, the model generated by this method provides easy access to physical insights into the system, unlike comparable methods like Recurrent Neural Networks (RNN). The low computational cost of this method suggests that it has the potential to enable machine learning approximations of complex physical phenomena for autonomy and robotic design tasks.

Keywords: Dynamics, Modeling, Learning, CFD

1 Introduction

One of the fundamental problems in bringing machine learning to physical domains is in modeling large-scale stochastic phenomena with both spatial and temporal (spatiotemporal) evolution [1]. Examples of such phenomena include temperature variation, CO_2 flux over large areas, extreme weather events [2] like wildfires, pedestrian traffic patterns, and fluid dynamics. The last example is a classic physics problem for numerical analysis, and is an ongoing subject of research in the field of Computational Fluid Dynamics (CFD). This field's aim is to model fluid flow using the first principles of fluid mechanics, e.g., by using numerical methods to solve the nonlinear Navier-Stokes partial differential equations. These simulations are costly and resource-intensive, sometimes requiring days on a supercomputer to generate. This means they are ill-suited for machine-learning tasks that require access to dozens or hundreds of simulations of different but similar situations. They are even more poorly suited for online robotic applications, such as autonomous aerial, ground, or water vehicles.

In contrast to first-principles approaches, data-driven models of spatiotemporally evolving phenomena have been gaining more attention in the machine learning and statistics communities [3]. The ultimate goal of this approach would be to generate highly efficient machine learning models that can be used instead of the costly numerical simulations for design and autonomy purposes. Success of this technique could revolutionize design and control of complex physical systems, such as soft robotics, as they would significantly reduce the cost and resources required in simulations. However, in order to be successful, these models need to be able to generalize across different physical situations. For example, in the context of fluid flows, these models must be able to predict fluid dynamics at different conditions (e.g. Reynolds number) than the training data. This is a difficult problem, as it requires that the model have the capability to actually learn the underlying physics and not just input-output relationships. Currently no machine learning technique has demonstrated the capability to generalize on these problems; it is goal of this paper to address this gap.

1.1 Contribution

We report highly exciting results that for the first time demonstrate that a machine learning model can learn *and* generalize over the physics of the Navier-Stokes partial differential equation (PDEs). We expect that the model can generalize over other similarly parameterized PDEs.

We leverage the very recent results published in [4], using the predictive mechanism therein to perform inference with a single machine learning model over multiple systems. We term this model Evolving Gaussian Processes (E-GPs), which is a differentially-constrained hierarchic modeling method that layers a linear dynamic transition model on the weights of a kernel-based model (such as a Gaussian Process or a Gaussian Radial Basis Function Neural Network). The advantage of E-GPs is that by separating the spatial and temporal dynamics hierarchically and using a linear transition model on the weights, the learning problem becomes more tractable while complex spatiotemporal behaviors can still be captured in a relatively low-complexity model. Furthermore, the linear transition model not only provides physical insights into the system, but also enables the design of observers and controllers [4, 5]. However, the focus in this paper is in demonstrating the generalizability of the E-GP modeling approach on learning complex PDEs. This approach also leaves open the possibility of modeling nonlinear behavior in the weight-space evolution using neural networks as the transition models, especially as the flow becomes turbulent at higher Reynolds numbers.

We demonstrate our results using CFD data of flow over a bluff body over a range of Reynolds numbers from 100 to 1000. The conventional wisdom would be to learn a separate model over each Reynolds number, but our results show that this is not necessary if one leverages our spatially encoded hierarchic Evolving Gaussian Process model. Using the learned dynamics over weights of successive kernel models, E-GP is capable of predicting the future states of functional evolution in a recurrent manner. The key benefit is that evolution of large function spaces can be transformed into learning the evolution of a relatively smaller Hilbert space to which is encoded by the kernels and the associated weight vector. Furthermore, the values of the weights and the associated linear dynamical systems provide critical insights such as spatial-correlations through the structure of the transition matrix, local modes of dynamic evolution through invariant subspaces of the transition matrix, and eigenmodes of evolution. The latter is of significant importance to the ongoing work in Koopman operator models of spatiotemporal phenomena.

1.2 Related Work

In the machine learning community, kernel methods constitute a very well-studied and powerful class of methods for inference in spatial domains [6], in which correlations between input variables are encoded via a covariance kernel, and the model is formed through a weighted sum of the kernels [7]. There is a significant body of literature on extending these methods to spatiotemporal modeling [8, 7]. A naive approach is to utilize both spatial and temporal variables as inputs to the kernel. However, this technique leads to an ever-growing kernel dictionary, which is computationally taxing. In recent years, some degree of success has been found [3] by focusing on designing nonseperable and nonstationary covariance kernels for environment-specific dynamics and optimizing/learning associated hyperparameters in local regions of the input space (ours being a significant exception). The Process Convolution with Local Smoothing Kernels (PCLSK) approach [9] captures nonstationary structures by allowing variation in kernel hyperparameters across the input space, which can be modeled using additional latent Gaussian processes [10, 11, 12]. Other such methods map the nonstationary process into a latent space where the problem becomes approximately stationary [13, 14]. However, there are a few major drawbacks to this approach. First of all, these methods currently have limited scalability to large-scale phenomena, due to the fact that the hyperparameter optimization problem is not convex in general, leading to methods that are difficult to implement (like MCMC), susceptible to local minima, and can become computationally intractable for large datasets like those generated by CFD. The scalability issue is only exacerbated by the fact that data is typically retained across both space and time. Secondly, the models generated by these methods do not lend themselves well to addressing the important challenges of monitoring systems with sensor feedback and designing controllers.

The geostatistics community literature has many examples of the dynamical spatiotemporal modeling approach, where the focus is on finding good dynamical transition models on the linear combination of weights in a parameterized model [15]. The advantage of this approach is that when the spatial and temporal dynamics are hierarchically separated, the learning problem can be made convex if linear transition models are used; as a result complex nonstationary kernels are often not necessary. The approach presented in this paper aligns closely with this vein of work. The main difference is that we view the problem from the more abstract viewpoint of constructing an observer in a reproducing kernel Hilbert space. One major payoff of this perspective, as shown in this paper, is that it enables us to determine the optimal number and location of sensors for efficiently monitoring and predicting the state of the distributed system. In particular, if feedback is allowed, monitoring (state recovery) and prediction (filtering) can be made more efficient than other nonstationary kernel methods [**?**].

Within the CFD community literature, a new framework for data-driven analysis of nonlinear fluid flow was introduced in the 90s, called Koopman operator theory of dynamical systems. A Koopman operator is a linear but infinite-dimensional operator that is defined for an autonomous dynamical system and governs the evolution of its *observables* [16]. If the most significant eigenfunctions, eigenvalues, and eigenmodes of the operator can be approximated from the data, many of the same advantages of E-GP are realized: the ability to transform the state space so the dynamics appear linear, to predict the temporal evolution of the linear system, to reconstruct the state of the original nonlinear system, and even to implement controller design. The best known mode approximation method is known as Proper Orthogonal Decomposition (POD) [17] or Principal Component Analysis (PCA). Dynamic Mode Decomposition (DMD) is the most widely used method for finding a finite-dimensional subspace of the Koopman operator's infinite-dimensional domain to work in. Williams et al., recently integrated DMD with the kernel trick, allowing the algorithm to be extended to systems with much larger dimensions [18]. However, this method is restricted to approximating the Koopman operator and is only indirectly concerned with generative models, whereas our method is concerned with the evolution of the weights which can be directly used to compute observables (we also use Gaussian kernels instead of polynomial). Most importantly, in this paper it is demonstrated that our method can generalize across similar systems.

To our knowledge, no neural network has been able to model large-scale CFD systems that evolve in both space and time. Recently, convolutional neural networks were used to model steady state (not evolving in time) flow profiles at low Reynolds numbers and at low resolution [19], but the CNN paradigm does not generalize well to modeling dynamic flows. For the purpose of aerospace design tasks, feedforward neural networks have been used to model a highly restricted subset of the CFD output, such as the pressure at a couple of points on an object surface [20]. We are not aware of any other application of RNNs to CFD data.

2 Background

2.1 Partial Differential Equations and the Navier-Stokes Equations

Partial differential equations are ubiquitous in science and engineering and have their origins in multivariate calculus with functions that operate in continuous space. In this space, any change in functional values can be represented as a combination of partial derivatives of this multivariable function with respect to the independent variables (usually time and space). Examples of such PDEs litter the various areas of science and engineering as a means to describe evolutionary dynamics of many complex systems, including areas of mechanics (solids, fluids, gases), transport phenomena in general (waves, information), electrostatics & electromagnetics, circuits, thermal sciences, quantum mechanics, transmission lines and more.

The predominant class of PDEs encountered in practical science and engineering are of the second order. These second order PDEs, especially when linear, can be classified into elliptic, parabolic or hyperbolic depending on the signature of eigenvalues of the coefficients of the PDE system. However, the generic PDEs encountered in practice are non-linear, i.e. the coefficients of the PDE system are functions of the independent variables or dependent variables or both. For example, the Navier-Stokes equations (NSE) can be classified as mixed type, i.e they can behave as hyperbolic or parabolic or elliptic systems in different regimes of the non-linear coefficients, depending on the boundary and initial conditions specified. The other consequence of the nonlinearity is chaotic dynamics, commonly referred to as turbulence, wherein any small disturbance evolve the system along bifurcations to excite unstable modes and new physical scales, in a cumulative cascading effect.

In our view, the Navier-Stokes equations represents most, if not all of the overall complexity of modeling 2nd order PDEs as it (a) allows of hybrid system behavior, i.e. elliptic-hyperbolic etc. and (b) the nonlinearity results in complex spatio-temporal dynamics that is prevalent in many practical situations. The form of the NSE for compressible Newtonian fluids is expressed below, where $\mathbf{u}$ is the fluid velocity, p is the fluid pressure, ρ is the fluid density, and μ is the fluid dynamic viscosity. [21]

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\right) + \nabla\left(-\frac{2\mu}{3}\nabla\cdot\mathbf{u}\right) + \rho\mathbf{g}$$

2.2 Kernel Observers

As presented in [4], the problem is predictive inference of a time-varying stochastic process, whose mean f evolves as $f_{\tau+1} \sim \mathbb{F}(f_\tau, \eta_\tau)$, where $\mathbb{F}$ is a distribution varying with time τ and exogenous inputs η. The goal of our approach is to hierarchically separate temporal evolution from spatial functional evolution. Our prototype is the classical and quite general *abstract evolution equation* (AEO), which can be defined as the evolution of a function u embedded in a Banach space $\mathcal{B}$: $\dot{u}(t) = \mathcal{L}u(t)$, subject to $u(0) = u_0$, and $\mathcal{L} : \mathcal{B} \to \mathcal{B}$ determines spatiotemporal transitions of $u \in \mathcal{B}$ [22]. To make this approach computationally realizable, we restrict the sequence f_τ to lie in a reproducing kernel Hilbert space (RKHS) [7]. Let $k : \Omega \times \Omega \to \mathbb{R}$ be a positive-definite Mercer kernel on a domain Ω, modeling the covariance betwen any two points in the input space. This also implies the existence of a smooth map $\psi : \Omega \to \mathcal{H}$, where $\mathcal{H}$ is an RKHS with the property $k(x, y) = \langle\psi(x), \psi(y)\rangle_{\mathcal{H}}$. The insight of the proposed model is in assuming spatiotemporal evolution in the input domain corresponds to temporal evolution of the mixing weights of a kernel model alone in the functional domain.

Let $y \in \mathbb{R}^N$ be the measurements of the function available from N sensors, $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ be a linear transition operator in the RKHS $\mathcal{H}$, and $\mathcal{K} : \mathcal{H} \to \mathbb{R}^N$ be a linear measurement operator. The model for the functional evolution and measurement studied in this paper is:

$$f_{\tau+1} = \mathcal{A}f_\tau + \eta_\tau, \quad y_\tau = \mathcal{K}f_\tau + \zeta_\tau, \tag{1}$$

where η_τ is a zero-mean stochastic process in $\mathcal{H}$, and ζ_τ is a Wiener process in $\mathbb{R}^N$. To avoid working in dual space and have the parameters grow with the data, we work with an approximate feature map $\widehat{\psi}(x) := [\,\widehat{\psi}_1(x) \cdots \widehat{\psi}_M(x)\,]$ to an approximate feature space $\widehat{\mathcal{H}}$. Typical examples of such maps include random Fourier features [23], FastFood [24], A la Carte [25], and the Nyström method [26]. Here we use the dictionary of atoms approach as follows: let Ω be compact. Given points $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$, define the dictionary of atoms $\mathcal{F}^{\mathcal{C}} = \{\psi(c_1), \cdots, \psi(c_M)\}$, $\psi(c_i) \in \mathcal{H}$, the span of which is a strict subspace $\widehat{\mathcal{H}}$ of the RKHS $\mathcal{H}$ generated by the kernel, where $\widehat{\psi}_i(x) := k(x, c_i)$. In the approximate space case, we replace the transition operator $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ in (1) by $\widehat{\mathcal{A}} : \widehat{\mathcal{H}} \to \widehat{\mathcal{H}}$. The finite-dimensional evolution equations approximating (1) in approximate dual form are

$$w_{\tau+1} = \widehat{A}w_\tau + \eta_\tau, \quad y_\tau = Kw_\tau + \zeta_\tau, \tag{2}$$

where we have matrices $\widehat{A} \in \mathbb{R}^{M\times M}$, $K \in \mathbb{R}^{N\times M}$, the vectors $w_\tau \in \mathbb{R}^M$, and where we have slightly abused notation to let y_τ, η_τ and ζ_τ denote their $\widehat{\mathcal{H}}$ counterparts. Here K is the matrix whose rows are of the form $K_{(i)} = [\,\widehat{\psi}_1(x_i)\ \widehat{\psi}_2(x_i) \cdots \widehat{\psi}_M(x_i)\,]$. In systems-theoretic language, the matrix acts as a measurement operator.

Modeling the system as a linear time-invariant dynamic system in the weight space enables the use of several important and useful techniques from control theory. For example, it was demonstrated in [5] that given a spatiotemporally evolving system modeled using (2), under certain conditions one may choose a set of N sensing locations such that even with $N \ll M$, the functional evolution of the spatiotemporal model can be estimated (which corresponds to *monitoring*) and can be predicted robustly (which corresponds to *Bayesian filtering*). The key to solving this problem is designing the measurement operator K so that the pair $(K, \widehat{A})$ is observable. By taking the Jordan decomposition of the $\widehat{A}$ and looking at the geometric multiplicities of the eigenvalues, one can determine the *cyclic index* of $\widehat{A}$. The cyclic index is a nonconservative lower bound on the number of distinct sampling locations required for the observability of system (2), and is equal to the number of *invariant subspaces* $\mathcal{H}_i \subset \mathcal{H}$ into which $\mathcal{A}$ can be uniquely decomposed.

3 Evolving Gaussian Processes

The Evolving Gaussian Processes method builds on the Kernel Observers method. The primary novelty in our method of generating a model is learning an $\widehat{A}$ matrix for *multiple* systems. We found that the class of functional evolutions $\mathbb{F}$ defined by linear Markovian transitions in a RKHS is still sufficient to model the nonlinear Navier Stokes equations, since the unknown map ψ allows us to model highly nonlinear dynamics in the input space. However, we do expect that phenomena such as bifurcation or turbulence will require nonlinear mappings $\mathcal{H}$. There are three steps to generate an Evolving Gaussian Process model:

1. After picking the kernel and estimating the bandwidth hyperparameter σ (we utilize the maximum likelihood approach, although other approaches can be used), find an optimal basis vector set $\mathcal{C}$ using the algorithm in [30].
2. Use Gaussian process inference to find weight vectors for each time-step in the training set(s), generating the sequence $w_\tau, \tau = 1, \ldots, T$ for each system. A uniform time-step makes next step easier but can be worked around for non-uniform data sets
3. Using the weight trajectory, use matrix least-squares with the equation $\widehat{A}[w_1, w_2, ..., w_{T-1}] = [w_2, w_3, ..., w_T]$ to solve for $\widehat{A}$.
4. To generate a multi-system model, concatenate the weight trajectories from each similar system in the least-squares computation of $\widehat{A}$. That is, let $W_\theta = [w_1^{(\theta)}, w_2^{(\theta)}, ..., w_{n-1}^{(\theta)}]$ and $W'_\theta = [w_2^{(\theta)}, w_3^{(\theta)}, ..., w_n^{(\theta)}]$ be the weight trajectory and next weight trajectory for some parameter . Then we solve the least-squares problem $\widehat{A} = [W_{\theta_1}, \ldots, W_{\theta_n}] = [W'_\theta, \ldots, W'_{\theta_n}]$

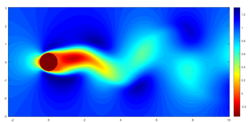
For the sake of defining when it is appropriate to expect our method to be able to generalize across different spatiotemporally evolving systems, we shall define what it means for two fluid flows to be *similar*. In configuring a fluid dynamics simulation, a set of quantifiable parameters are defined. Two dynamical fluid systems S_1 and S_2 are considered *similar* if they have the same configuration of parameters and differ only in the value of at most one parameter. Furthermore, we require that the parameter be continuously variable and that any observable data point in the domain of the system vary smoothly as that parameter varies from its value in S_1 to its value in S_2. For example, for fluids flowing past identical cylinders, the Reynolds number associated with the free stream velocity may be varied to produce similar systems. However, to replace the system's cylinder with a triangle would be to qualitatively change the configuration of the system parameters, and thus would produce a non-similar system.

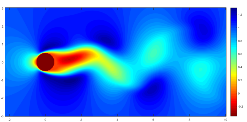
Unlike neural networks, the weights in an E-GP do not exist in some abstract, difficult-to-comprehend space, but are associated with kernel centers in specific locations in the domain. We refer to this attribute of E-GPs as the *spatial encoding* property. This property is an extremely valuable tool for gaining insight into the learned model works. For example, by plotting which kernel centers are associated with which invariant subspaces in the transition matrix, one can visualize where the eigenfunctions are found and how the dynamic modes are separated spatially. For another example, by plotting arrows from center c_j to c_i for each of the largest elements $\hat{a}_{ij}$ of $\widehat{A}$, one can visualize how different areas of the domain influence each other's evolution.

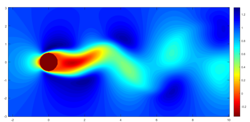
4 Results

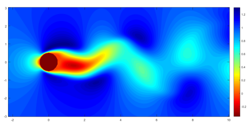
4.1 Modeling the Individual Flows

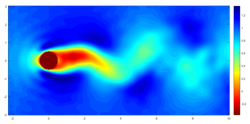
We used CFD methods to generate the states for a canonical fluid mechanics problem: flow past a cylinder at various Reynolds numbers, namely Re = 100, 300, 600, 800 and 1000. This deterministic, high-dimensional spatiotemporal dynamical system is well-studied in the fluid dynamics literature, both experimentally and numerically [27, 28, 29]. In our CFD simulation, we used a 4th-order polynomial expansion with spectral element method on the incompressible Navier-Stokes equation to generate the cylinder flow data. The spatial domain is $[-2, 10] \times [-3, 3]$, excluding the diameter-1 cylinder at the origin. Neumann boundary conditions are applied to the far-field of the cylinder in the y-direction and the outlet of the flow field; and a Dirichlet boundary condition is applied to the inlet. Each data set contains at least 200 snapshots with a uniform time step of $\tilde{0}.03$ sec. Each snapshot contains 24,000 velocity data points for Re=100 or 95,000 velocity data points for Re=300,600,800,1000. Each data set took at least 10 hours in a high performance computer cluster to generate. Figures 1,2(a-d) visualize the horizontal velocity for Re=100 and Re=1000, with red being the greatest negative velocity and blue the greatest positive velocity. The flow is unstable, periodic, and clearly nonlinear.

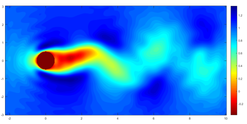

(a) Snapshot 0

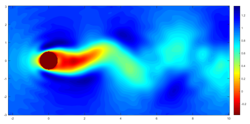

(b) Snapshot 10


(c) Snapshot 20


(d) Snapshot 30


(e) Snapshot 0


(f) Snapshot 10


(g) Snapshot 20

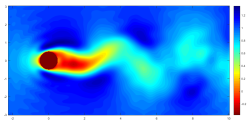
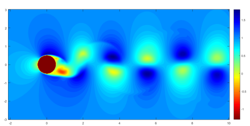
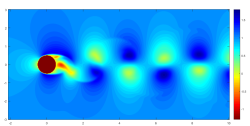
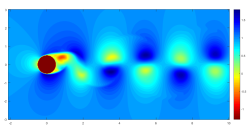
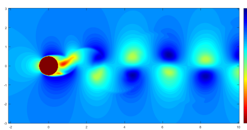

(h) Snapshot 30

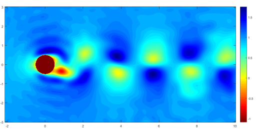
Figure 1: Visualization of Fluid Flow at Re = 100, CFD (a-d), E-GP (e-h)

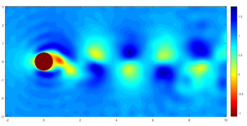

(a) Snapshot 0

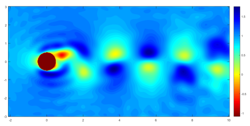

(b) Snapshot 5


(c) Snapshot 10


(d) Snapshot 15


(e) Snapshot 0


(f) Snapshot 5


(g) Snapshot 10

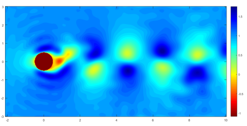
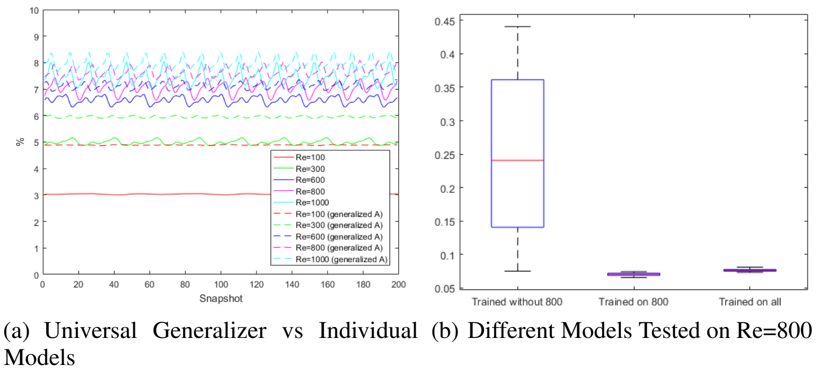

(h) Snapshot 15

Figure 2: Visualization of Fluid Flow at Re = 1000, CFD (a-d), E-GP (e-h)

We used the Gaussian RBF kernel $k(x, y) = e^{-\|x-y\|^2/2\sigma^2}$ in our E-GP model, with σ estimated to be 0.4. Using a budget of 600 kernel centers (see Figure 4(a)-4(b), and note how they cluster in the most dynamic regions), we find a 600×600 matrix $\widehat{A}$ which accurately (Figure 3(a)) captures the dynamics of the nonlinear system. We can use this to propagate single initial condition w_0 forward to make predictions, then compare the predictions to the original training data. We found total percentage errors between 3% for Re=100 and 7-8% for Re=1000, as can be seen in the solid lines in Figure 3(a). We define the total percentage errors as $E_\tau = \frac{\|y_\tau - \bar{y}_\tau\|_2}{\|\bar{y}_\tau\|_2}$ where $\bar{y}_\tau$ is the output vector for time τ and y_τ is the E-GP estimate at that time. Note that *the size of the model has been reduced by almost two orders of magnitude* from the original CFD data. This process takes about 13 minutes in MATLAB for a 200 snapshot by 95,000 point set on an ordinary Intel i7 4.00 GHz processor.

4.2 One Transition Matrix for Everything

In order to approach the challenge of generalizing across similar spatiotemporally evolving systems, the first question we had to answer is whether we can find an $\widehat{A}$ matrix that accurately captures the dynamics of multiple similar flows. The answer to that question is *yes*, using the trajectory concatenation method. Amazingly, a single model generated this way works almost as well on all five data sets as do five individual models trained on each data set separately. This is confirmed by both the total error plots (Figure 3(a)), which show only slight increases in each of the total percentage error plots, and visual inspection of the dynamic modes displayed. This result is even more surprising in light of the fact that the rate of vortex shedding for each Reynolds number is different. By taking a Fourier transform of the time evolution of a data point located at (0.5,8), we find that for the original data sets the vortex shedding frequency is 0.448 Hz, 1.260 Hz, 1.380 Hz, 1.388, and 1.401 Hz for Re=100, 300, 600, 800, and 1000 respectively, and for the E-GP models the frequencies are 0.452 Hz, 1.21 Hz, 1.36 Hz, 1.36 Hz, and 1.36 Hz respectively.



(a) Universal Generalizer vs Individual Models (b) Different Models Tested on Re=800

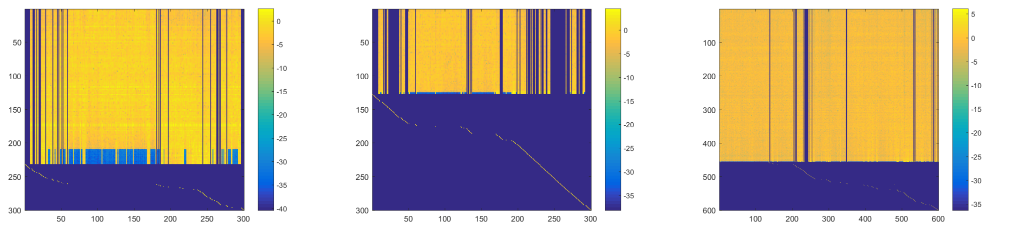
Figure 3: Total Percentage Errors

4.3 Generalizing from Learned Dynamics to Unknown Dynamics

Having seen that it is possible to find a single transition in the weight space that models the dynamics systems over a range of parameters, the next challenge is to be able to model flows with parameters that the model has not been trained on. We derived an $\widehat{A}$ matrix from the Re=100, 300, 600, and 1000 data sets and tested it against the Re=800 data set. The results are below in Figure 3(b). For the first 120 snapshots, the total percentage error remains under 10%, which is satisfactory. After this, however, the total percentage error curves upwards as the slight errors in the transition matrix compound. Over 800 snapshots, we found an average total percentage error of less than 25%.

4.4 Linear Dynamical Layer Analysis & Insights

Due to the spatial encoding of the weights which the linear transition model operates on, we are able to analyze the dynamics and find physical insights into the process. We demonstrate two techniques: (1) using eigendecomosition of the transition matrix to discover the eigenfunctions and invariant subspaces of system, and (2) visualizing the most significant spatial interactions in the system.



(a) Re = 100, ε = 0.005 (b) Re = 1000, ε = 0.05 (c) All Reynolds numbers, ε =0.069

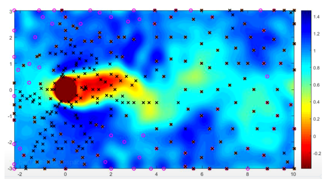
Figure 4: Eigenvector Heat Maps

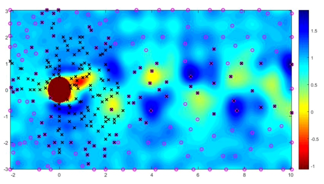
An invariant subspace of a linear operator is a subspace of the Hilbert space such that any vector in the subspace remains in the subspace under transformation by the operator. By marking which kernel centers are associated with different subspaces, we can spatially separate the space into multiple dynamic modules. The physical insight is some areas of the space are dynamically entangled with each other, and other are independent of each other. For those interested in monitoring spatiotemporally evolving systems, the number and location of the invariant subspaces determines how many and where feedback sensors ought to be for robust prediction of the weights.

Before doing the Jordan decomposition of $\widehat{A}$, we zero any elements smaller than some small ε in order to stabilize the algorithm for matrices with many elements close to zero. Afterwards we visualize the eigenvector matrix using a *logarithmic* color chart, as seen in Figures 4(a),4(b),4(c). These plots are for models trained individually on Re=100 and Re=1000 with 300 kernels, and on all five with 600 kernels, for comparison. We see three categories of eigenvector in the rows: (1) Rows at the bottom that have exactly one non-zero elements, (2) In the middle, a couple rows with a dozen significant elements, and (3) at the top a number of rows that affect the majority of the kernel centers in the space.

Each eigenvector of (1) spans its own invariant subspace, and is depicted in magenta circles in Figures 5(a),5(b),5(c). Category (3) is one invariant subspace, depicted with black crosses. Category (2) is subsumed in Category (3). The figures show that the dynamics near/around the cylinder and in its wake are so entangled that a single sensor measurement in that area may be sufficient to estimate over that entire subspace. On the other hand, areas far from the core of dynamic excitement are their own independent, invariant subspaces, and thus must be monitored locally.

Another way to visualize the operation of the linear transition matrix is to plot lines between kernel centers that are influencing each other strongly. That is, if we draw a line center c_j to c_i for each of the (relatively) largest elements a_{ij} of $\widehat{A}$, one can see how the system dynamics are coupled spatially (Figures 6(a),6(b),6(c)). We can also plot the magnitude of $a_i j$ in a third axis for further insight into the most dominant dynamic connection in the system.


(a) Re = 100, ε = 0.005


(b) Re = 1000, ε = 0.05

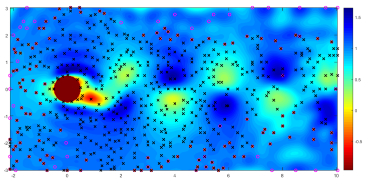
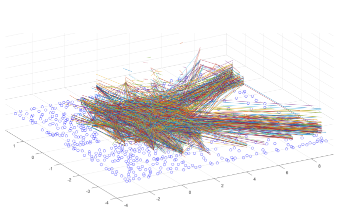
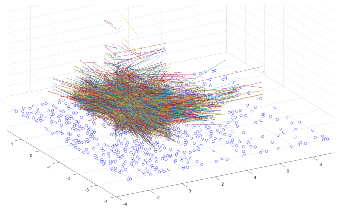

(c) All Reynolds numbers, ε = 0.069

Figure 5: Invariant Subspaces


(a) Re=100


(b) Re=1000

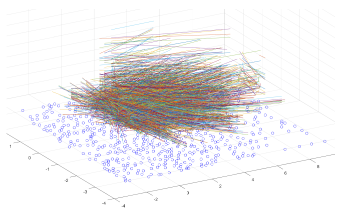

(c) Trained on all 5 data sets

Figure 6: Visualization of Co-Relations in Transition Matrix

5 Conclusion

In this paper we presented a systems-theoretic approach to the problem of modeling complex spatiotemporally evolving phenomena and generalizing across continuously similar systems. Our approach focused on deriving a linear transition matrix in a space of weights layered over a kernel-based model. This was demonstrated on computational flow dynamics data of a fluid moving past a cylinder at various Reynolds numbers. We found that a single model could predict the evolution of the system at five very different Reynolds numbers with almost the same accuracy as a model of the same size trained on only one of the data sets. We also found that our model was able to predict the evolution of similar systems that it had never been trained on.

Acknowledgments

We would like to acknowledge Balaji Jayakumar and Chen Lu from the Department of Mechanical & Aerospace Engineering at Oklahoma State University for their provision of the CFD data sets. This work was sponsored by Air Force Office of Scientific Research Award Number FA9550-14-1-0399

References

[1] T. P. Barnett, D. W. Pierce, and R. Schnur. Detection of anthropogenic climate change in the world's oceans. *Science*, 292(5515):270–274, 2001.

[2] M. J. Heaton, M. Katzfuss, S. Ramachandar, K. Pedings, E. Gilleland, E. Mannshardt-Shamseldin, and R. L. Smith. Spatio-temporal models for large-scale indicators of extreme weather. *Environmetrics*, 22(3):294–303, 2011.

[3] N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.

[4] H. Kingravi, H. Maske, and G. Chowdhary. Kernel observers: Systems theoretic modeling and inference of spatiotemporally varying processes. In *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, 2016. accepted.

[5] H. A. Kingravi, H. Maske, and G. Chowdhary. Kernel controllers: A systems-theoretic approach for data-driven modeling and control of spatiotemporally evolving processes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 7365–7370, Dec 2015. doi:10.1109/CDC.2015.7403382.

[6] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[7] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

[8] C. K. Wikle. A kernel-based spectral model for non-gaussian spatio-temporal processes. *Statistical Modelling*, 2(4):299–314, 2002.

[9] D. Higdon. A process-convolution approach to modelling temperatures in the north atlantic ocean. *Environmental and Ecological Statistics*, 5(2):173–190, 1998.

[10] C. Paciorek and M. Schervish. Nonstationary covariance functions for gaussian process regression. *Advances in neural information processing systems*, 16:273–280, 2004.

[11] C. Plagemann, K. Kersting, and W. Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Machine learning and knowledge discovery in databases*, pages 204–219. Springer, 2008.

[12] S. Garg, A. Singh, and F. Ramos. Learning non-stationary space-time models for environmental monitoring. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

[13] A. M. Schmidt and A. O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):743–758, 2003.

[14] T. Pfingsten, M. Kuss, and C. E. Rasmussen. Nonstationary gaussian process regression using a latent extension of the input space, 2006.

[15] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

[16] I. G. K. Williams, Matthew O. and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6): 1307–1346, 2015.

[17] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7): 808–817, 2000.

[18] C. W. R. Williams, Matthew O. and I. G. Kevrekidis. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2), 2015.

[19] W. L. Guo, Xiaoxiao and F. Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[20] e. a. Cao, Yi. Prediction of convergence dynamics of design performance using differential recurrent neural networks. In *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008.

[21] R. L. Panton. *Incompressible flow*. John Wiley & Sons, 2006.

[22] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[23] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

[24] Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the International Conference on Machine Learning*, 2013.

[25] Z. Yang, A. Wilson, A. Smola, and L. Song. {A la Carte–Learning Fast Kernels}. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 1098–1106, 2015.

[26] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2001.

[27] A. Roshko. On the development of turbulent wakes from vortex streets. *California Institute of Technology*, Report 1191, 1954.

[28] C. P. Braza, M. and H. Minh. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of fluid mechanics*, 165(130), 1986.

[29] K. A. Rajani, B.N. and S. Majumdar. Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3):1228–1247, 2009.

[30] L. Csatö and M. Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.