Author Notes

- ✔ The type has been replaced in your figures; please proof the figures carefully for any errors.
- ✔ The last page contains callouts that might be used in the article, depending on design and spacing considerations. Are these callouts appropriate? If not, please provide several alternatives.
- ✔ When the magazine is printed, you will receive courtesy copies of the issue to distribute to your coauthors. Please reply with your complete mailing address, including phone number.

Evolving Gaussian Processes and Kernel Observers for Learning and Control in Spatiotemporally Varying Domains

APPLICATIONS IN AGRICULTURE, WEATHER MONITORING, AND FLUID DYNAMICS

JOSHUA E. WHITMAN, HARSHAL MASKE, HASSAN A. KINGRAVI, and GIRISH CHOWDHARY

xxxxxx

Monitoring and modeling large-scale stochastic phenomena with both spatial and temporal (spatiotemporal) evolution by using a network of distributed sensors is a critical problem in many control applications (see "Summary"). **<AU: "Summary" must be cited in the article before other sidebars and figures. Is this placement ok?>** Consider, for example, a team of robots that has the task of destroying herbicide-resistant weeds on a farm (see Figure 1 and "Key Control Problems in Agriculture"). This team must predict weed growth across the whole farm to make intelligent, coordinated decisions [1]. However, the robots can observe only a limited part of a field at any time, leading to the critical problem: how can a few robots that can only partially observe a field at any time predict the full state of the spatiotemporally evolving weed growth across the entire area? When building an observer across a spatiotemporal process, which locations should be sampled to obtain the necessary information to render the problem observable?

The goal of this tutorial is to show the steps taken toward addressing this kind of challenging problem. Examples of such challenges abound across many domains, including modeling and monitoring ocean heat content and acidification for oceanography by using a network of satellites and surface sensors [2]; predicting traffic patterns by using data from vehicles, cell phones, and traffic cameras; predicting enemy movements through the use of ground and aerial surveillance; and predicting extreme weather events via data from weather stations and aerial drones [3]. The rapid advances in the computational power of compact systems and robotics as a whole has led to an explosion of real-world applications for such distributed cyberphysical systems.

These types of applications must estimate complex, stochastic dynamics that are distributed through space and time. The key constraint is the number of available

Digital Object Identifier 10.1109/MCS.2020.3032801
Date of current version: xxxxxx

1066-033X/21©2021IEEE

spatially distributed sensors, which is not enough to entirely cover a whole space at any given moment in time. One approach to solving the problem under this constraint is to use a predictive model of the phenomenon that informs our sensing strategy. While the modeling of such spatiotemporal phenomena has traditionally been the object of study in geostatistics, it has (in recent years) gained more attention in the machine learning community [4]. The data-driven models developed by machine learning techniques provide a way to capture complex spatiotemporal phenomena that are not easily modeled by first principles alone.

However, these models are limited by the data sets they are trained on, and the high variability of complex distributed physical systems makes it even more challenging. For example, a model trained via years of weed growth data in one field doesn't necessarily generalize to another year, and a model trained on the past few years of data still cannot reliably predict weather variability in the following year. What is needed is not just a *prediction* system but a *predict-and-correct* system. This tutorial shows how to design just such a system for these kinds of problems. The system utilizes kernel methods for modeling and Bayesian filtering theory for prediction correction, and it exploits the mathematical structure of the regression and dynamics models to place the sensors. In the machine learning community, kernel methods represent a class of well-studied

Summary

This article should be useful for anyone interested in using robots in large-scale environments that are changing in time and space. The methods presented here have been used to help teams of robots monitor and destroy weeds within a field of crops. In general, this article discusses modeling and monitoring complex systems that vary in both space and time, given a limited number of agents or sensors providing measurements spread out across a large area. A novel method for solving this problem is presented, with several tremendously useful properties. First, it can be easily trained and updated even with large, "dirty" data collected at many places at many times. Second, this model lends itself well to the kinds of analyses familiar to the controls community, which means that several formerly very challenging problems become much easier (for example, predicting future evolution, deciding how many sensors are needed and where to place them, and determining the basic structures beneath the system dynamics). As far as it is possible to tell, the methods presented here are unmatched in their scope and power. A graduate-level mathematical background is recommended for this article, and an open code repository is provided for the ease of implementation.

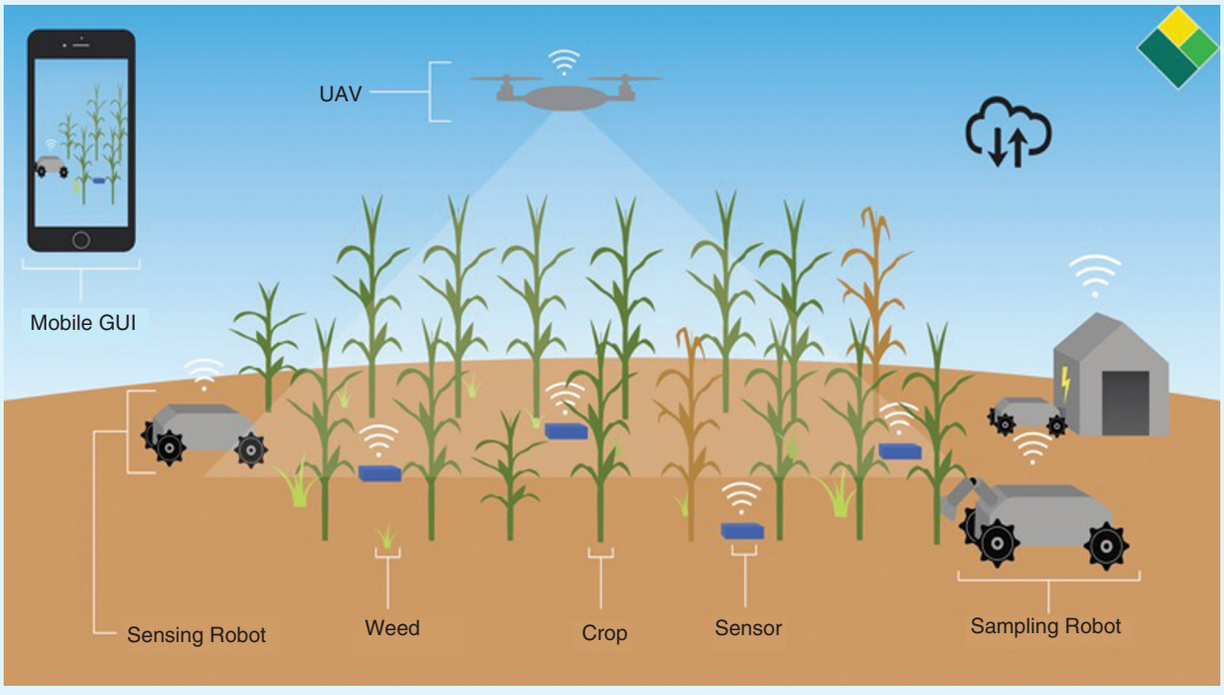

FIGURE 1 A cyberphysical system consisting of a distributed team of robots for mechanical weed management on a farm. GUI: graphical user interface; UAV: unmanned aerial vehicle. (Source: EarthSense; used with permission.)

Key Control Problems in Agriculture

A shortage of qualified human labor is a key challenge facing farmers [S1], [S2], **<AU: Please note references cited in sidebars were removed from the main reference list, renumbered starting with [S1], and placed at the end of the sidebars.>** leading to smaller profit margins and preventing the adoption of truly sustainable agricultural practices. The lack of timely available labor was a principal reason behind the tens of millions of dollars of unharvested fruits and vegetables that rotted on California farms in 2017 [S3], [S4]. Worker shortages can be a major barrier to more sustainable agricultural practices that are labor intensive. For example, with current agricultural equipment, more sustainable alternatives to prevalent methods of agriculture that would not need large amounts of chemicals and other inputs, such as perennial polycultures (mixed species of fruit- and nut-producing trees and shrubs [S5]; see Figure S1), **<AU: Please check whether the preceding edited sentence conveys the intended meaning. Note that Figure S2 is renumbered as Figure S1 as figures need to appear in numerical order in the text.>** are currently impractical at scale. Polyculture systems can leverage the cohabitation of mutually beneficial plants (and animals, insects, and microbiomes) to create a more sustainable engineered ecosystem. The labor shortage has become a primary barrier to the adoption of this sustainable agricultural alternative [S6], [S7]. One way to address the challenges of labor shortages in agriculture is by creating new robotic technology that can work in harsh, uncertain, and dynamically changing field environments. The following sections outline some of the fundamental challenges in autonomy, estimation, and control that the controls community can help overcome to enable the future of agricultural robotics and relates it to the problem of spatiotemporal function estimation studied in this article.

PERSISTENT MULTIAGENT AUTONOMY UNDER PARTIAL OBSERVABILITY

The digital farm of the future will employ teams of distributed heterogeneous agents to autonomously manage, optimize, and harvest large acres of diverse crops during the entire season, without encumbering humans. This level of autonomy in unstructured field environments is beyond the reach of the current state of the art, which requires constant human monitoring and oversight (especially in the presence of change and unforeseen events). Efficient and reliable control will be central

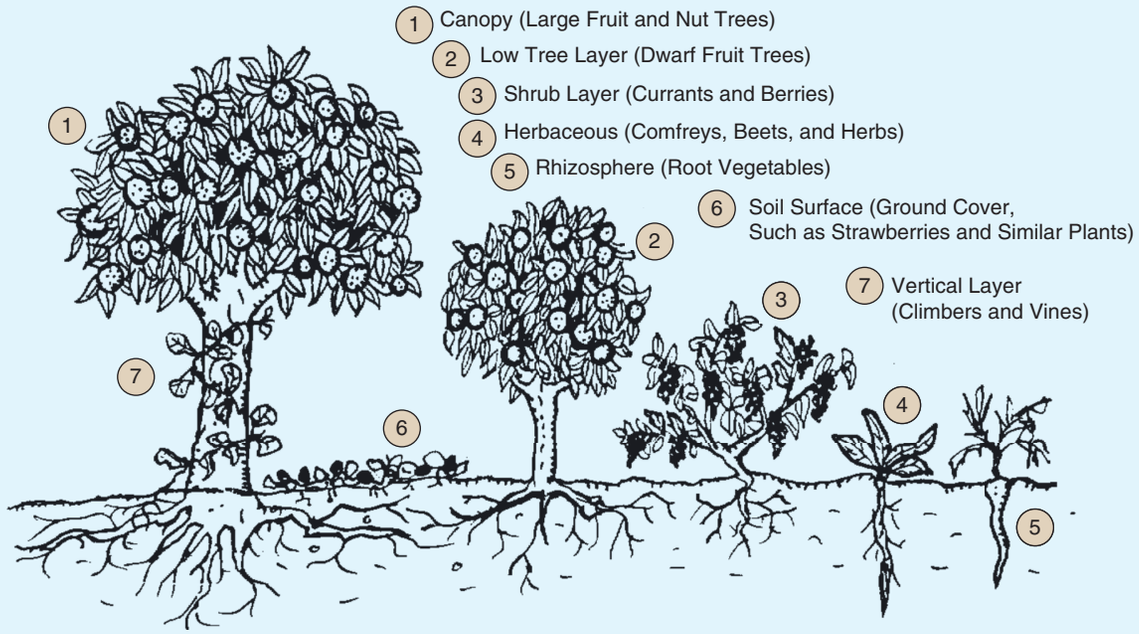

FIGURE S1 The seven layers of the forest garden. Polyculture agricultural production systems are designed ecosystems that can be more productive and sustainable than traditional monoculture systems. Managing such complex systems requires fundamental advances in robotics, spatiotemporal modeling, and controlling complex biological processes, all areas where the control community can help (see [S20]). (Source: [S19].)

to the success of these robots. Small, below-canopy robots (such as in Figure S2 [S8]) must provide precision care, including pruning, weeding, and reseeding without damaging plants and causing soil compaction.

Deployed at scale, these robots can make large-scale organic farming practical and enable enhanced breeding through field-scale phenotyping [S8]–[S10]. The big controls challenge involves making decisions across large spatiotemporal scales while using information obtained from a few stationary and mobile sensors that can only partially observe the environment at any given time. The work pursued in this article lays a foundation for solving this problem by enabling a team of agents to estimate the varying state of the environment.

DEXTEROUS AND UBIQUITOUS ROBOTICS FOR PRECISE CARE

The digital farm of the future will strive to eliminate costly inputs (chemicals, labor, energy, and knowledge) with low-cost, dexterous, and highly autonomous agricultural equipment [S11]. Advances in *soft* arms and grippers can enable robots that have a far better reach and dexterity around plants than robots equipped with traditional *hard* industrial arms. Soft arms, which are often actuated with pressurized tubes, can be far less expensive to manufacture and significantly lighter than their hard counterparts. Alternately, soft arms can be slow to actuate and have limited payloads. To make soft robots practical, optimal feedback control techniques are necessary that work with conformal objects with very large degrees of freedom. Specifically, soft arms tend to significantly deform under weight and behave quite differently when carrying with different payloads. Unlike hard arms, encoders are not sufficient to estimate the pose of the arm and manipulator. Strain and angle sensors must be judicially positioned to keep costs down, and image-based feedback control will be necessary.

The evolving Gaussian process and kernel observer techniques described in this article could be utilized to create distributed observers for such soft systems. The complex interaction between closely spaced, diverse plant species in a polyculture results in both spatial and temporal dynamics as the plants grow and interact with one another. Plant development often exhibits hybrid dynamical systems behavior, with rapid thresholded growth bursts followed by slow progression. The triggers for growth bursts are dependent on 1) environmental factors such as temperature, soil moisture, and sunlight reaching individual and cumulative thresholds and 2) complex interrelations between neighboring plants, soil chemistry, insects, and soil microbes. Simulating individual plant growth is an active area of research with many open questions in modeling and plant biology [S12].

Arguably, very-high-resolution plant growth models may not even be necessary for the effective control of polycultures. However, the existing models of plants and interactions with ecosystems [S13]–[S18] are not well suited for designing and managing polycultures because of the simplifying assumptions that are often made. A good balance could be found with data-driven machine learning models that have sufficient resolution for aggregate prediction across multiple spatiotemporal scales and are lightweight enough for control and decision making. With these models, predictive control strategies can be created that task teams of robots for management duties. Furthermore, these predictive models can enable quantified mechanisms of

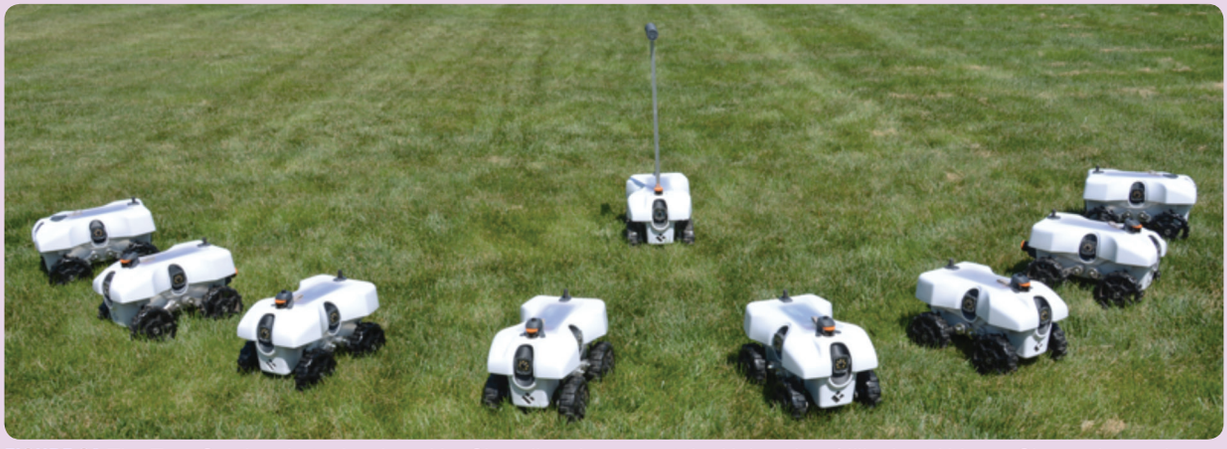

FIGURE S2 The TerraSentia robots developed by Chowdhary's group at the University of Illinois at Urbana-Champaign and commercialized by EarthSense. Agricultural robots such as these present exciting possibilities for distributed agricultural management by using teams of compact, ultralight, under-canopy robots equipped with advanced autonomy and machine learning. Such robots can fill the niche between large farm equipment and manual labor as well as enable perennial polycultures. They can weed gardens. Advances by the controls community in the area of persistent multiagent autonomy in harsh, changing, and uncertain environments are driving exciting possibilities of the future of agriculture. (Source: EarthSense; used with permission.) ***<AU: Please provide a high-resolution (300 dpi or better at size needed) figure file (.jpeg, .tiff, etc.).>***

designing and planning efficient agroecosystems. Obtaining the required data to train these models and using the information to create effective control techniques for managing profitable polycultures remains an exciting direction of future work where the controls community can help. **<AU: Because the acknowledgments presented in this sidebar are included in the "Acknowledgments" section, they have been deleted from here.>**

REFERENCES

[S1] T. J. Richards, "Immigration reform and farm labor markets," *Amer. J. Agric. Econ.*, vol. 100, no. 4, pp. 1050–1071, 2018. doi: 10.1093/ajae/aay027.

[S2] T. Hertz and S. Zahniser, "Is there a farm labor shortage?" *Amer. J. Agric. Econ.*, vol. 95, no. 2, pp. 476–481, 2013. doi: 10.1093/ajae/aas090.

[S3] J. Guthman, "Paradoxes of the border: Labor shortages and farmworker minor agency in reworking California's strawberry fields," *Econ. Geogr.*, vol. 93, no. 1, pp. 24–43, 2017. doi: 10.1080/00130095.2016.1180241.

[S4] C. Morris. "California crops rot as immigration crackdown creates farmworker shortage." Fortune. http://fortune.com/2017/08/08/immigration–worker–shortage–rotting–crops/ **<AU: Please provide the date of access.>**

[S5] S. T. Lovell et al., "Temperate agroforestry research: Considering multifunctional woody polycultures and the design of long-term field trials," *Agroforestr. Syst.*, vol. 92, no. 5, pp. 1397–1415, 2017. doi: 10.1007/s10457-017-0087-4.

[S6] D. Mitchell. "Labor shortage provides obstacle for berry growers." The Packer. https://www.thepacker.com/article/labor-shortage-proves-obstacle-berry-growers#:~:text=Labor%20shortage%20proves%20obstacle%20for%20berry%20growers%20Diamond,of%20the%20New%20Jersey%20Blueberry%20Industry%20Advisory%20Council **<AU: Please confirm the URL and provide the date of access.>**

[S7] G. Mohan. "As California's labor shortage grows, farmers race to replace workers with robots." *LA Times*. https://www.latimes.com/business/la-fi-farm-mechanization-20170721-story.html#:~:text=As%20California%E2%80%99s%20labor%20shortage%20grows%2C%20farmers%20race%20to,in%20May%20%28Gary%20Coronado%20%2F%20Los%20Angeles%20Times%29 **<AU: Please confirm the URL and provide the date of access.>**

[S8] E. Kayacan, Z. Zhang, and G. Chowdhary, "Embedded high precision control and corn stand counting algorithms for an ultra-compact 3d printed field robot," in *Proc. Robot., Sci. Syst.*, Pittsburgh, 2018. **<AU: Please provide the page range.>**

[S9] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The Robotanist: A ground-based agricultural robot for high-throughput crop phenotyping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Singapore, 2017, pp. 3634–3639. doi: 10.1109/ICRA.2017.7989418.

[S10] N. Virlet, K. Sabermanesh, P. Sadeghi-Tehran, and M. J. Hawkesford, "Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring," *Funct. Plant Biol.*, vol. 44, no. 1, pp. 143–153, 2017. doi: 10.1071/FP16163.

[S11] S. M. Pedersen, S. Fountas, H. Have, and B. S. Blackmore, "Agricultural robots–system analysis and economic feasibility," *Precis. Agric.*, vol. 7, no. 4, pp. 295–308, 2006. doi: 10.1007/s11119-006-9014-9.

[S12] X.-G. Zhu, J. P. Lynch, D. S. LeBauer, A. J. Millar, M. Stitt, and S. P. Long, "Plants in silico: Why, why now and what?–an integrative platform for plant systems biology research," *Plant, Cell Environ.*, vol. 39, no. 5, pp. 1049–1057, 2016. doi: 10.1111/pce.12673.

[S13] E. Stehfest, M. Heistermann, J. A. Priess, D. S. Ojima, and J. Alcamo, "Simulation of global crop production with the ecosystem model DayCent," *Ecol. Model.*, vol. 209, no. 2-4, pp. 203–219, Dec. 2007. doi: 10.1016/j.ecolmodel.2007.06.028.

[S14] J. A. Foley et al., "An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics," *Global Biogeochem. Cycles*, vol. 10, no. 4, pp. 603–628, Dec. 1996. doi: 10.1029/96GB02692.

[S15] C. J. Kucharik, "Evaluation of a process-based agro-ecosystem model (Agro-IBIS) across the U.S. corn belt: Simulations of the interannual variability in maize yield," *Earth Inter.*, vol. 7, no. 14, pp. 1–33, Dec. 2003. doi: 10.1175/1087-3562(2003)007<0001:EOAPAM>2.0.CO;2.

[S16] M. A. Friedl et al., "MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets," *Remote Sens. Environ.*, vol. 114, no. 1, pp. 168–182, 2010. doi: 10.1016/j.rse.2009.08.016.

[S17] M. A. Rodrigues, D. M. Lopes, M. S. Leite, and V. M. Tabuada, "Calibration and application of FOREST-BGC in NorthWestern of Portugal," in *Proc. EGU General Assembly Conf. Abstracts*, 2010, vol. 12, p. 7270.

[S18] L. Nunes, M. A. Rodrigues, and D. Lopes, "Evaluation of the climate change impact on the productivity of Portuguese pine ecosystems using the forest-BGC model," in *Advances in Meteorology, Climatology and Atmospheric Physics*, C. Helmis and P. Nastos, Eds. New York: Springer-Verlag, 2013, pp. 655–661.

[S19] G. Burnett. "The seven layers of the forest garden." Accessed on: Dec. 2018. [Online]. Available: https://en.wikipedia.org/?title=Forest_gardening#/media/File:Forgard2-003.gif

[S20] C. J. Rhodes, "Feeding and healing the world: Through regenerative agriculture and permaculture," *Sci. Progr.*, vol. 95, no. 4, pp. 345–446, 2012. doi: 10.3184/003685012X13504990668392.

and powerful methods for regression in spatial domains. In these techniques, correlations between the input variables are related via covariance kernels, and the model is generated by a linear combination of the kernels [5]–[7].

In recent years, kernel methods have been applied to spatiotemporal regression problems with varying degrees of success [4], [5]. Many recent approaches have focused on nonstationary covariance kernel design and algorithms for learning the associated hyperparameters [8]–[11]. These techniques, which focus on the careful design of covariance kernels, have been proposed as an alternative to the naive approach of simply including time as an additional input dimension in the kernel [12]. The careful design/optimization of a covariance kernel avoids an explosion in the number of parameters used by the model, which would be inevitable in the naive approach, and can better account for spatiotemporal couplings. Such covariance kernels, however, do not scale in the face of large-scale phenomena since the optimization of the kernel hyperparameters is nonconvex and computationally demanding for large data sets [13]. Deep learning also suffers from similar issues and, moreover, lacks the spatial encoding properties of certain kernels, which are exploited by the strategy outlined in this tutorial. No matter how many training data are used, the model cannot completely capture the variability of real-world systems that have complex spatiotemporal dynamics.

This is a problem that the controls community is quite aware of; the practitioners' solution is built on feedback, leading to fundamental notions of observability and controllability that can be used to build robust state estimators and controllers. To bring these ideas to fruition in the spatiotemporal problem, a method was needed to determine where sensing/control should be performed to ensure that

the state estimation problem can be made observable/controllable. While approaches such as [11] have succeeded in using nonstationary kernels that evolve efficiently by using feedback, it is unclear how notions of observability and controllability could be utilized with [11] and other existing kernel-based machine learning models and how observers and controllers can be embedded within such models. Computational challenges can be addressed with faster methods and by increasing computational power. However, addressing the latter, more fundamental challenge in designing robust observers/controllers is particularly important in the design of reliable engineering systems, such as distributed sensor/actuator networks intended for monitoring physical phenomena, autonomous soft robots, and other physical systems that have distributed sensing and actuation.

CONTRIBUTIONS

This tutorial presents a different perspective on solving the spatiotemporal monitoring problem that brings together kernel-based modeling, systems theory, and Bayesian filtering. The monitoring problem is defined as follows: *given an approximate predictive model of the spatiotemporal phenomena learned using historical data, estimate the current latent state of the phenomena in the presence of uncertainty, employing as few sensors as possible*. Ideally, the solution to the problem should also provide guidance on how many sensors are needed and where to place them. This article argues that, when it comes to predictive inference across spatiotemporal phenomena, a Kalman filter-type approach of predicting and correcting with feedback from a set of minimal sensors is a robust way of dealing with real-world uncertainties and inherent modeling errors.

In the context of this specific problem, the main contributions are twofold. First, it is demonstrated that spatiotemporal functional evolution can be modeled using stationary kernels that have a linear dynamical systems layer on their mixing weights. In contrast to existing work, this approach does not necessarily require the design of complex spatiotemporal kernels, and it can accommodate positive-definite kernels on any domain in which it is possible to define them (which includes non-Euclidean domains such as Riemannian manifolds, strings, graphs, and images [14]). Second, it is shown that such a model can be utilized to determine sensing locations that guarantee that the hidden states of functional evolution can be estimated using a Bayesian state estimator (Kalman filter), which is embedded in the feature space of the kernel model with very few sensors.

A benefit of this solution's approach is that it provides guidance on how many sensors are needed and where to place them. Accordingly, sufficient conditions are provided regarding the required number and location of sensor measurements, and nonconservative lower bounds on the minimum number of sampling locations are proved by developing fundamental results for the observability of kernel-based models. Our model is also analyzed in terms of Koopman operator theory, and several key theoretical results are proved, demonstrating that the model can produce the Koopman modes, eigenvalues, and eigenfunctions. The validity of the presented model and sensing techniques is corroborated using synthetic and large real data sets.

BROADER CONTEXT

The fundamental idea of building observers and controllers embedded in the feature spaces of machine learning models introduced in this article is generalizable beyond the particular application of spatiotemporal monitoring. Figure 2 presents a general landscape of problems relevant for engineering. Since the controls literature is strongest when the system dynamics can be represented as ordinary differential equations (ODEs), some of the major successes of controls have included results such as linear quadratic Gaussian control, reinforcement learning, and adaptive control in state spaces with well-defined, finite, and *physically meaningful* state variables. The estimation of the states of these temporally evolving, finite-dimensional state-space systems have been extensively studied in the context of Kalman filtering and observer design [15].

A different approach for modeling complex spatiotemporal dynamical systems comes from machine learning, where trained models reside in abstract feature spaces that are relatable only to physical quantities through complex functional operations (see "Feature Spaces in Machine Learning"). There has been some work that relates approaches from controls to machine learning (see, for example, [16] for an extension of the Kalman filter to the functional domain). However, the results are not studied in the context of the spatiotemporal monitoring problem presented here. To fuse machine learning with controls for building robust systems for engineering, fundamental questions must be answered, such as 1) the smallest number of sensors required to observe a distributed system, 2) the placement of sensors/actuators to guarantee observability/controllability of the system, and 3) the effect of random-sensor placement on system observability/controllability.

This tutorial presents an approach that can provide one formal way of addressing these and other questions about complex systems that are modeled with machine learning. Specifically, it is demonstrated how linear dynamical systems can be embedded in the reproducing kernel Hilbert space (RKHS) [6], [17], [18] generated by features used in Gaussian process (GP) modeling [19], [20] and utilized to answer fundamental questions, such as controllability and observability. It is expected that follow-up work will exploit the framework presented in this article by utilizing linear models in RKHSs and the feature spaces of other machine learning models to enable practical and analyzable data-driven engineering systems. To facilitate the development

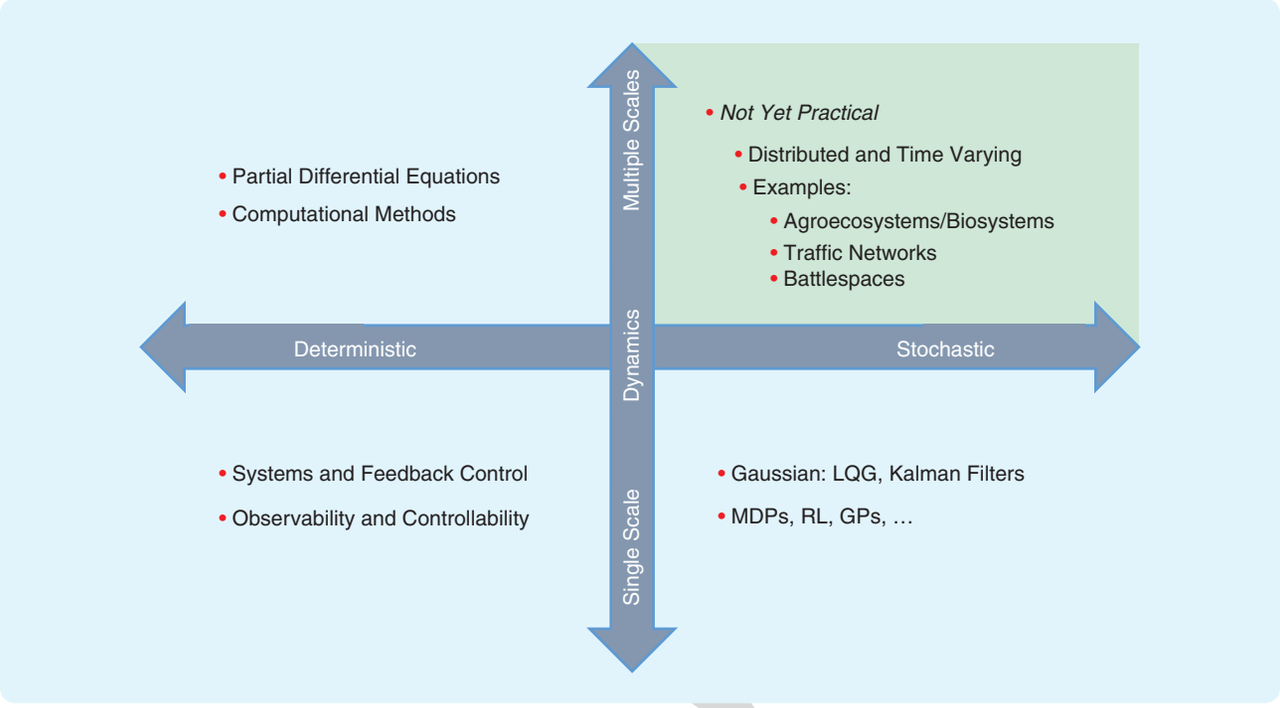

FIGURE 2 Modeling, monitoring, and controlling dynamical systems with complex and uncertain dynamics (such as agricultural, traffic, and monitoring systems) presents an exciting open challenge for the controls community. The bottom-left quadrant describes linear and time-invariant systems with single-scale dynamics, for which the theory of feedback control of dynamical systems is often sufficient. The bottom-right quadrant shows stochastic single-scaled systems, where approaches such as Kalman filters and Gaussian optimization have marked several control systems successes, enabling endeavors from lunar landings to Global Positioning System navigation. The top-left quadrant denotes systems with dynamics at multiple scales, where the efficient computation of solutions to partial differential equations is a highly active area of research. Fundamental theoretical advances and practical algorithms are needed, however, to enable autonomous decision making for distributed stochastic cyberphysical systems with dynamics at multiple scales (shown in the top-right quadrant), such as distributed agricultural robotic systems, traffic networks, and weather monitoring systems with mobile and stationary sensors. LQG: linear-quadratic-Gaussian; MDP: Markov decision process; RL: reinforcement learning; GP: Gaussian process.

of the theory, this article focuses on the problem of monitoring spatiotemporal phenomena. However, the idea should be generalizable to any distributed cyberphysical system that is changing through space and time.

ARTICLE OUTLINE AND RELATIONSHIP TO PRIOR WORK BY THE AUTHORS

The rest of the article begins by summarizing related work in machine learning in this area. "Feature Spaces in Machine Learning" discusses the concept of feature spaces and its importance in machine learning in a broader context. The problem is then formulated, *kernel observers* are introduced, and the main theoretical and algorithmic results are developed. This is followed by results for the expected number of randomly placed sensors required to monitor a spatiotemporal process in the context of our model. An extension to the kernel observer method, called *evolving GPs (E-GPs)*, is presented that learns one model for multiple similar spatiotemporal processes [their efficacy for real-world computational fluid dynamics (CFD) data is presented in "Learning Fluid Flows With Evolving Gaussian Processes"]. Elements of the work presented in this article first appeared at the 2016 Conference on Neural Information Processing Systems [21], [22], the 2015 IEEE Conference on Decision and Control [23], the 2017 Conference on Robot Learning [24], and the 2018 American Control Conference [25].

This article presents a comprehensive set of results and fills the missing links in a single encompassing publication, and it introduces new results for observability in the presence of random-sensor placement and connections to Koopman operator theory. Therefore, it primarily focuses on the fundamental theory and practical algorithms for modeling, estimation, and control, while the details of how to optimally implement the presented algorithms are omitted. Instead, an open source code base is made available in Matlab at http://daslab.illinois.edu/software.html **<AU: Please provide an updated URL as this one does not work.>** and https://github.com/hkingravi/FunctionObservers and in Python at https://github.com/hkingravi/funcobspy.

RELATED WORK

There is a large amount of literature about GPs and spatiotemporal modeling, a complete survey of which is beyond the scope of this article. Since our contributions are in the

area of creating a feedback-based observer in the feature spaces of GP models, related work is discussed in three areas: spatiotemporal modeling with GPs, the connection of GPs to Kalman filtering, and sensor placement for inference in spatiotemporal domains.

The use of process-dependent kernels for spatiotemporal modeling in geostatistics is well studied [4], [26], [27]. Other approaches that utilize the hierarchy and the evolution of kernels have also been used for modeling spatiotemporal functions [28]–[30]. From the machine learning perspective, a naive approach is to utilize both spatial and temporal variables as inputs to a Mercer kernel [31]. However, this technique leads to an ever-growing kernel dictionary. Furthermore, constraining the dictionary size and utilizing a moving window will occlude the learning of long-term patterns. A cleaner approach is to use state-space representations of time-varying GPs [28], [32]. From this viewpoint, each GP instance is viewed as a snapshot of an evolving set of weights. We follow in a similar vein here, with added emphasis on exploiting mathematical structures relevant to observability and controllability.

Periodic and nonstationary covariance functions and nonlinear transformations have been proposed for spatiotemporal modeling [5], [9]. Work focusing on nonseparable and nonstationary covariance kernels seeks to design kernels optimized for environment-specific dynamics and tune the kernels' hyperparameters in local regions of the input space. Seminal work in [33] proposes a process convolution approach for space–time modeling. This model captures nonstationary structure by enabling the convolution kernel to vary across the input space. This approach can be extended to a class of nonstationary covariance functions, thereby facilitating the use of a GP framework, as shown in [34]. However, since this model's hyperparameters are inferred using Markov chain–Monte Carlo (MCMC) integration, its application has been limited to smaller data sets. To overcome this restriction, [10] proposes to use the mean estimates of a second isotropic GP (defined across latent length scales) to parameterize the nonstationary covariances.

Finally, [8] considers nonisotropic variation across different dimensions of the input space for the second GP, as opposed to the isotropic variation by [10]. Issues with this line of approach include the nonconvexity of the hyperparameter optimization problem and the fact that selecting an appropriate nonstationary covariance function for the task at hand is a nontrivial design decision (as noted in [35]). Apart from directly modeling the covariance function using additional latent GPs, there exist several other techniques for specifying nonstationary GP models. One maps the nonstationary spatial process into a latent space, in which the problem becomes approximately stationary [36]. Along similar lines, [37] extends the input space by adding latent variables, which enables the model to capture nonstationarity in the original space. Both these approaches require MCMC sampling for inference and thus are subject to the limitations mentioned in the preceding paragraph. A geostatistics approach that finds dynamical transition models on the linear combination of weights of a parameterized model [4], [16] is advantageous when the spatial and temporal dynamics are hierarchically separated, leading to a convex learning problem. This technique has been utilized in magnetic resonance imaging [38], [39]. As a result, complex nonstationary kernels are often not necessary (although they can be accommodated). This method is essentially the starting point of this tutorial.

A systems-theoretic study of this viewpoint enables our fundamental contributions, which are 1) enabling inference in more general domains with a larger class of basis functions than those typically considered in the geostatistics community and 2) quantifying the minimum number of measurements required to estimate the state of the system. Kalman filtering in the context of GP and kernel models has also been quite widely studied [27], [40]–[43]. There is a direct link between the Bayesian approach to inference taken in GPs and its natural extension to Kalman filters. The contributions here are in creating explicit connections between feedback observers and inference by deriving conditions for observability in the kernel space. This leads to explicit conditions for the number of sensors required and where to place them.

Sensor placement optimization is also a well-studied area. Examples include, but are not limited to, 1) geometric approaches (which seek to cover the operating space without making assumptions about the spatiotemporal dynamics [44]) and 2) information-theoretic techniques (which place their focus on sensor placement-optimizing strategies based on mutual information and information entropy for GP models [45]). It should be noted that the contribution of this article concerning sensor placement is to provide *sufficient conditions* for monitoring, rather than optimizing the placement locations, and therefore a comparison with these approaches is not considered in the experiments.

This article is connected to the large body of literature produced during the past decade about Koopman operator theory and dynamic mode decomposition (DMD), particularly in the CFD community. These methods rely on discovering *modes* of motion, which show the spatial distribution, oscillation frequency, and growth rate/decay of the component dynamics of the system. Many applications have been realized through these methods, including the ability to transform the state space so the dynamics appear linear, predicting the temporal evolution of the linear system, reconstructing the state of the original nonlinear system, and even implementing controller design. DMD is the most widely used method for finding a finite-dimensional subspace of the Koopman operator's infinite-dimensional domain to work in [46]. Williams et al. recently integrated DMD with the kernel trick, enabling the algorithm to be extended to systems with much larger

Feature Spaces in Machine Learning

Assume data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, where $x_i \in \Omega_I$ and $y_i \in \Omega_O$. Here, Ω_I is the input domain, and it is generally a subset of $\mathbb{R}^D$, although more general sets, such as discrete spaces, graphs, and text documents, can be considered. Similarly, Ω_O (the output domain) can be just as general as Ω_I. The goal is to solve for functions f in some space of functions $\mathcal{J}$ such that $f(x_i) = y_i \; \forall i$. Generally, to restrict the complexity of the space $\mathcal{J}$, a *loss function* $L(f, \mathcal{D}) \mapsto \mathbb{R}$ is chosen that measures the error between a prediction $f(x_i)$ (given a data point x_i) and y_i (averaged across the entire data set $\mathcal{D}$). The optimization problem becomes

$$f^* = \operatorname*{argmin}_{\mathcal{J}} L(f, \mathcal{D}) + \lambda g(f), \tag{S1}$$

where $\lambda \in \mathbb{R}$, and $g(f_i)$ represents some constraints on the function f, such as smoothness.

Control theorists are most likely familiar with input–output pairs, where $x_i \in \mathbb{R}^N$ and y_i is either in $\mathbb{R}$ or $\mathbb{R}^M$ (*regression*). In machine learning, the most common task occurs when the y_i are discrete (*classification*). Different combinations of tasks, loss functions, and spaces $\mathcal{J}$ result in various algorithms to solve these problems (which can sometimes form entire subfields of machine learning). The choice of the function space $\mathcal{J}$ can be critical for the task to perform, similar to how the choice of the state space is in control theory. Consider a simple example. Assume there are data from two *classes* $\mathcal{D}_A = \{(x_1^A, y_1^A), \ldots, (x_N^A, y_N^A)\}$ and $\mathcal{D}_B = \{(x_1^B, y_1^B), \ldots, (x_N^B, y_N^B)\}$, where $x_i^{\{A,B\}} \in \mathbb{R}^D$ and $y_i^{\{A,B\}} \in \{-1, +1\}$ (shown in Figure S3). Let f be chosen from the class of linear algorithms; that is, $f = w^T x + b$, where $w \in \mathbb{R}^D$ and $b \in \mathbb{R}$. A loss L is selected that returns a loss of zero when the prediction is the correct class; if the prediction is the incorrect class, it returns a higher value for misclassifications that are closer to the boundary. A classic example of such a loss is that used by the *perceptron algorithm*, which can be written as

$$L(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \max(0, -y_i w^T x). \tag{S2}$$

This loss measures how accurate the prediction of the perceptron is on average. The general algorithm is as follows:

1) Initialize $w \in \mathbb{R}^D$ to all zeros.
2) For a fixed number of iterations or until some stopping criterion is met:
   a) For each training example (x_i, y_i),
      i) Let $\hat{y}_i = \operatorname{sgn}(w^T x_i)$.
      ii) If $y_i \neq \hat{y}_i$, update $w \leftarrow w + y_i x_i$.

The perceptron was one of the first machine learning models and the genesis of modern neural networks [S21]. Figure S3 shows where the perceptron algorithm can solve for the decision boundary with zero error. However, if the structure of the data has some nonlinearities, no solution will be found (as seen in Figure S4). In this case, the original space where the data reside is, in some sense, not a rich enough representation. If a mapping of the data to a different space can be constructed that gives a learning algorithm more degrees of freedom to work with, then linear algorithms can still be deployed. If the same data are mapped using a nonlinear map $\phi(x, y) := (x^2, y^2, 2xy)$, the perceptron now finds a solution in three dimensions, as observed in Figure S5. This example shows why so much of the work in machine learning focuses

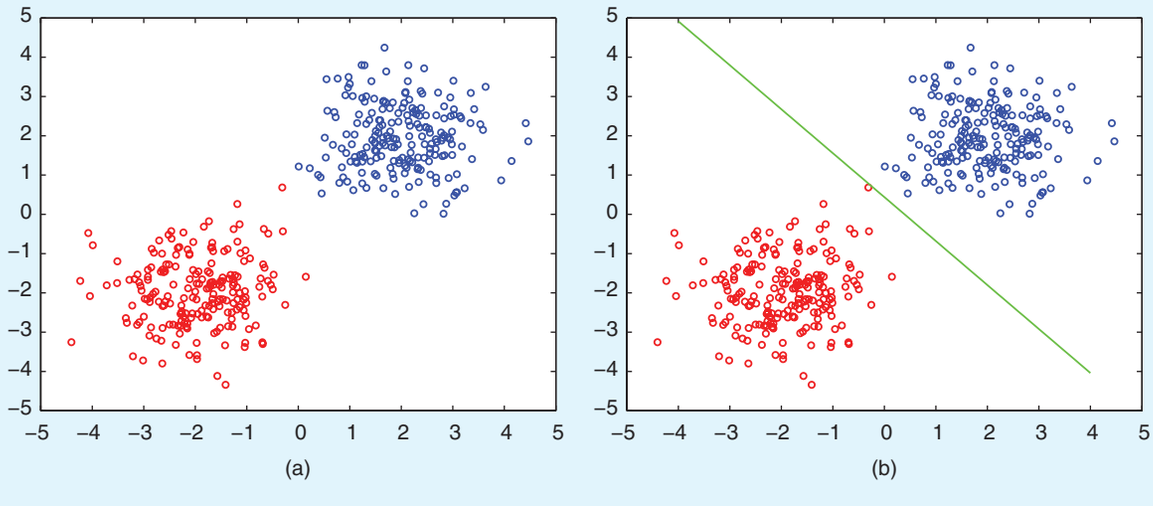

FIGURE S3 An example of linearly separable data. Any simple linear learning algorithm (for example, a perceptron) finds a solution. (a) Data from classes *A* and *B*. (b) The linear boundary separating data.

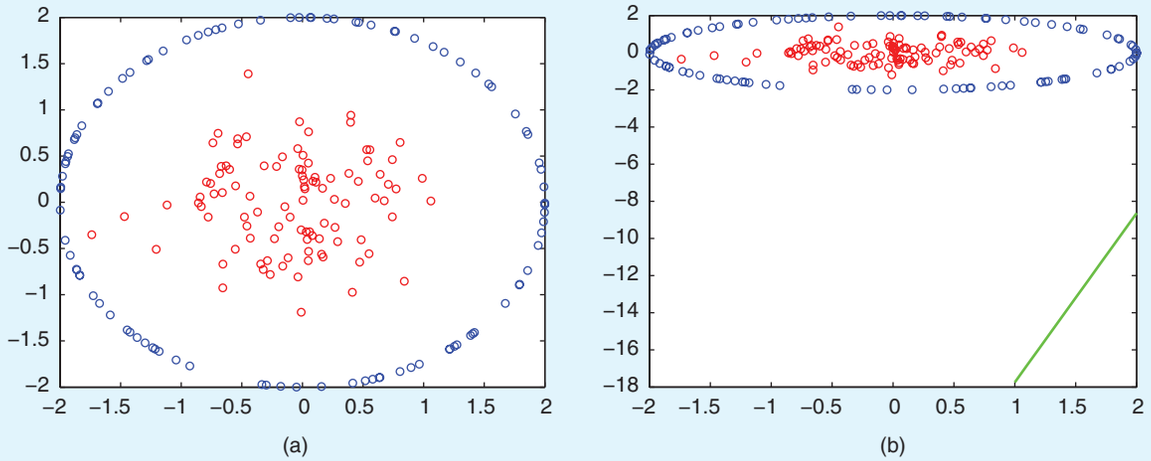

FIGURE S4 An example of nonlinearly separable data. A perceptron fails to find a solution and diverges. (a) Data from classes *A* and *B*. (b) The linear boundary fails to separate data.

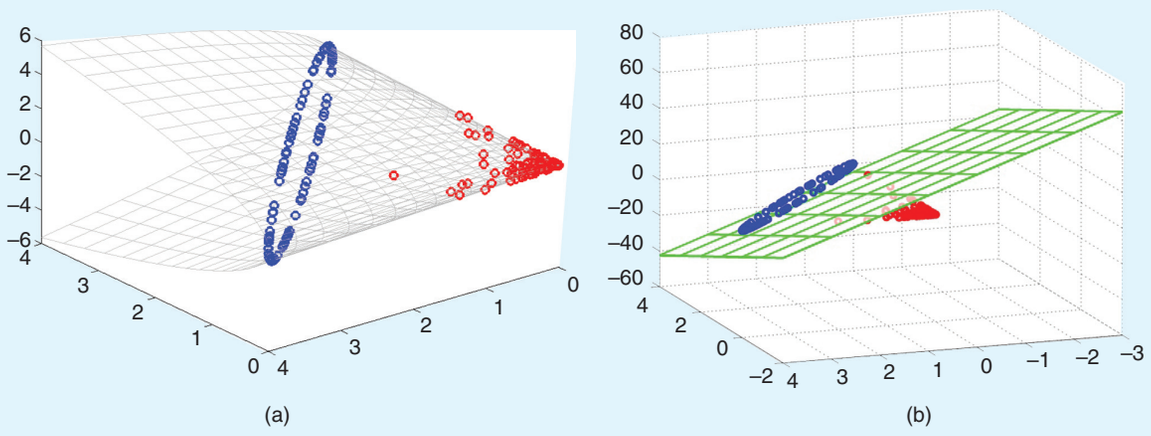

FIGURE S5 Mapping the same data using a nonlinear map $\phi(\mathrm{x},\mathrm{y}) := (x^2, y^2, 2xy)$, the perceptron now finds a solution in three dimensions. (a) The nonlinear mapping of data. (b) The linear boundary in new space.

on learning the right representation for the data, as this simplifies the classification task.

Two major threads of research in the arena of feature maps during the past 40 years are kernel methods and neural networks, the latter of which has gained remarkable notoriety since 2010. These lines of research represent distinctly different strategies for generating feature maps from data. In Figure S4, the data were mapped using an explicit feature map. Kernel methods, of which Gaussian processes (GPs) are a great example, utilize an elegant strategy for generating feature maps from data by using a remarkably simple technique called *the kernel trick*. Given a positive definite kernel function $k(x, y): \Omega \times \Omega \to \mathbb{R}$, Mercer's theorem guarantees the existence of a feature map $\psi: \Omega \to \mathcal{H}$, where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), and the map ψ obeys the property

$$k(x, y) = \langle \psi(x), \psi(y) \rangle_{\mathcal{H}}. \tag{S3}$$

Recall that since $\mathcal{H}$ is an RKHS, given $c \in \Omega$, $k(x, c) = \langle \psi(x), \psi(c) \rangle_{\mathcal{H}}$ and $k(x, c) := \psi_c \in \mathcal{H}$. Furthermore, $\mathrm{span}\{\psi_x\}_{x \in \Omega}$ is dense in $\mathcal{H}$. There exist kernels that generate $\mathcal{H}$s that are extremely high dimensional; for example, the radial basis function kernel $k(x, y) = e^{-\gamma \|x-y\|^2}$ is infinite dimensional. This high degree of freedom enables the design of powerful learning

algorithms that are linear in $\mathcal{H}$ but nonlinear in the input domain Ω. The canonical example of this is the support vector machine [S22]. However, a more instructive example is the perceptron algorithm. Assume there exists a trained perceptron using $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \mathbb{R}^D$ for some D, $y_i \in \{-1, 1\}$. The prediction of the perceptron is $\hat{y} = \operatorname{sgn}(w^T x)$, where $w \in \mathbb{R}^D$. It can be shown that $w = \Sigma_{i=1} \alpha_i y_i x_i$, where α_i is the number of times x_i was misclassified. This facilitates the derivation of the dual version of this algorithm because

$$\hat{y} = \operatorname{sgn}(w^T x) = \operatorname{sgn} \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle_{\mathbb{R}^D}.$$

The dot product $\langle x_i, x \rangle_{\mathbb{R}^D}$ can be replaced with the kernel, leading to the *kernel perceptron algorithm*:

1) Initialize $\alpha \in \mathbb{R}^N$ to all zeros.
2) For a fixed number of iterations or until some stopping criterion is met:
   a) For each training example (x_j, y_j),
      i) Let $\hat{y}_i = \operatorname{sgn} \Sigma_{i=1}^{n} \alpha_i y_i k(x_i, x_j)$.
      ii) If $y_i \neq \hat{y}_i$, update $\alpha_j \leftarrow \alpha_j + 1$.

This algorithm is nonlinear in the input domain but linear in the feature space $\mathcal{H}$ (which led the deep learning community to, somewhat pejoratively, label this as an example of a *shallow learning architecture*). Kernel methods can also be used in a more direct fashion. If there is a subspace $\mathcal{H}' \subset \mathcal{H}$ with a basis generated from $\mathcal{C} = \{c_1, \ldots, c_M\}$, that is, $\mathcal{H}' = \operatorname{span}\{\psi_{c_1}, \ldots, \psi(c_M)\}$, a linear model in $\mathcal{H}'$ is again given by a vector $w \in \mathbb{R}^M$. Suppose this weight vector represents a boundary in $\mathcal{H}'$. To compute which side of this boundary a point x would lie on in $\mathcal{H}'$, simply compute $\operatorname{sgn}(\Sigma_{i=1}^{M} w_i \langle \psi(x), \psi(c_i) \rangle_{\mathcal{H}}) = \operatorname{sgn}(\Sigma_{i=1}^{M} w_i k(x, c_i))$. The choice of the kernel and its parameters depends on the data set and the loss function. The kernel and the data together form the feature space. Because kernel methods are linear in their parameters and restricted to RKHSs, they are amenable to somewhat straightforward mathematical analysis and very well studied because of this.

The very recent review of GPs in control that appeared in *IEEE Control Systems* contains further details, examples, and pointers to software relating to GPs and their use in control [19]. Attention has now shifted to a different way of obtaining the features: deep neural networks (DNNs). While GPs build features using positive–semidefinite symmetric kernels that compare any two points, DNNs build features using nested nonlinear operations. DNNs are models in which the representing function f has nested nonlinearities. Fix a width $M \in \mathbb{N}$. Deep nets are parameterized models with weight matrices $W^l \in \mathbb{R}^{M \times M}$, bias vectors $b^l \in \mathbb{R}^M$, and a pointwise nonlinearity $\phi: \mathbb{R} \to \mathbb{R}$, with $l = 1, \ldots, L$. Vectors $h^l \in \mathbb{R}^M$ are called preactivations, and $x^l \in \mathbb{R}^M$ are called postactivations, each element of which is called a neuron. Let $h^0 \in \mathbb{R}^M$ be the input. The canonical feedforward neural network is given by

$$x^l = \phi(h^l),\ h^l = W^l x^{l-1} + b^l.$$

Therefore, $f(h^0) = x^l$. The individual steps l are called the *layers* of the network, and the nesting property enables these networks to learn much more complicated functions than shallow architectures given the same number of nonlinearities [S23]. Different choices of nonlinearities, connections, and layer architectures lead to different types of neural networks, which are used for different applications [S24]. Deep learning has had an enormous impact on both the machine learning literature and industrial applications, and that impact has rapidly bled over to other fields. Due to the nested structure of nonlinearities in DNNs, the networks are more difficult to analyze using simple mathematical tools. Therefore, most of the literature in the field has focused on the empirical performance of these methods, where they significantly outperform competing methods.

The significance of the achievements of machine learning, in general, and deep learning, specifically, has been in their ability to simplify the implementation of complex functional representation—essentially, in their ability to make system identification more accessible for difficult problems. However, the feature spaces generated by deep networks are not as well behaved and accessible to analysis as those generated by kernel models. Specifically, deep network feature spaces do not naturally have the properties of RKHSs. This presents one barrier to analyzing and understanding the nature of these models. For the reliable and verifiable inclusion of DNNs in engineering control systems, further insights into the structure of the feature spaces these models generate are necessary to restrict certain forms of output and to shape decision making.

REFERENCES

[S21] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958. doi: 10.1037/h0042519.

[S22] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995. doi: 10.1007/BF00994018.

[S23] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009. doi: 10.1561/2200000006.

[S24] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA: MIT Press, 2016.

dimensions [47]. Brunton et al., inspired by DMD, were able to generate governing equations from data through the sparse identification of nonlinear dynamical systems [48]. However, these methods are restricted to approximating the Koopman operator, given a fixed vector-valued observable, and they have no way of effectively using measurements that vary in both number and location through time. Furthermore, the state of research into data-driven generalizing across similar systems with varying parameters is, at best, preliminary.

Learning Fluid Flows With Evolving Gaussian Processes

Second-order partial differential equations are ubiquitous in practical science and engineering, from mechanics to transport phenomena and electromagnetics. From this perspective, the Navier–Stokes equations governing fluid dynamics represent most, if not all, of the overall complexity of modeling these, as 1) they exhibit hybrid system behavior, such as elliptic hyperbolic, and 2) their nonlinearity results in the complex spatiotemporal dynamics that are prevalent in many practical situations. The evolving Gaussian process (E-GP) method is demonstrated on computational fluid dynamics (CFD) data describing a flow over a bluff body (a cylinder) across a range of Reynolds numbers from 100 to 1000 (the Reynolds number is a dimensionless flow rate). This deterministic, high-dimensional spatiotemporal dynamical system is well studied in fluid dynamics literature, both experimentally and numerically [S25]–[S27].

The conventional wisdom would dictate learning a separate model across each Reynolds number. However, results show that the E-GP method is capable of learning the dynamics of all flow patterns at once. Using the learned dynamics across weights of successive kernel models, the E-GP is capable of predicting the future states of functional evolution in a recursive manner. The key advantage of the E-GP is that the evolution of large function spaces can be transformed into learning the evolution in a relatively smaller Hilbert space encoded by the kernels and associated weight vector. The CFD simulation used a fourth-order polynomial expansion with the spectral element method on the incompressible Navier–Stokes equation to generate the cylinder flow data for Re = 100, 300, 600, 800, and 1000. 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. A Dirichlet boundary condition is applied to the inlet. Each data set contains at least 200 snapshots with a uniform time step of 0.03 s. Each snapshot contains 24,000 velocity data points for Re = 100 or 95,000 velocity data points for Re = 300, 600, 800, and 1000.

Each data set took at least 10 [illegible] a high-performance computer cluster to generate. Figures [illegible] and S7(a)–(d) visualize the horizontal velocity for Re = 100 and 1000, with red being the greatest negative velocity and blue the greatest positive velocity. The flow is unstable, periodic, and clearly nonlinear. The Gaussian radial basis function kernel $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$ was used in the E-GP model, with σ estimated to be 0.4. Using a budget of 600 kernel centers [see Figure S8(a)–(b) and note how the kernel centers cluster in the most dynamic regions], a 600 × 600 matrix $\hat{A}$ was determined, which accurately [Figure S9(a)] captured the dynamics of the nonlinear system. **<AU: Please note that Figure S9 has been renumbered as Figure S8 as figures need to**

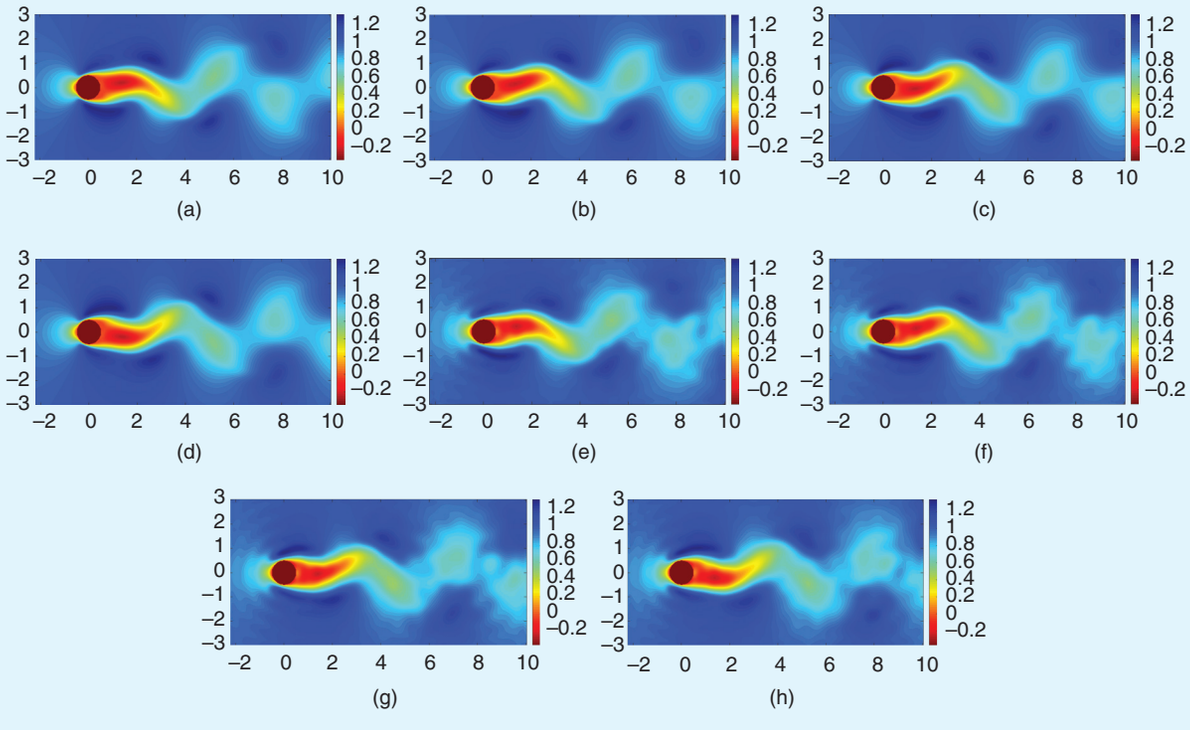

FIGURE S6 A visualization of fluid flow at Re = 100. (a)–(d): The computational fluid dynamics. (e)–(h): The evolving Gaussian process. (a) Snapshot 0. (b) Snapshot 10. (c) Snapshot 20. (d) Snapshot 30. (e) Snapshot 0. (f) Snapshot 10. (g) Snapshot 20. (h) Snapshot 30.

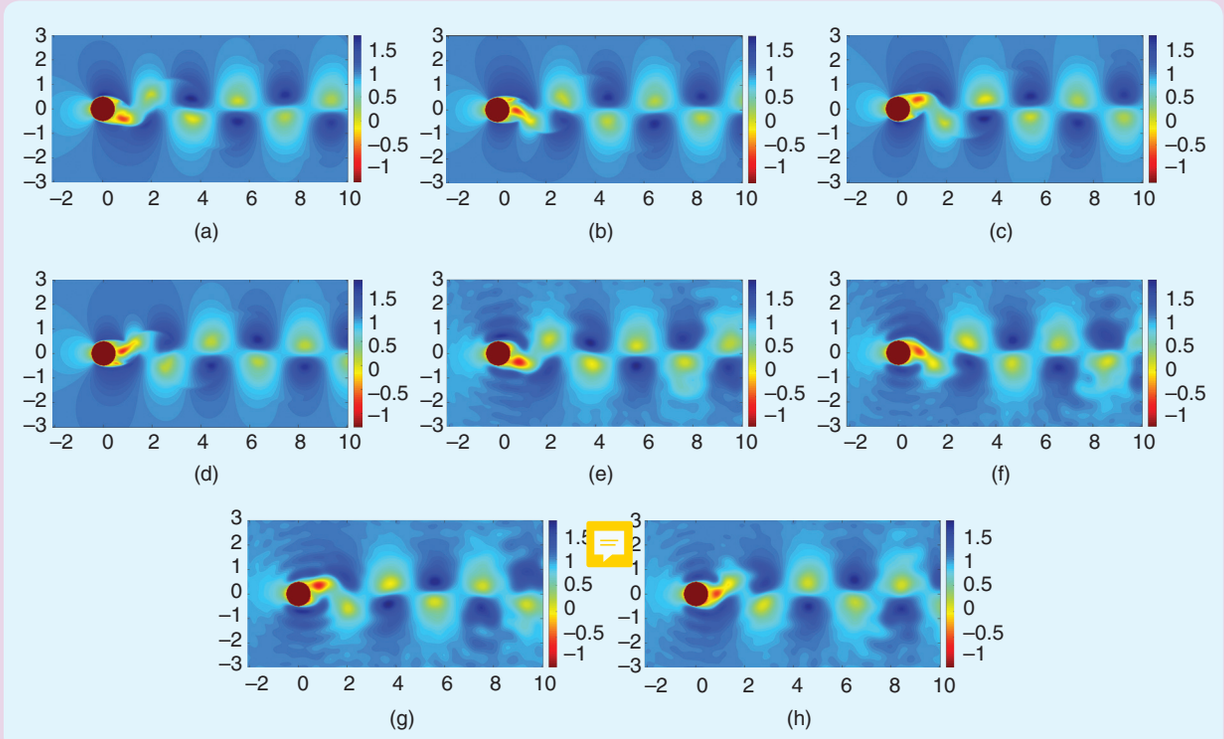

FIGURE S7 A visualization of fluid flow at Re = 1000. (a)–(d): The computational fluid dynamics. (e)–(h): The evolving Gaussian process. (a) Snapshot 0. (b) Snapshot 5. (c) Snapshot 10. (d) Snapshot 15. (e) Snapshot 0. (f) Snapshot 5. (g) Snapshot 10. (h) Snapshot 15.

appear in numerical order in the text.> This was used to propagate a single initial condition w_0 forward to make predictions and then to compare the predictions to the original training data. Total percentage errors between 3% for Re = 100 and 7–8% for Re = 1000 were found, as shown by the solid lines in Figure S9(a). The total percentage errors are defined as $E_\tau = \left(\| y_\tau - \bar{y}_\tau \|_2 / \| \bar{y}_\tau \|_2\right)$, 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 approximately 13 min in Matlab for a 200-snapshot × 95,000-point set on an ordinary Intel i7 4-GHz processor.

ONE TRANSITION MATRIX FOR EVERYTHING

To approach the challenge of generalizing across similar spatiotemporally evolving systems, the first question to answer is whether an $\hat{A}$ matrix can be found 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 five individual models trained separately on each data set. This is confirmed by both the total error plots [Figure S9(a)], which show only slight increases in each of the total percentage error plots, and a visual inspection of the displayed dynamic modes. This result is even more surprising given 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), it is determined that for the original data sets, the vortex shedding frequency is 0.448, 1.26, 1.38, 1.388, and 1.401 Hz for Re = 100, 300, 600, 800, and 1000, respectively. For the E-GP models, the frequencies are 0.452, 1.21, 1.36, 1.36, and 1.36 Hz, respectively.

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 across a range of parameters, the next challenge is to represent flows with parameters that the model has not been trained on. An $\hat{A}$ matrix was derived from the Re = 100, 300, 600, and 1000 data sets and tested against the Re = 800 data set. The results appear in Figure S9(b). For the first 120 snapshots, the total percentage error remains under 10%, which is satisfactory. After this, however, the total percentage error curves upward as the slight errors in the transition matrix compound. Across 800 snapshots, an average total percentage error of less than 25% was determined.

LINEAR DYNAMICAL LAYER ANALYSIS AND INSIGHTS

Due to the spatial encoding of the weights that the linear transition model operates on, we are able to analyze the dynamics and find physical insights into the process. Two techniques are demonstrated. The first uses eigendecomposition of the

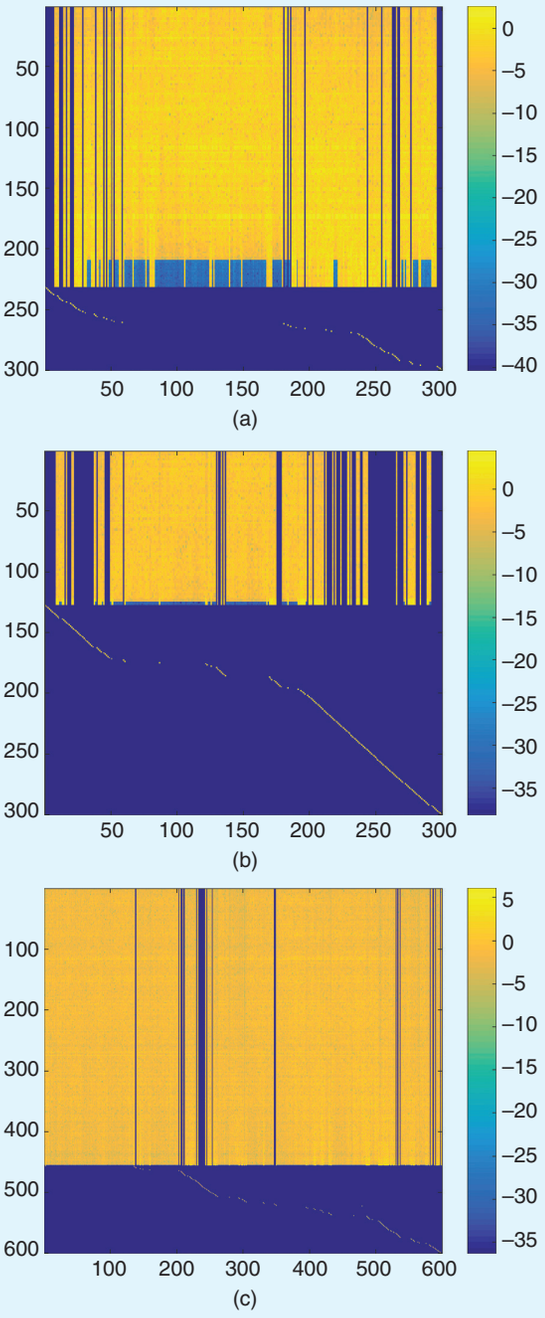

FIGURE S8 Eigenvector heat maps. (a) Re = 100, $\varepsilon = 0.005$. (b) Re = 1000, $\varepsilon = 0.05$. (c) All Reynolds numbers, $\varepsilon = 0.069$.

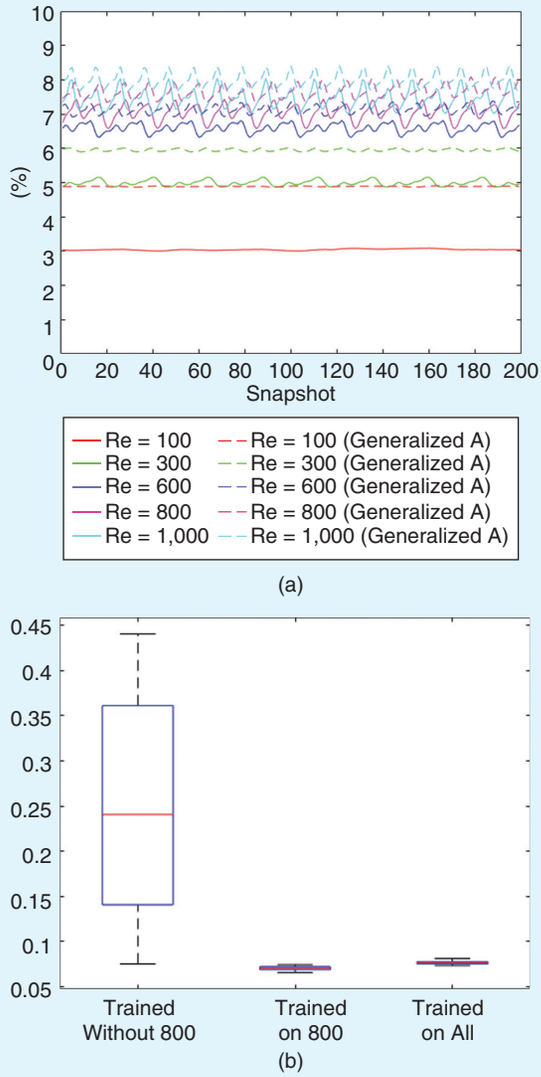

FIGURE S9 Total percentage errors. (a) The universal generalizer versus individual models. (b) Different models tested on Re = 800.

transition matrix to discover the eigenfunctions and invariant subspaces of the system. The second visualizes the most significant spatial interactions in the system. By marking which kernel centers are associated with different invariant subspaces, the area can be spatially separated into multiple dynamic modules. The physical insight is that some regions of the space are dynamically entangled with one another, and others are independent. For those interested in monitoring spatiotemporally evolving systems, the number and location of the invariant subspaces determines how many and where feedback sensors must be for robust prediction of the system state.

Before attempting a Jordan decomposition of $\hat{A}$, any elements smaller than some small ε are zeroed to stabilize the algorithm for matrices that have many elements close to zero. Afterward, the eigenvector matrix is visualized using a *logarithmic* color chart, as seen in Figure S8(a)–(c). These plots are for models individually trained on Re = 100 and 1000 with 300 kernels; in addition, all five models were trained with 600 kernels, for comparison. Three classes of eigenvector are observed in the rows, constituted by the following:

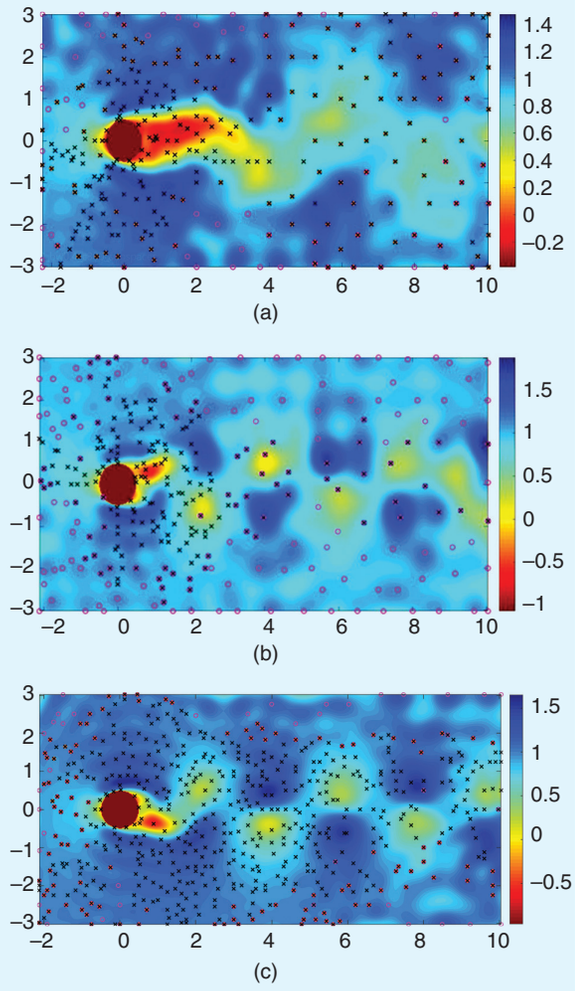

FIGURE S10 Invariant subspaces. (a) Re = 100, ε = 0.005. (b) Re = 1000, ε = 0.05. (c) All Reynolds numbers, ε = 0.069. ***<AU: Please provide high-resolution (300 dpi or better at size needed) figure files (.jpeg, .tiff, etc.).>***

1) *category 1*: rows at the bottom that have exactly one nonzero element
2) *category 2*: a few middle rows with a dozen significant elements
3) *category 3*: a number of top rows that affect the majority of the kernel centers in the space.

Each eigenvector of category 1 spans its own invariant subspace and is depicted by magenta circles in Figure S10. Category 3 is one invariant subspace, depicted by black crosses. Category 2 is subsumed in category 3. The figure shows 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 across that entire subspace. Alternately, areas far from the core of the dynamic excitement are their own independent, invariant subspaces and thus must

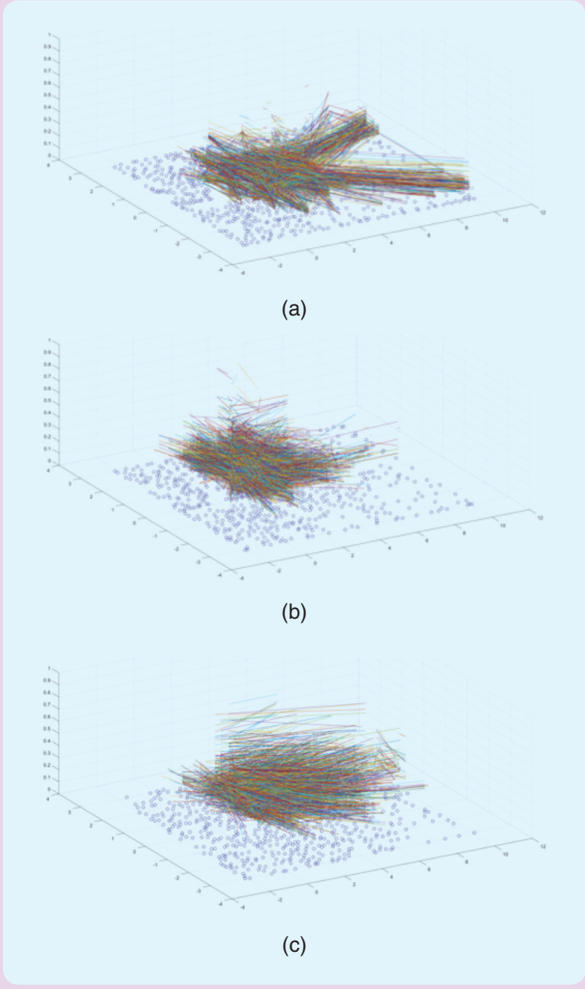

FIGURE S11 A visualization of correlations in a transition matrix. (a) Re = 100. (b) Re = 1000. (c) Trained on all five data sets. ***<AU: Please provide high-resolution (300 dpi or better at size needed) figure files (.jpeg, .tiff, etc.).>***

be locally monitored. Another way to visualize the operation of the linear transition matrix is to plot lines between kernel centers that are strongly influencing one another. That is, if a line center c_j to c_i is drawn for each of the (relatively) largest elements a_{ij} of $\hat{A}$, it can be shown how the system dynamics are coupled spatially [Figure S11(a)–(c)]. The magnitude of a_{ij} can also be plotted on a third axis for further insight into the most dominant dynamic connection in the system.

REFERENCES

[S25] A. Roshko, "On the development of turbulent wakes from vortex streets," California Inst. of Technol., Pasadena, Rep. 1191, 1954. **<AU: Please confirm the location for the associated organization.>**

[S26] M. Braza, P. H. H. M. Chassaing, and H. H. Minh, "Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder," *J. Fluid Mech.*, vol. 165, no. 1, pp. 79–130, 1986. doi: 10.1017/S0022112086003014.

[S27] B. N. Rajani, A. Kandasamy, and S. Majumdar, "Numerical simulation of laminar flow past a circular cylinder," *Appl. Math. Model.*, vol. 33, no. 3, pp. 1228–1247, 2009. doi: 10.1016/j.apm.2008.01.017.

KERNEL OBSERVERS

This section outlines the modeling framework and presents theoretical results associated with the number of sampling locations required for monitoring functional evolution.

Problem Formulation

This work focuses on the predictive inference of a time-varying stochastic process whose mean f evolves temporally via $f_{\tau+1} \sim \mathbb{F}(f_\tau, \eta_\tau)$, where $\mathbb{F}$ is a distribution varying with time τ and exogenous inputs η. The approach builds on the fact that, in several cases, temporal evolution can be hierarchically separated from spatial functional evolution. A classical and quite general example of this is the *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 where $\mathcal{L}: \mathcal{B} \to \mathcal{B}$ determines spatiotemporal transitions of $u \in \mathcal{B}$ [49]. This model of spatiotemporal evolution is very general (AEOs, for example, model many partial differential equations). However, working in Banach spaces can be computationally taxing. A simple way to make the approach computationally feasible is to place restrictions on $\mathcal{B}$: specifically, restrict the sequence f_τ to lie in an RKHS, the theory of which provides powerful tools for generating flexible classes of functions with relative ease [5].

In a kernel-based model, $k: \Omega \times \Omega \to \mathbb{R}$ is a positive definite Mercer kernel on a domain Ω that represents the covariance between any two points in the input space and 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 key insight behind the proposed model is that spatiotemporal evolution in the input domain corresponds to the temporal evolution of the mixing weights of a kernel model that is alone in the functional domain. Therefore, f_τ can be modeled by tracing the evolution of its mean embedded in an RKHS by using ODEs when the evolution is continuous or switched difference equations when it is discrete (Figure 3). The advantage of this method is that it enables the utilization of powerful ideas from systems theory for deriving necessary and sufficient conditions for spatiotemporal monitoring. This article restricts attention to the class of functional evolutions $\mathbb{F}$ defined by linear Markovian transitions in an RKHS. While extension to the nonlinear case is possible (and nontrivial), it is not pursued here so as to ease the exposition of the key ideas. The class of linear transitions in an RKHS is rich enough to approximately model many real-world data sets, as suggested by the experiments.

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 article is

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

where η_τ is a zero-mean stochastic process in $\mathcal{H}$, and ζ_τ is a Wiener process in $\mathbb{R}^N$. Classical treatments of kernel methods emphasize that, for most kernels, the feature map ψ is unknown and possibly infinite dimensional. This forces practitioners to work in the dual space of $\mathcal{H}$, whose dimensionality is the number of samples in the data set being modeled. This conventional wisdom precludes the use of kernel methods for most tasks involving modern data sets, which may have millions (and sometimes billions) of samples [50]. An alternative is to work with a feature map $\hat{\psi}(x) := [\hat{\psi}_1(x) \dots \hat{\psi}_M(x)]^T$ to an approximate feature space $\hat{\mathcal{H}}$ with the property that for every element $f \in \mathcal{H}, \exists \hat{f} \in \hat{\mathcal{H}}$, subject to $\|f - \hat{f}\| < \epsilon$, for an appropriate function norm, and $\epsilon > 0$. A few such approximations are listed in the following.

Dictionary of Atoms

Let Ω be compact. Given points $C = \{c_1, \dots, c_M\}$, $c_i \in \Omega$, this results in a dictionary of atoms (Figure 4) **<AU: Please check that the citation of Fig. 4 is appropriate.>** $\mathcal{F}^C = \{\psi(c_1), \dots, \psi(c_M)\}$, $\psi(c_i) \in \mathcal{H}$, the span of which is a strict subspace $\hat{\mathcal{H}}$ of the RKHS $\mathcal{H}$ generated by the kernel. Here,

$$\hat{\psi}_i(x) := \langle \psi(x), \psi(c_i) \rangle_{\mathcal{H}} = k(x, c_i). \tag{2}$$

Low-Rank Approximations

Let Ω be compact, let $C = \{c_1, \dots, c_M\}$, $c_i \in \Omega$, and let $K \in \mathbb{R}^{M \times M}$, $K_{ij} := k(c_i, c_j)$ be the Gram matrix computed from C. This matrix can be diagonalized to compute approximations $[\hat{\lambda}_i, \hat{\phi}_i(x)]$ of the eigenvalues and eigenfunctions $[\lambda_i, \phi_i(x)]$ of the kernel [51]. These spectral quantities can then be used to compute $\hat{\psi}_i(x) := \sqrt{\hat{\lambda}_i}\hat{\phi}_i(x)$.

Random Fourier Features

Let $\Omega \subset \mathbb{R}^n$ be compact, and let $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$ be the Gaussian radial basis function (RBF) kernel. Then, random Fourier features approximate the kernel feature map as

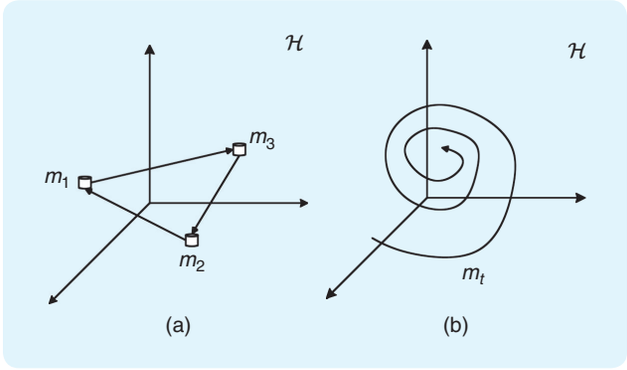

FIGURE 3 Two types of Hilbert space evolutions. (a) Discrete switches in reproducing kernel Hilbert space $\mathcal{H}$. (b) Smooth evolution in $\mathcal{H}$.

$\hat{\psi}_\omega : \Omega \to \hat{\mathcal{H}}$, where ω is a sample from the Fourier transform of $k(x,y)$, with the property that $k(x,y) = \mathbb{E}_\omega[\langle \hat{\psi}_\omega(x), \hat{\psi}_\omega(y)\rangle_{\hat{\mathcal{H}}}]$ [50]. In this case, if $V \in \mathbb{R}^{M/2\times n}$ is a random matrix representing the sample ω, then $\hat{\psi}_i(x) := [(1/\sqrt{M})\sin([Vx]_i), (1/\sqrt{M})\cos([Vx]_i)]$. Similar approximations exist for other radially symmetric kernels as well as dot product kernels. In the approximate space case, replace the transition operator $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ in (1) with $\hat{\mathcal{A}} : \hat{\mathcal{H}} \to \hat{\mathcal{H}}$. This approximate regime, which combines the flexibility of a truly nonparametric approach with computational realizability, still enables the representation of rich phenomena (as will be seen in the sequel and in Figure 5).

The finite-dimensional evolution equations approximating (1) in dual form (Figure 6) **<AU: Please check that the citation of Figure 6 is appropriate.>** are

$$w_{\tau+1} = \hat{A}w_\tau + \eta_\tau, \qquad y_\tau = Kw_\tau + \zeta_\tau, \tag{3}$$

where matrices $\hat{A} \in \mathbb{R}^{M\times M}$; $K \in \mathbb{R}^{N\times M}$; the vectors $w_\tau \in \mathbb{R}^M$; slightly abused notation enables y_τ, η_τ, and ζ_τ denote their

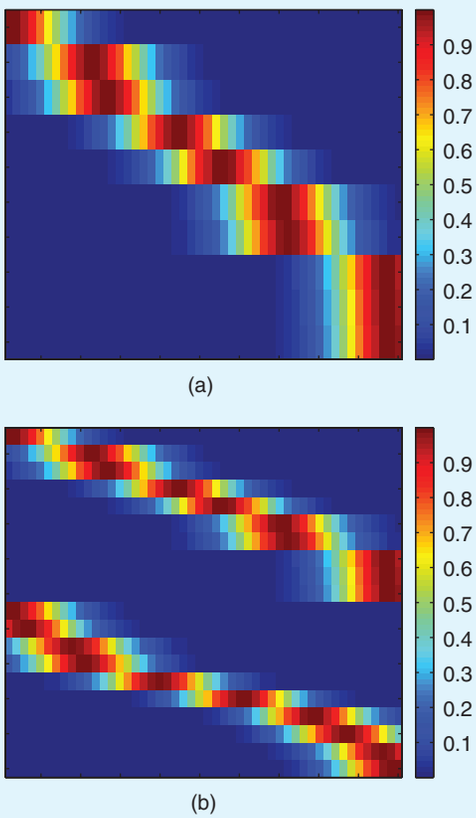

FIGURE 4 Shaded observation matrices for a dictionary of atoms. Each row represents a sensing location, with the color map indicating the evaluation of the kernel function with respect to the others points in the domain. (a) One shaded (Definition 1). (b) Two shaded [(8)].

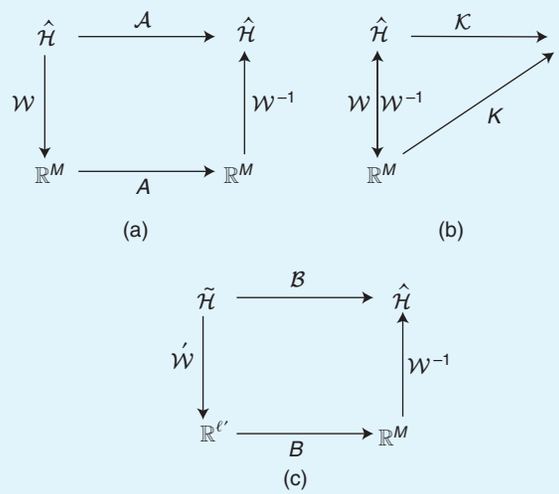

FIGURE 6 Commutative diagrams between primal and dual spaces. (a) The relationship between $\mathcal{A}$ and A. (b) The relationship between $\mathcal{K}$ and K. (c) The relationship between $\mathcal{B}$ and B.

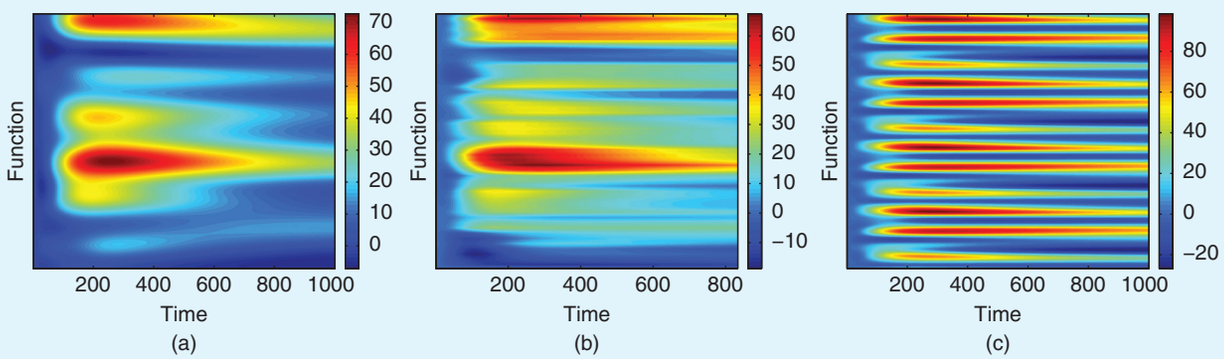

FIGURE 5 The 1D function evolution across a fixed transition matrix A, initial condition w_0, and centers C [but with different kernels $k(x, y)$]. Each y-vector at a given value of x represents the output of the function, which evolves from left to right. As seen, changing the kernel creates very different dynamic behaviors. (a) Gaussian. (b) Laplacian. (c) Periodic.

$\hat{\mathcal{H}}$ counterparts. Here, K is the matrix whose rows are of the form $K_{(i)} = \hat{\Psi}(x_i) = [\hat{\psi}_1(x_i) \ \hat{\psi}_2(x_i) \ \cdots \ \hat{\psi}_M(x_i)]$. In systems theoretic language, each row of K corresponds to a *measurement* at a specific location, and the matrix itself acts as a measurement operator. The equations in (1) suggest an immediate extension to functional control problems. Select another basis for $\mathcal{H}$ as $\tilde{\psi}(x) := [\tilde{\psi}_1(x) \ \cdots \ \tilde{\psi}_{\ell'}(x)]^T$, where the functions $\tilde{\psi}_j(x)$ are used to approximate the RKHS $\mathcal{H}$ generated by the kernel. Denote the span of these functions as $\tilde{\mathcal{H}}$. In the dictionary of atoms case, an example would be another set of atoms $\mathcal{F}_D = [\psi(d_1) \ \cdots \ \psi(d_{\ell'})]$, $\psi(d_j) \in \mathcal{H}, d_j \in \Omega$, with $\tilde{\mathcal{H}}$ being a strict subspace of the RKHS $\mathcal{H}$ generated by the kernel. The functional evolution equation is then

$$f_{\tau+1} = \mathcal{A} f_\tau + \mathcal{B}\delta_\tau + \eta_\tau, \quad y_\tau = \mathcal{K}_\tau f_\tau + \zeta_\tau, \tag{4}$$

where the control functions δ_τ evolve in $\tilde{\mathcal{H}}$ and $\mathcal{B} : \tilde{\mathcal{H}} \to \hat{\mathcal{H}}$.

To derive the finite-dimensional equivalent of $\mathcal{B}$, the structure of the matrix B must be determined. Since $\hat{\mathcal{H}}$ is not, in general, isomorphic to $\tilde{\mathcal{H}}$, this imposes strict restrictions on B. Derive B using least squares, employing the inner product on $\mathcal{H}$. An instructive example is where both $\hat{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ are generated by dictionaries of atoms. Recall that, in this case, $\mathcal{F}^C = [\psi(c_1) \ \cdots \ \psi(c_M)]$ is the basis for $\hat{\mathcal{H}}$, $\delta = \Sigma_{j=1}^{\ell'} \acute{w}_j \psi(d_j)$, and $\mathcal{F}^C = [\psi(c_1) \ \cdots \ \psi(c_M)]$ is the basis for $\mathcal{H}^C$. The projection of δ onto $\hat{\mathcal{H}}$ can be derived as

$$\begin{bmatrix} \langle \delta, \psi(c_1) \rangle_{\mathcal{H}} \\ \vdots \\ \langle \delta, \psi(c_M) \rangle_{\mathcal{H}} \end{bmatrix} = \underbrace{\begin{bmatrix} \langle \psi(d_1), \psi(c_1) \rangle_{\mathcal{H}} & \cdots & \langle \psi(d_{\ell'}), \psi(c_1) \rangle_{\mathcal{H}} \\ \vdots & \ddots & \vdots \\ \langle \psi(d_1), \psi(c_M) \rangle_{\mathcal{H}} & \cdots & \langle \psi(d_{\ell'}), \psi(c_M) \rangle_{\mathcal{H}} \end{bmatrix}}_{K_{CD}} \begin{bmatrix} \acute{w}_1 \\ \vdots \\ \acute{w}_{\ell'} \end{bmatrix}. \tag{5}$$

Note that in the dictionary of atoms case, the entries of K_{CD} can be computed in closed form as $K_{CDij} := k(d_i, c_j)$ using the reproducing property. This derivation shows that the operator B is simply $K_{CD} \in \mathbb{R}^{M \times \ell'}$, the kernel matrix between the data C generating the atoms $\mathcal{F}^C$ of $\hat{\mathcal{H}}$ and the data D generating the atoms $\mathcal{F}_D$ of $\tilde{\mathcal{H}}$. Thus, the finite-dimensional evolution equations equivalent to (4) are

$$w_{\tau+1} = \hat{A} w_\tau + K_{CD} \acute{w}_\tau, \quad y_\tau = K_\tau w_\tau. \tag{6}$$

Define the *generalized observability matrix* [52] as

$$\mathcal{O}_\Upsilon = \begin{bmatrix} K\hat{A}^{\tau_1} \\ \cdots \\ K\hat{A}^{\tau_L} \end{bmatrix},$$

where $\Upsilon = \{\tau_1, \ldots, \tau_L\}$ are the set of instances τ_i when applying the operator K. A linear system is said to be *observable* if $\mathcal{O}_\Upsilon$ has full column rank [that is, $\text{Rank}(\mathcal{O}_\Upsilon) = M$] for $\Upsilon = \{0, 1, \ldots, M-1\}$ [52]. Observability guarantees two critical facts. First, the state w_0 can be exactly recovered from a finite series of measurements $\{y_{\tau_1}, y_{\tau_2}, \ldots, y_{\tau_L}\}$. Specifically, defining $y_\Upsilon = [y_{\tau_1}^T, y_{\tau_2}^T, \ldots, y_{\tau_L}^T]^T$, $y_\Upsilon = \mathcal{O}_\Upsilon w_0$. Second, it guarantees that a feedback-based *observer* can be designed such that the estimate of w_τ, denoted by $\hat{w}_\tau$, converges exponentially fast to w_τ in the limit of the samples. Note that all theoretical results assume $\hat{A}$ is available. While system identification is performed in the experiments [21], it is not the focus of this article.

We are now in a position to formally state the spatiotemporal modeling, control, and inference problems being considered. Given a spatiotemporally evolving system modeled using (3), 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*), robustly predicted (which corresponds to *Bayesian filtering*), and controlled (which corresponds to *functional control*). This approach to solve the monitoring and prediction problem relies on the design of the measurement operator K so that the pair $(K, \hat{A})$ is observable: any Bayesian state estimator (for example, a Kalman filter) utilizing this pair is denoted as a *kernel observer* (Figure 7). **<AU: Please check that the citation of Figure 7 is appropriate.>** In the case where no measurements are taken, for the sake of consistency, denote the state estimator as an *autonomous* kernel observer. In the controls case, given a spatiotemporally evolving system modeled using (6), a set of N sensing locations and ℓ' control locations must be chosen such that, even with $N \ll M, \ell' \ll M$, the functional evolution of the spatiotemporal model can be controlled. In this case, both a measurement operator K and a control operator K_{CD} must be designed such that the pair $(K_{CD}, \hat{A})$ is controllable. A controls system utilizing this pair and the measurement operator K is denoted as a *kernel controller*.

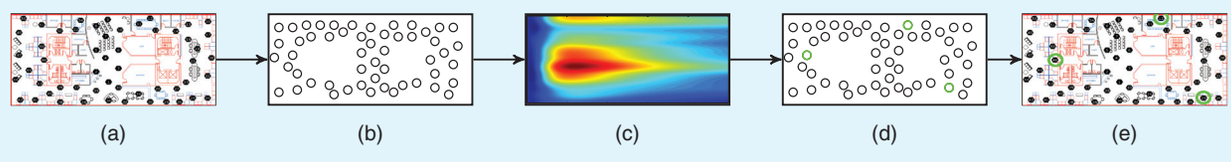

FIGURE 7 A description of how the kernel observer fits in the sensing framework. Physical locations are mapped to data locations across which historical information is collected as a time series. Functional inference is performed across $\hat{\mathcal{H}}$ to solve for $\hat{A}$. The measurement operator K is then computed (see Figure 8), leading to the sensor placement. (a) Physical sampling locations. (b) Data locations. (c) The functional inference (for $\hat{A}$). (d) ***<AU: The caption information for Figure 7(d) is cut off and illegible. Please provide the caption information.>*** (e) Physical sensor placement.

Preliminaries of Rational Canonical Structures

Using a geometric approach to the choice of sampling locations for inferring w_τ in (3), the extension to control is similar. Use the notation $\mathcal{V}$, with $\dim(\mathcal{V}) = M$, to emphasize the fact that these theorems hold for any finite-dimensional vector space. Consider the linear operator $\mathcal{A}: \mathcal{V} \to \mathcal{V}$, and recall that the definition of observability requires the construction of a linear operator $\mathcal{K}: \mathcal{V} \to \mathcal{U}$, with $\dim(\mathcal{U}) = N$, such that $\operatorname{rank}\left[(\mathcal{K})^T \ \cdots \ (\mathcal{K}\mathcal{A}^{M-1})^T\right]^T = M$. In most applications, if $N \geq M$ and $\operatorname{rank}(\mathcal{K}) = N$, it is reasonable to expect that observability may be achieved. However, for this purpose, N must be *significantly* less than M. Therefore, $\mathcal{K}$ must be designed with as small a rank as possible. To do so, we require a series of vectors v_i that, under repeated iterations of $\mathcal{A}$, can generate a basis for $\mathcal{V}$. For this task, a fundamental decomposition result from the theory of modules (known as the *rational canonical structure*) of $\mathcal{A}$ [53] is used.

The intuition here is that if the sequence $\{v_i\}_i$ can generate this basis, it can be directly used to construct $\mathcal{K}$. The linear operator $\mathcal{A}: \mathcal{V} \to \mathcal{V}$ has a characteristic polynomial $\pi(\lambda)$ such that $\pi(\mathcal{A}) = 0$ by the Cayley–Hamilton theorem. The minimal polynomial (MP) of $\mathcal{A}$ is the monic polynomial $\alpha(\cdot)$ of the least degree [denoted by $\deg(\cdot)$], given as $\alpha(\lambda) = a_0 + a_1\lambda + \cdots + \lambda^{\deg(\alpha)} = 0$ such that $\alpha(\mathcal{A}) = a_0 I + a_1\mathcal{A} + \cdots + \mathcal{A}^{\deg(\alpha)} = 0$. The MP is unique and divides $\pi(\lambda)$ so that $\deg(\alpha) \leq \deg(\pi)$. The MP of a vector $v \in \mathcal{V}$ *relative to* $\mathcal{A}$ is the unique monic polynomial ξ_v of the least degree such that $\xi_v(\mathcal{A})v = a_0 v + a_1\mathcal{A}v + \cdots + \mathcal{A}^{\deg(\alpha)}v = 0$. If $\deg(\alpha) = M$, then $\mathcal{A}$ is *cyclic*, and $\exists v \in \mathcal{V}$ such that the vectors $\{v, \mathcal{A}v, \ldots, \mathcal{A}^{M-1}v\}$ form a basis for $\mathcal{V}$. This is the same as saying that the pair $(v^T, \mathcal{A}^T)$ is observable. A subspace $\mathcal{V}_\mathcal{S} \subset \mathcal{V}$ subject to $\mathcal{A}\mathcal{V}_\mathcal{S} \subset \mathcal{V}_\mathcal{S}$ is $\mathcal{A}$-cyclic if $\mathcal{A}_{|\mathcal{V}_\mathcal{S}}$, the restriction of $\mathcal{A}$ to the subspace $\mathcal{V}_\mathcal{S}$, is cyclic. If $\alpha(\lambda)$ is the MP of $\mathcal{A}$ and $\deg(\alpha) = m < M$, $\exists\, v \in \mathcal{V}$ such that $\{v, \mathcal{A}v, \ldots, \mathcal{A}^{m-1}v\}$ span an m-dimensional $\mathcal{A}$-cyclic subspace $\mathcal{V}_\mathcal{S}$, with v being the *cyclic generator* of $\mathcal{V}_\mathcal{S}$. The subspace $\mathcal{V}_\mathcal{S}$ decomposes $\mathcal{V}$ relative to $\mathcal{A}$. By the rational canonical structure theorem [53, Th. 0.1], $\mathcal{A}$ can be successively decomposed into subspaces $\mathcal{V}_i \subset \mathcal{V}, i \in \{1, \ldots, \ell\}$, subject to $\mathcal{V} = \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_\ell$, $\mathcal{A}\mathcal{V}_i \subset \mathcal{V}_i$, and $\mathcal{A}_{|\mathcal{V}_i}, i \in \{1, \ldots, \ell\}$ are cyclic.

In general, the subspaces $\mathcal{V}_i$ are not unique for a fixed $\mathcal{A}$. The integer ℓ is unique and called the *cyclic index of* $\mathcal{A}$. One of our main results shows that the cyclic index is a lower bound on the number of measurements required to reconstruct w_τ (see Proposition 3 and Algorithm 1). The matrix transform associated with this theorem is known as the *Frobenius normal form* (denoted by $C \in \mathbb{R}^{M\times M}$): for $\mathcal{A} \in \mathbb{R}^{M\times M}$, $\exists Q \in \mathbb{R}^{M\times M}$ is invertible such that $\mathcal{A} = QCQ^{-1}$. We will also use *Jordan decomposition*, where for $\mathcal{A} \in \mathbb{R}^{M\times M}$, $\exists P \in \mathbb{R}^{M\times M}$ is invertible such that $\mathcal{A} = P\Lambda P^{-1}$, where Λ is a unique block diagonal matrix with Jordan blocks with λ_i along the diagonal. If all the eigenvalues λ_i are nonzero and real, we say the matrix has a *full-rank Jordan decomposition*.

ALGORITHM 1 Measurement map $\tilde{K}$.

Input: $\hat{A} \in \mathbb{R}^{M\times M}$

Compute Rational Canonical Form, subject to $C = Q^{-1}\hat{A}^T Q$.

Set $C_0 := C$ and $M_0 := M$.

for $i = 1$ **to** ℓ **do**

Obtain (minimum polynomial) $\alpha_i(\lambda)$ of C_{i-1}. This returns associated indices $\mathcal{J}^{(i)} \subset \{1, 2, \ldots, M_{i-1}\}$.

Construct vector $v_i \in \mathbb{R}^M$ such that $\xi_{v_i}(\lambda) = \alpha_i(\lambda)$.

Use indices $\{1, 2, \ldots, M_{i-1}\}\backslash\mathcal{J}^{(i)}$ to select matrix C_i. Set $M_i := \left|\{1, 2, \ldots, M_{i-1}\}\backslash\mathcal{J}^{(i)}\right|$

end for

Compute $\mathring{K} = [v_1^T, v_2^T, \ldots, v_\ell^T]^T$

Output: $\tilde{K} = \mathring{K}Q^{-1}$

MAIN RESULTS

This section proves the results concerning the observability of spatiotemporally varying functions modeled by the functional evolution and measurement (3). Specifically, observability of the system states implies that the current state of the spatiotemporally varying function can be recovered using a small number of sampling locations N, which enables 1) tracking the function and 2) predicting the function's evolution forward in time. Using the approximation $\hat{\mathcal{H}} \approx \mathcal{H}$, given M basis functions, implies that the dual space of $\hat{\mathcal{H}}$ is $\mathbb{R}^M$. Proposition 1 shows that if $\hat{A}$ has a full-rank Jordan decomposition, the observation matrix K that meets a condition called *shadedness* (Definition 1) is sufficient for the system to be observable. Proposition 2 provides a lower bound on the number of sampling locations required for observability, which holds for any $\hat{A}$. Proposition 3 constructively shows the existence of an abstract measurement map $\tilde{K}$ achieving this lower bound. Since the measurement map does not have the structure of a kernel matrix, a slightly weaker sufficient condition for the observability of any $\hat{A}$ is in Theorem 1. Finally, since both K and K_{CD} are kernel matrices generated from a shared kernel, these observability results translate directly into controllability results.

Definition 1: Shaded Observation Matrix

Given that $k: \Omega \times \Omega \to \mathbb{R}$ is positive definite on a domain Ω, let $\{\hat{\psi}_1(x), \ldots, \hat{\psi}_M(x)\}$ be the set of bases generating an approximate feature map $\hat{\psi}: \Omega \to \hat{\mathcal{H}}$, and let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be the set of sampling (or sensing) locations, with each $x_i \in \Omega$. Let $K \in \mathbb{R}^{N\times M}$ being the observation matrix, where $K_{ij} := \hat{\psi}_j(x_i)$. For each row $K_{(i)} := [\hat{\psi}_1(x_i) \ \cdots \ \hat{\psi}_M(x_i)]$, define the set $\mathcal{I}_{(i)} := \{\iota_1^{(i)}, \iota_2^{(i)}, \ldots, \iota_{M_i}^{(i)}\}$ to be the indices in the observation matrix row i that are nonzero. Then, if $\cup_{i\in\{1,\ldots,N\}} \mathcal{I}^{(i)} = \{1, 2, \ldots, M\}$, denote K as a *shaded observation matrix* [see Figure 4(a)]. This definition seems quite abstract. Hence, the following remark considers a more concrete example.

Remark 1

Let $\hat{\psi}$ be generated by the dictionary given by $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$. Note that since $\hat{\psi}_j(x_i) = \langle \psi(x_i), \psi(c_j) \rangle_{\mathcal{H}} = k(x_i, c_j)$, K is the kernel matrix between $\mathcal{X}$ and $\mathcal{C}$. For the kernel matrix to be shaded, the implication is that there does not exist an atom $\psi(c_j)$ such that the projections $\langle \psi(x_i), \psi(c_j) \rangle_{\mathcal{H}}$ vanish for all x_i, $1 \leq i \leq N$. Intuitively, the shadedness property requires that the sensor locations x_i are privy to information propagating from every c_j. As an example, note that (in principle) for the Gaussian kernel, a single row generates a shaded kernel matrix. However, in this case, the matrix can have many entries that are extremely close to zero and will probably be very ill conditioned.

Proposition 1

Given that $k : \Omega \times \Omega \to \mathbb{R}$ is positive definite on a domain Ω, let $\{\hat{\psi}_1(x), \ldots, \hat{\psi}_M(x)\}$ be the set of bases generating an approximate feature map $\hat{\psi} : \Omega \to \hat{\mathcal{H}}$, and let $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \Omega$. Consider the discrete linear system on $\widehat{\mathcal{H}}$ given by the evolution and measurement (3). Suppose that a full-rank Jordan decomposition of $\hat{A} \in \mathbb{R}^{M \times M}$ of the form $\hat{A} = P \Lambda P^{-1}$ exists, where $\Lambda = [\Lambda_1 \ \cdots \ \Lambda_O]$ and there are no repeated eigenvalues. Then, given a set of time instances $\Upsilon = \{\tau_1, \tau_2, \ldots, \tau_L\}$ and a set of sampling locations $\mathcal{X} = \{x_1, \ldots, x_N\}$, the system (3) is observable if the observation matrix K_{ij} is shaded according to Definition 1, Υ has distinct values, and $|\Upsilon| \geq M$.

Proof

Consider a system where $\hat{A} = \Lambda$, with Jordan blocks $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_O\}$ along the diagonal. Then, $\hat{A}^{\tau_i} = \operatorname{diag}\left([\Lambda_1^{\tau_i} \ \Lambda_2^{\tau_i} \ \cdots \ \Lambda_O^{\tau_i}]\right)$. This results in

$$\mathcal{O}_\Upsilon = \underbrace{\begin{bmatrix} K\hat{A}^{\tau_1} \\ \cdots \\ K\hat{A}^{\tau_L} \end{bmatrix}}_{\mathcal{O}_\Upsilon \in \mathbb{R}^{NL \times M}}.$$

It must be proved that the column rank of $\mathcal{O}_\Upsilon$ is M, which is not immediately obvious since typically $N \ll M$. To prove the statement, it is shown that computing the rank of $\mathcal{O}_\Upsilon$ is equivalent to the rank computation of the product of two simple matrices. In the following, use the notation $\mathbf{0}_{\mathbb{R}^{I \times J}}$ to denote an $I \times J$ matrix of all zeros.

In the first step, write the preceding matrix as the product of two matrices. It can be shown that $\mathcal{O}_\Upsilon$ is the product of two block matrices,

$$\mathcal{O}_\Upsilon = \underbrace{\begin{bmatrix} K & \cdots & \mathbf{0}_{\mathbb{R}^{N \times M}} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{\mathbb{R}^{N \times M}} & \cdots & K \end{bmatrix}}_{\hat{\mathbf{K}} \in \mathbb{R}^{NL \times ML}} \underbrace{\left[\begin{array}{ccc} \Lambda_1^{\tau_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Lambda_O^{\tau_1} \\ \hline \vdots & \ddots & \vdots \\ \hline \Lambda_1^{\tau_L} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Lambda_O^{\tau_L} \end{array}\right]}_{\hat{\mathbf{A}} \in \mathbb{R}^{ML \times M}}.$$

Note that $\hat{\mathbf{K}}$ must be simplified even further. Recall that a matrix's rank is preserved under a product with an invertible matrix. Design a matrix of elementary row operations $U \in \mathbb{R}^{N \times N}$ such that $\check{K} := UK$ is a matrix with at least one row vector of nonzeros; this can be achieved by having an elementary matrix that adds rows together. By the shadedness assumption, such a matrix exists. This operation can be written as

$$UK = \begin{bmatrix} \check{K}_{11} & \check{K}_{12} & \cdots & \check{K}_{1M} \\ \check{K}_{21} & \check{K}_{22} & \cdots & \check{K}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \check{K}_{N1} & \check{K}_{N2} & \cdots & \check{K}_{NM}. \end{bmatrix}.$$

Without a loss of generality (and slightly abusing the notation), let this multiplication lead to one nonzero row (with the rest of the elements of the matrix being zero) as

$$UK = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1M} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Since elementary matrices are full rank, $\operatorname{rank}(UK) = \operatorname{rank}(K)$.

To analyze the rank of $\mathcal{O}_\Upsilon$, apply these elementary matrices to every $K \in \hat{\mathbf{K}}$. Consider the block diagonal matrix $\mathcal{U} \in \mathbb{R}^{NL \times NL}$, with $U \in \mathbb{R}^{N \times N}$ along the diagonal and zeros everywhere else. That is,

$$\mathcal{U} := \begin{bmatrix} U & \mathbf{0}_{\mathbb{R}^{N \times N}} & \cdots & \mathbf{0}_{\mathbb{R}^{N \times N}} \\ \mathbf{0}_{\mathbb{R}^{N \times N}} & U & \cdots & \mathbf{0}_{\mathbb{R}^{N \times N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{\mathbb{R}^{N \times N}} & \mathbf{0}_{\mathbb{R}^{N \times N}} & \cdots & U \end{bmatrix}. \tag{7}$$

It can be shown that $\mathcal{U}$ is full rank; in other words, it has rank NL. Referring back to the observability matrix,

$$\mathcal{U}\mathcal{O}_\Upsilon = \mathcal{U}\hat{\mathbf{K}}\hat{\mathbf{A}} = \underbrace{\begin{bmatrix} UK & \mathbf{0}_{\mathbb{R}^{N \times M}} & \cdots & \mathbf{0}_{\mathbb{R}^{N \times M}} \\ \mathbf{0}_{\mathbb{R}^{N \times M}} & UK & \cdots & \mathbf{0}_{\mathbb{R}^{N \times M}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{\mathbb{R}^{N \times M}} & \mathbf{0}_{\mathbb{R}^{N \times M}} & \cdots & UK \end{bmatrix}}_{\mathcal{U}\hat{\mathbf{K}} \in \mathbb{R}^{NL \times ML}} \underbrace{\hat{\mathbf{A}}}_{\in \mathbb{R}^{ML \times M}},$$

since $\mathbf{0}_{\mathbb{R}^{N \times N}} \mathbf{0}_{\mathbb{R}^{N \times M}} = \mathbf{0}_{\mathbb{R}^{N \times M}}$. Due to the fact that $\operatorname{rank}(\mathcal{U}\mathcal{O}_\Upsilon) = \operatorname{rank}(\mathcal{O}_\Upsilon)$, the rank analysis can therefore be performed on the simpler matrix $\operatorname{rank}(\mathcal{U}\mathcal{O}_\Upsilon)$. Note that

$$UK\hat{A}^{\tau_j} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1M} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \hat{A}^{\tau_j}$$

$$= \begin{bmatrix} k_{11}\lambda_1^{\tau_j} & \binom{\tau_j}{1}\lambda_1^{\tau_j - 1} + k_{12}\lambda_1^{\tau_j} & \cdots & k_{1M}\lambda_O^{\tau_j} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Thus, following some more elementary row operations encoded by $V \in \mathbb{R}^{ML \times ML}$,

$$VU\mathcal{O}_\Upsilon = \begin{bmatrix} k_{11}\lambda_1^{\tau_1} & \cdots & k_{1M}\lambda_O^{\tau_1} \\ k_{11}\lambda_1^{\tau_2} & \cdots & k_{1M}\lambda_O^{\tau_2} \\ \vdots & \ddots & 0 \\ k_{11}\lambda_1^{\tau_L} & \cdots & k_{1M}\lambda_O^{\tau_L} \\ \mathbf{0}_{\mathbb{R}^{M(L-1)\times 1}} & \cdots & \mathbf{0}_{\mathbb{R}^{M(L-1)\times 1}} \end{bmatrix} = \begin{bmatrix} \Phi \\ \mathbf{0}_{\mathbb{R}^{M(L-1)\times M}} \end{bmatrix}.$$

If the individual entries k_{1i} are nonzero and the Jordan block diagonals have nonzero eigenvalues, the columns of Φ become linearly independent. Therefore, if $L \geq M$, the column rank of $\mathcal{O}_\Upsilon$ is M, which results in an observable system. To extend this proof to matrices $\hat{A} = P\Lambda P^{-1}$, note that

$$\begin{aligned} \mathcal{O}_\Upsilon &= \begin{bmatrix} K\hat{A}^{\tau_1} \\ \cdots \\ K\hat{A}^{\tau_L} \end{bmatrix} \\ &= \begin{bmatrix} KP\Lambda^{\tau_1}P^{-1} \\ \cdots \\ KP\Lambda^{\tau_L}P^{-1}. \end{bmatrix} \\ &= \hat{\mathbf{K}}\boldsymbol{P}\Lambda^t\boldsymbol{P}^{-1}, \end{aligned}$$

where $\boldsymbol{P} \in \mathbb{R}^{ML \times ML}$, $\Lambda^t \in \mathbb{R}^{ML \times ML}$, and $\boldsymbol{P}^{-1} \in \mathbb{R}^{ML \times ML}$ are the block diagonal matrices associated with the system. Since $\boldsymbol{P}$ is an invertible matrix, the conclusions about the column rank drawn before still hold, and the system is observable. ■

When the eigenvalues of the system matrix are repeated, it is not enough for K to be shaded. The next proposition takes a geometric approach and utilizes the rational canonical form of $\hat{A}$ to obtain a lower bound on the number of required sampling locations. Let r be the number of unique eigenvalues of $\hat{A}$, and let γ_{λ_i} denote the geometric multiplicity of eigenvalue λ_i. Then, the *cyclic index* of $\hat{A}$ is defined as $\ell = \max_{1 \leq i \leq r} \gamma_{\lambda_i}$ [53].

Proposition 2

Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan blocks $[\Lambda_1 \ \cdots \ \Lambda_O]$ may have repeated eigenvalues (that is, $\exists \Lambda_i$ and Λ_j, subject to $\lambda_i = \lambda_j$). Then, there exist kernels $k(x, y)$ such that the lower bound ℓ on the number of sampling locations N is given by the cyclic index of $\hat{A}$. In other words, the system in (3) is observable if $N \geq \ell$.

Proof by Contrapositive

It is shown that if the number of sampling locations is $N = \ell - 1$ (that is, $N < \ell$), then the system is not observable. Select the Gaussian kernel in the dictionary of atoms framework with sampling locations $x_i \in \mathcal{X}$ and that centers $c_j \in \mathcal{C}$, with the additional property that $x_i \neq x_j \, \forall i, j \in \{1, \ldots, N\}$, $i \neq j$. In this case, $\mathbf{K}$ has $\ell - 1$ nonzero, linearly independent rows, and it can be written as

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1M} \\ \vdots & \vdots & \cdots & \vdots \\ k_{(\ell-1)1} & k_{(\ell-1)2} & \cdots & k_{(\ell-1)M} \end{bmatrix}.$$

The fact that the cyclic index is ℓ implies that at least one eigenvalue, λ, has ℓ Jordan blocks. Define indices $j_1, j_2, \ldots, j_\ell \in \{1, 2, \ldots, M\}$ as the columns corresponding to the leading entries of the ℓ Jordan blocks corresponding to λ. Without a loss of generality, let $j_1 = 1$. Using ideas similar to the previous proof, we can write the observability matrix as

$$\mathcal{O}_\Upsilon := \begin{bmatrix} k_{11}\lambda^{\tau_1} & \cdots & k_{1j_\ell}\lambda^{\tau_1} & \cdots \\ \vdots & \ddots & \vdots & \ddots \\ k_{11}\lambda^{\tau_L} & k_{1j_\ell}\lambda^{\tau_L} & \cdots & \\ \vdots & \ddots & \vdots & \ddots \\ k_{(\ell-1)1}\lambda^{\tau_1} \cdots & k_{(\ell-1)j_\ell}\lambda^{\tau_1} & \cdots & \\ \vdots & \ddots & \vdots & \ddots \\ k_{(\ell-1)1}\lambda^{\tau_L} & \cdots & k_{(\ell-1)j_\ell}\lambda^{\tau_L} & \cdots \end{bmatrix}.$$

Define $\boldsymbol{\lambda} := [\lambda^{\tau_1} \ \lambda^{\tau_2} \ \cdots \lambda^{\tau_L}]^T$. The preceding matrix becomes

$$\mathcal{O}_\Upsilon := \begin{bmatrix} k_{11}\boldsymbol{\lambda} & \cdots & k_{1j_2}\boldsymbol{\lambda} & \cdots & k_{1j_\ell}\boldsymbol{\lambda} & \cdots \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \\ k_{(\ell-1)1}\boldsymbol{\lambda} & \cdots & k_{(\ell-1)j_2}\boldsymbol{\lambda} & \cdots & k_{(\ell-1)j_\ell}\boldsymbol{\lambda} & \cdots \end{bmatrix}.$$

It must be shown that one of the columns can be written in terms of the others. This is equivalent to solving the linear system

$$\begin{bmatrix} k_{1j_1} \\ k_{2j_1} \\ \vdots \\ k_{(\ell-1)j_1} \end{bmatrix} = \begin{bmatrix} k_{1j_2} & \cdots & k_{1j_\ell} \\ k_{2j_2} & \cdots & k_{2j_\ell} \\ \vdots & \ddots & \vdots \\ k_{(\ell-1)j_2} & \cdots & k_{(\ell-1)j_\ell} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{(\ell-1)} \end{bmatrix}.$$

Since the kernel matrix on the right-hand side is generated from the Gaussian kernel from [54], it is known that every principal minor of a Gaussian kernel matrix is invertible, which implies that O_Υ cannot be observable. ■

A concrete example will be given to build intuition regarding this lower bound. For now, note the following proposition.

Proposition 3

Given the conditions stated in Proposition 2, it is possible to construct a measurement map $\tilde{K} \in \mathbb{R}^{\ell \times M}$ for the system given by (3) such that the pair $(\tilde{K}, \hat{A})$ is observable.

Proof

The construction of the measurement map $\tilde{K}$ is based on the rational canonical structure of $\hat{A}^T$, which decomposes $\mathcal{V}$ into $\hat{A}^T$-cyclic direct summands such that $\mathcal{V} = \mathcal{V}_1 \oplus \cdots \oplus \mathcal{V}_\ell$ **<AU: Should these be \ldots or \cdots? It is \ldots on p 21>** (where ℓ is the cyclic index of $\hat{A}$). Let ξ_v be the MP of v (relative to $\hat{A}^T$); it is, then, the unique monic polynomial of the least degree such that $\xi_v(\hat{A}^T)v = 0$. Let $\alpha_1(\lambda)$ be the MP of $\hat{A}^T_{|\mathcal{V}_1}$; then, $\deg(\alpha_1(\lambda)) < M$. By the rational canonical structure theorem [53], there exists a vector $\hat{v}_1$ such that $\xi_{v_1}(\lambda) = \alpha_1(\lambda)$. Similarly, there exists a

vector $\hat{v}_2$ such that $\xi_{v_2}(\lambda) = \alpha_2(\lambda)$ [where $\alpha_2(\lambda)$ is the MP of $\hat{A}^T_{|\mathcal{V}_2}$]. Thus, ℓ vectors are obtained that form the measurement map $\tilde{K} = [\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_\ell]^T$. The construction of these vectors $\widehat{v}_i$ can be simplified by first performing the Jordan decomposition as $\hat{A}^T = P\Lambda P^{-1}$. Then, the vectors $\tilde{v}_i, i \in \{1, \ldots, \ell\}$ for Λ can be constructed such that the entries corresponding to the leading entries of Jordan blocks of $\Lambda_{|\mathcal{V}_i}$ are nonzero. Such a construction ensures that the MP of vector $\tilde{v}_i$ with respect to $\Lambda_{|\mathcal{V}}$ is also the corresponding MP of $\Lambda_{|\mathcal{V}_i}$. Finally, the required map can be obtained as $\tilde{K} = [\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_\ell]^T P^{-1}$. ■

The construction provided in the proof of Proposition 3 is utilized in Algorithm 1, which uses the rational canonical structure of $\hat{A}$ to generate a series of vectors $v_i \in \mathbb{R}^M$ whose iterations $\{v_1, \ldots, \hat{A}^{m_1 - 1} v_1, \ldots, v_\ell, \ldots, \hat{A}^{m_\ell - 1} v_\ell\}$ generate a basis for $\mathbb{R}^M$. Unfortunately, the measurement map $\tilde{K}$, being an abstract construction unrelated to the kernel, does not directly select $\mathcal{X}$. It is shown how to use the measurement map to guide a search for $\mathcal{X}$ in Remark 2. For now, a sufficient condition for observability of a general system is stated.

Theorem 1

Suppose that the conditions in Proposition 1 hold, with the relaxation that the Jordan blocks $[\Lambda_1 \quad \cdots \quad \Lambda_O]$ may have repeated eigenvalues. Let ℓ be the cyclic index of $\hat{A}$. Define

$$\mathbf{K} = [K^{(1)^T} \cdots K^{(\ell)^T}]^T \tag{8}$$

as the ℓ-shaded matrix [see Figure 4(b)], which consists of ℓ shaded matrices with the property that any subset of ℓ columns in the matrix is linearly independent from the others. Then, system (3) is observable if Υ has distinct values and $|\Upsilon| \geq M$.

Proof

A cyclic index of ℓ for this system implies that there exists an eigenvalue λ that's repeated ℓ times. The theorem is proved for repeated eigenvalues of dimension 1. The same statement can be proved for repeated eigenvalues for Jordan blocks using the ideas in the proof of Proposition 1. Without a loss of generality, let $\mathbf{K}$ have ℓ fully shaded, linearly independent rows, and assume that the column indices corresponding to this eigenvalue are $\{1, 2, \ldots, \ell\}$. Define $\boldsymbol{\lambda}_i := [\lambda_i^{\tau_1} \quad \lambda_i^{\tau_2} \quad \cdots \quad \lambda_i^{\tau_L}]^T$. Then,

$$O_\Upsilon := \begin{bmatrix} k_{11}\boldsymbol{\lambda}_1 & k_{12}\boldsymbol{\lambda}_2 & \cdots & k_{1M}\boldsymbol{\lambda}_M \\ \vdots & \vdots & \ddots & \vdots \\ k_{\ell 1}\boldsymbol{\lambda}_1 & k_{\ell 2}\boldsymbol{\lambda}_2 & \cdots & k_{\ell M}\boldsymbol{\lambda}_M \end{bmatrix}.$$

Let $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2 = \cdots \boldsymbol{\lambda}_\ell := \boldsymbol{\lambda}$. Focusing on these first ℓ columns of this matrix, this implies that constants $c_1, c_2, \ldots, c_{\ell-1}$ must be determined such that

$$\begin{bmatrix} k_{11} \\ \vdots \\ k_{\ell 1} \end{bmatrix} = c_1 \begin{bmatrix} k_{12} \\ \vdots \\ k_{\ell 2} \end{bmatrix} + \cdots + c_{\ell-1} \begin{bmatrix} k_{1\ell} \\ \vdots \\ k_{\ell\ell} \end{bmatrix}.$$

ALGORITHM 2 Sampling locations set $\mathcal{X}$.

Input: $\hat{A} = C$, lower bound ℓ
Decompose C to generate invariant subspaces $\hat{\mathcal{H}}_j, j \in \{1, 2, \ldots, \ell\}$ (see the "Preliminaries of Rational Canonical Structures" section)
for $j = 1$ **to** ℓ **do**
  Obtain centers $C^{(j)}$ with respect to subspace $\hat{\mathcal{H}}_j$,
  Generate samples $x_i^{(j)}$ to create a kernel matrix $K^{(j)}$ that is shaded only with respect to centers $C^{(j)}$
end for
Output: Sampling locations set $\mathcal{X} = \{x^{(1)}, x^{(2)} \ldots, x^{(l)}\}$.

However, these columns are linearly independent by assumption, and thus no such constants exist (implying that O_Υ is observable). ■

While Theorem 1 is quite a general result, the condition that any ℓ columns of $\mathbf{K}$ be linearly independent is a very stringent condition. One scenario where this condition can be met with minimal measurements is when the feature map $\hat{\psi}(x)$ is generated by a dictionary of atoms with the Gaussian RBF kernel evaluated at sampling locations $\{x_1, \ldots, x_N\}$ according to (2) (where $x_i \in \Omega \subset \mathbb{R}^d$), while x_i are sampled from a nondegenerate probability distribution on Ω, such as the uniform distribution. For a semideterministic approach, when the dynamics matrix $\hat{A}$ is block diagonal, utilize a simple heuristic.

Remark 2

Let Ω be compact, $C = \{c_1, \ldots, c_M\}$, $c_i \in \Omega$, and the approximate feature map be defined by (2). Consider the system (3) with $\hat{A} = \Lambda$, and let $\Upsilon = \{0, 1, \ldots, M-1\}$. Then, the measurement map $\tilde{K}$'s values lie in $\{0, 1\}$; specifically, each row $\tilde{K}^{(j)}, j \in \{1, \ldots, \ell\}$ corresponds to a subspace $\hat{\mathcal{H}}_j$ generated by a subset of centers $C^{(j)} \subset C$. Generate samples $x_i^{(j)}$ to create a kernel matrix $K^{(j)}$ that is shaded only with respect to centers $C^{(j)}$. Once this is completed, move on to the next subspace $\hat{\mathcal{H}}_{j+1}$. When all ℓ rows of $\tilde{K}$ are accounted for, construct the matrix $\mathbf{K}$, as in (8). Then, the resulting system $(\mathbf{K}, \hat{A})$ is observable.

This heuristic is formalized in Algorithm 2. Note that, in practice, the matrix $\hat{A}$ must be inferred from measurements of the process f_τ. If no assumptions are placed on $\hat{A}$, it is clear that at least M sensors are required for the system identification phase. Future work will study the precise conditions under which system identification is possible with fewer than M sensors.

DISCUSSION OF THEORETICAL RESULTS

The systems-theoretic approach in this article reveals something rather surprising: functions with complex dynamics (with a small cyclic index) can be recovered with fewer sensor placements than functions that have simpler

dynamics. Although seemingly counterintuitive, it becomes clear that this is because complex dynamics, which are characterized by a lower geometric multiplicity of the eigenvalues, ensure that the orbit $\boldsymbol{\Theta} := \{\hat{A}w_\tau\}_{\tau\in\Upsilon}$ traverses a greater portion of $\mathbb{R}^M \equiv \hat{\mathcal{H}}$, and thus fewer sensors can recover more geometric information. In simpler functional evolution, $\boldsymbol{\Theta}$ develops along strict subspaces of $\mathbb{R}^M$. Thus, more independent sensors are required to infer the same amount of information.

In the case described in Remark 2, we have a set of centers $C = \{c_1, \ldots, c_M\}$, which generates the bases $\mathcal{F}^C = \{\psi(c_1), \ldots, \psi(c_M)\}$. Let the cyclic index be (ℓ). This implies that there exist ℓ subsets $\boldsymbol{\Psi}^{(i)}$ of $\mathcal{F}^C$ with at least one element $\psi(c_j)$ each, leading to $\binom{M}{\ell}$ possible choices. Figure 8 represents these choices as hyperplanes separating the subsets. The measurement map described in [21, Algorithm 1] induces this *decomposition of bases* $\mathcal{F}^C = \{\boldsymbol{\Psi}^{(1)}, \ldots, \boldsymbol{\Psi}^{(\ell)}\}$ in polynomial time. Furthermore, each subset $\boldsymbol{\Psi}^{(i)}$ is directly associated to a subset of centers $C^{(i)} \subset C$, which enables selecting targeted sensor locations $x_i \in \Omega$. Specifically, for radially symmetric kernels, such as Gaussian, the centroid of the convex hull of $C^{(i)}$ is sufficient for generating a sensor placement. The measurement map is a significant theoretical insight into sensor placement for dynamically changing environments because it directly takes into account the dynamics of the process. Of course, in practice, this may be too expensive for approximate feature spaces where M is very large. Thus, random sampling can be used to generate the sensor locations, instead, at the cost of N being larger than ℓ. The advantage is that, since random sampling is computationally inexpensive, different choices of sensor placements can be generated and evaluated relatively quickly.

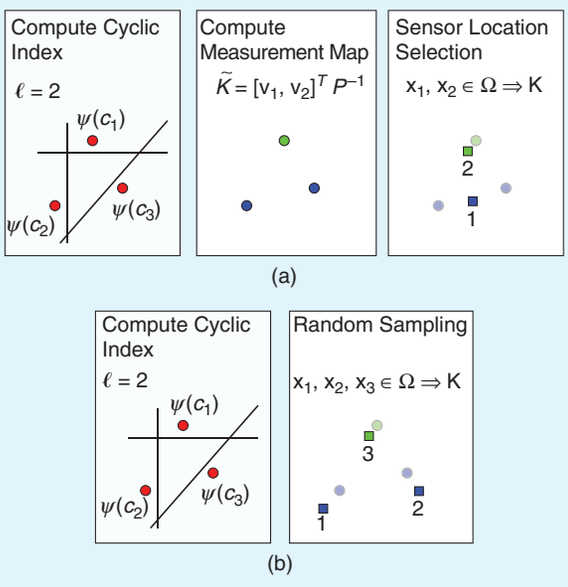

FIGURE 8 Sensor placement using the measurement map and random sampling approaches. The circles represent data locations associated to bases [such as $c_j \Leftrightarrow \psi(c_j)$], and the squares indicate sensor locations [such as $x_i \Leftrightarrow \psi(x_i)$]. The cyclic index ($\ell = 2$) shows how many possible couplings of bases exist, which can be represented as a choice of $\binom{M}{\ell}$ hyperplanes in Ω. (a) If the measurement map is computed, the correct couplings are chosen (green versus blue), and a smaller number of sensors (two) can be placed. (b) Alternatively, random sampling is more computationally efficient but generally requires more sensors (three).

Another point to note is that since the collection of bases $\{\hat{\psi}_i(x)\}_{i=1}^M$ determines the richness of the function space $\hat{\mathcal{H}} \approx \mathcal{H}$ we operate in, it governs the fidelity of the model approximation to the true time-varying function. As a consequence, observability of the system in $\hat{\mathcal{H}}$ refers to the best possible approximation in $\hat{\mathcal{H}}$. The greater the number of bases, the higher the dimensionality (which results in greater model fidelity but may require a much greater number of measurements for state recovery). This is where the lower bounds presented in the article are particularly useful. They show that for functional evolutions corresponding to a certain $\hat{A}$, *the number of sensor placements is essentially independent of the dimensionality* M but depends, rather, on the cyclic index of $\hat{A}$. Figure 7 gives an overall picture of the process of generating a kernel observer, while Figure 8 displays two approaches to sensor selection in our framework. The measurement map approach can generate a smaller set of sensors than the random placement approach but comes at an additional computational cost.

RANDOM SENSOR PLACEMENT

We now elaborate on how the challenging problem of sensor placement can be addressed through random selection. The process of random selection is a product of the kernel observer model described previously. The theoretical background required to prove Theorem 2 (which states the expected number of randomly placed sensors required to monitor a given spatiotemporal process) and Theorem 3 (which determines the probability with which optimal sensor placement is ensured) is presented, given that N number of sensors has been placed. As discussed earlier, an approximate feature space $\hat{\mathcal{H}}$ is used, with the corresponding transition operator $\hat{A}: \hat{\mathcal{H}} \to \hat{\mathcal{H}}$ representing finite-dimensional functional evolution.

To achieve observability for the pair $(\hat{A}, K)$, row vectors of the corresponding observability matrix $\boldsymbol{O}$ should form the basis for the $\mathbb{R}^M$-dimensional space $\hat{\mathcal{H}}$. According to the rational canonical structure theorem [53], $\hat{A}$ can successively decompose the dual space $\mathbb{R}^M$ into subspaces, $\mathcal{V}_i \subset \mathcal{V}, i \in \{1, \ldots, \ell\}$, which have the following properties: 1) $\mathcal{V} = \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_\ell$, 2) $\hat{A}\mathcal{V}_i \subset \mathcal{V}_i$, and 3) $\hat{A}|_{\mathcal{V}_i}, i \in \{1, \ldots, \ell\}$ are cyclic. The integer ℓ is unique and called the *cyclic index of* $\hat{A}$. Each of these properties contributes to the theorem related to the number of random samples required to achieve observability. The first

property shows that the space $\mathbb{R}^M$ can be decomposed into ℓ independent subspaces. The second property shows that the vector $v_i \in \mathcal{V}_i$ stays in $\mathcal{V}_i$, even when operated on by $\hat{A}$. To generate bases for $\mathbb{R}^M$, at least ℓ vectors $v_1, \ldots, v_\ell$ are needed, where each v_i arises from a corresponding subspace $\mathcal{V}_i$ from the set of subspaces $\mathcal{V}_1, \ldots, \mathcal{V}_\ell$. This holds due to the third property, but it requires that the vectors $v_1, \ldots, v_\ell$ are the cyclic generators of their corresponding subspaces.

This analysis is based on whether a randomly selected sensor can generate a cyclic generator. To examine this, recall that a row vector $K_{(i)}$ generated by a randomly selected sensor location x_i takes the form

$$K_{(i)} = [k(x_i, c_1), \ldots, k(x_i, c_M)]. \tag{9}$$

For radial kernels, for example, the entries corresponding to the centers closer to x_i tend to be nonzero, whereas the others tend to be zero. Overall, our construction is as follows. For each subspace $\mathcal{V}_i$, let $C_{\mathcal{V}_i} \subset C$ be the centers corresponding to the leading entries of the Jordan blocks. The minimum number of random samples required to generate the bases for $\mathcal{V}_i$ is equal to the number of Jordan blocks comprising $\mathcal{V}_i$. Altogether, the minimum number of random samples required to generate a basis for $\mathbb{R}^M$ is equal to the total number of Jordan blocks in $\hat{A}$.

Let ς be the total number of Jordan blocks in $\hat{A}$. Then,

$$\varsigma = \sum_{\lambda \in \sigma(\hat{A})} \gamma_{\hat{A}}(\lambda), \tag{10}$$

where $\sigma(\hat{A})$ represents the spectrum of $\hat{A}$ whose elements are the eigenvalues of $\hat{A}$ and $\gamma_{\hat{A}}(\lambda)$ is the geometric multiplicity corresponding to the eigenvalue λ (which is also equal to the total number of Jordan blocks corresponding to the eigenvalue λ). Define a set of centers C_ς with elements $\{c_1, c_2, \ldots, c_\varsigma\}$ to be the centers corresponding to the leading entries of the Jordan blocks. For sensor location $x \in \Omega$ and $\epsilon > 0$, let $k(x, c_j) > \epsilon$. Denote the region $\Omega_j \subset \Omega$ such that the kernel evaluation with respect to center c_j is greater than ϵ; that is, $\Omega_j \equiv \{x \in \Omega : k(x, c_j) > \epsilon\}$. Define p_ϵ as

$$p_\epsilon = \min_{c_j \in C_\varsigma} \frac{\nu(k(x, c_j) > \epsilon)}{\nu(\Omega)}, \tag{11}$$

where ν is a measure in the real analysis sense. Hence, p_ϵ corresponds to a lower bound on the probability that a random sample lies within the ϵ-shaded region of a particular center c_j. With all of this in place, the following theorem can be proved.

Theorem 2

Given the spatiotemporal function $f(x, \tau)$ with $x \in \Omega \subseteq \mathbb{R}^D, \tau \in \mathbb{Z}^+$, kernel observer model (3), and tolerance parameter $\epsilon > 0$, the expected number of randomly placed sensor locations required to achieve observability for the pair $(K, \hat{A})$ is ς / p_ϵ, where ς is the summation across geometric multiplicities of each $\lambda \in \sigma(\hat{A})$ given by (10).

Proof

For each random sample, the probability that it lies within the ϵ-shaded region of a particular center $c_j \in \mathbb{C}_\varsigma$ is at least p_ϵ. The series of random samples can be considered Bernoulli trials in which p_ϵ is the probability of a successful outcome. Note that this assumes the worst-case scenario that the intersection between any two ϵ-shaded regions of centers belonging to the set C_ς is empty. Observability for the pair $(K, \hat{A})$ is achieved after ς successful outcomes are obtained because each success ensures a row vector with a nonzero entry corresponding to the leading entry of the Jordan block.

Let $X_1, X_2, \ldots, X_N$ be independent identically distributed random variables whose common distribution is the Bernoulli distribution with parameter p_ϵ. The random variable $X = X_1 + X_2 + \cdots + X_N$ denotes the number of successes after a certain number N of random samples. Since each X_i has the Bernoulli distribution, X will have a binomial distribution,

$$P(X = h) = \binom{N}{h} p_\epsilon^h (1 - p_\epsilon)^{N-h},$$

in which h is the number of success. The expectation for the binomial distribution (that is, the expected number of success) is Np_ϵ, and thus the expected number of required trials will be $N = \varsigma / p_\epsilon$. ■

Theorem 3

Given the spatiotemporal function $f(x, \tau)$ with $x \in \Omega \subseteq \mathbb{R}^D; \tau \in \mathbb{Z}^+$; kernel observer model (3); tolerance parameter $\epsilon > 0$; summation across geometric multiplicities of each $\lambda \in \sigma(\hat{A})$, denoted by ς, as in (10); and a constant $\delta \in (0, 1]$, the probability that pair $(K, \hat{A})$ is unobservable after the selection of N random sensors is, at most, $e^{\frac{-1}{2}(Np_\epsilon - 2\varsigma)}$, where p_ϵ is given by (11) and $N > 2\varsigma / p_\epsilon$.

Proof

The random variable X from the proof of Theorem 1 has a binomial distribution, which enables the application of a Chernoff-type bound on its tail probabilities. A well-known result on the multiplicative Chernoff bound [55] is directly applied to establish this theorem. If X is binomially distributed, $\delta \in (0, 1]$, and $\mu = \mathbb{E}[X]$, then $P[X \leq (1 - \delta)\mu] \leq \exp(-\mu\delta^2/2)$, where $\delta := 1 - \varsigma/(Np_\epsilon)$. The expression in the exponent can be simplified to $-(1/2)Np_\epsilon + \varsigma - (\varsigma^2/2Np_\epsilon)$ using $\mu = Np_\epsilon$. Note that $e(-\varsigma^2/2Np_\epsilon) \leq 1$. This implies that

$$\exp(-\mu\delta^2/2) = e^{-\frac{1}{2}(Np_\epsilon - 2\varsigma)} \cdot e^{\frac{-\varsigma^2}{2Np_\epsilon}} \leq e^{-\frac{1}{2}(Np_\epsilon - 2\varsigma)}.$$

Note that $(1 - \delta)\mu = \varsigma$, and thus $P[X \leq \varsigma] \leq e^{-\frac{1}{2}(Np_\epsilon - 2\varsigma)}$. ■

For the case when $\hat{A} \neq \Lambda$, a change of basis can be used to obtain $\Lambda = P^{-1}\hat{A}P$, where P is the projection map. There are two challenges in performing the preceding analysis for Λ. First, the leading entries of the Jordan blocks do not directly correspond to the centers $\{c_1, \ldots, c_M\}$, which was the case for $\hat{A} = \Lambda$. Second, although the transformation of the row vector (9) is obtained using the projection map P, the definition of the probability p_ϵ is no longer obtained, as in (11). The existence of the similarity transform hints that the results in Theorems 2 and 3 should hold for any $\hat{A}$. However, the mathematical tools utilized in the article seem to be insufficient to prove them. Nevertheless, we present evidence for these claims for when $\hat{A} \neq \Lambda$, in the "Empirical Results" section.

GENERALIZING ACROSS SIMILAR SPATIOTEMPORALLY EVOLVING SYSTEMS

Building on the kernel observers method, the E-GP is introduced. The primary novelty in this technique for generating a model concerns learning an $\hat{A}$ matrix for *multiple* systems. The ultimate goal of this research would be to generate highly efficient machine learning models that can be used instead of costly numerical simulations for design and autonomy purposes. This would be a major success for the design and control of complex physical systems, such as soft robotics, as machine learning models would significantly reduce the cost and resources required for simulations. The ability to generalize across different physical situations is critical. 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. For example, in the context of fluid flows, these models must be able to predict fluid dynamics in conditions (such as the Reynolds number) that differ from the training data.

The E-GP, as best determined, was the first machine learning method to generalize across spatiotemporally evolving systems of such complexity using end-to-end data. It was found that the class of functional evolutions $\mathbb{F}$ defined by linear Markovian transitions in an RKHS is still sufficient to model the nonlinear Navier–Stokes equations, which govern fluid dynamics, since the unknown map enables the representation of highly nonlinear dynamics in the input space. However, it is expected that phenomena such as bifurcation and turbulence will require nonlinear mappings $\mathcal{H}$. There are three steps to generate an E-GP model.

<AU: Four steps are listed. Should it be four steps?>

1) After selecting the kernel and estimating the bandwidth hyperparameter σ (utilizing the maximum likelihood approach, although other methods can be used), find an optimal basis vector set C using the algorithm in [56].
2) Use GP 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 the next step easier but can be worked around for nonuniform data sets
3) Using the weight trajectory, use matrix least squares with the equation $\hat{A}[w_1, w_2, \ldots, w_{T-1}] = [w_2, w_3, \ldots, w_T]$ to solve for $\hat{A}$.
4) To generate a multisystem model, concatenate the weight trajectories from each similar system in the least-squares computation of $\hat{A}$. That is, let $W_\theta = [w_1^{(\theta)}, w_2^{(\theta)}, \ldots, w_{n-1}^{(\theta)}]$ and $W'_\theta = [w_2^{(\theta)}, w_3^{(\theta)}, \ldots, w_n^{(\theta)}]$ be the weight trajectory and next weight trajectory for some parameter. Then, solve the least-squares problem $\hat{A}[W_{\theta_1}, \ldots, W_{\theta_n}] = [W'_\theta, \ldots, W'_{\theta_n}]$.

For the sake of defining when it is appropriate to expect this method to generalize across different spatiotemporally evolving systems, we specify what it means for two fluid flows to be *similar*. In configuring a fluid dynamics simulation, a set of quantifiable parameters is 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 constraint. Furthermore, it is required that the constraint be continuously variable, and any observable quantity 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, thus producing a nonsimilar 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. This attribute of E-GPs is referred to as the *spatial encoding property*. This property is an extremely valuable tool for gaining insight into how the learned model works.

1) By plotting which kernel centers are associated with which invariant subspaces in the transition matrix, we can visualize where the eigenfunctions are found and how the dynamic modes are spatially separated.
2) By plotting arrows from center c_j to c_i for each of the largest elements $\hat{a}_{ij}$ of $\hat{A}$, we can visualize how different areas of the domain influence one another's evolution.
3) By performing an eigendecomposition of the $\hat{A}$ matrix and transforming the eigenvectors back from the weight space to the function space, the Koopman modes (and associated eigenvalues) of the system can be obtained (see the next section).

SPECTRAL ANALYSIS OF EVOLVING GAUSSIAN PROCESS MODELS

We would be remiss not to examine our methods in light of Koopman operator theory, which has served as a major source of inspiration for a number of new methods of analyzing dynamical systems during the past decade,

especially within the CFD community. The results are presented, showing a direct connection between the Koopman modes, eigenfunctions, and eigenvalues and the spectral decomposition of the transition matrix in this model. In fact, the final theorem shows that the eigenvalues of the model are a subset of the eigenvalues of the infinite-dimensional Koopman operator, the eigenvectors transformed to the input space are congruent in shape to the Koopman modes, and the eigenfunctions are identical. These results enabled the construction of new methods for sensor placement under even more difficult conditions than described previously—specifically, the situation of one (or a few) sensors attached to moving agents and robots.

For a general dynamical system $f_{\tau+1} = \mathbb{F}(f_\tau)$ defined in a state space (in this case, an RKHS $f \in \mathcal{H}$), define an arbitrary, vector-valued *observable* $g : \mathcal{H} \to \mathbb{R}^N$. Note that the space of observables g is a vector space. The Koopman operator U is defined to be the operator on the space of observables such that

$$U_g(f_\tau) = g(\mathbb{F}(f_\tau)) = g(f_{\tau+1}). \tag{12}$$

This operator is clearly linear, from its definition. Thus, it is reasonable to examine its spectral properties. The special observables $\phi : \mathcal{H} \to \mathbb{C}$ that have the property

$$U_\phi(f_\tau) = \phi(\mathbb{F}(f_\tau)) = \phi(f_{\tau+1}) = \lambda\phi(f_\tau) \tag{13}$$

are the eigenfunctions of U, and the associated λ are the eigenvalues. Suppose a vector-valued observable $u(f, x)$, where $x \in \Omega$ and $f \in \mathcal{H}$.

Definition 2

The Koopman mode $s(x)$ at isolated eigenvalue λ of algebraic multiplicity 1 is the projection of $u(f, x)$ onto the eigenfunction $\phi_\lambda(f)$ of U at λ [57]. As previously shown by Rowley in 2009 [58], the modes produced by the DMD algorithm constitute a subset of Koopman modes. Mezic (2013) [57] showed that there exists, in principle if not in practice, a rigorous method for computing the full set of Koopman modes by a method known as generalized Laplace analysis (GLA). A number of results interpreting the E-GP model in terms of Koopman operator theory has been generated, culminating in the proof that the eigenvalues and eigenvectors of $\hat{A}$ are related to the Koopman eigenvalues and modes.

Proposition 4

Let $\mathcal{H}$ be an RKHS with an approximate feature space $\hat{\mathcal{H}}$, and let $u(f, x)$ be an observable in the Koopman sense with respect to the dynamical system $f_{\tau+1} = \mathbb{F}(f_\tau)$ in $\mathcal{H}$. Then, $\hat{u}(f,x) := u(\hat{f}, x)$, where $\hat{f}$ the projection of $f \in \mathcal{H}$ onto $\hat{\mathcal{H}}$ is also an observable. This follows from the fact that the projection from the function space to its subspace is well defined when $\hat{\psi}_i$ are independent. Koopman modes of observables are of interest because they are akin to the eigenvector expansions utilized in linear dynamics. If the Koopman operator is separated into $U = U_s + U_r$ (where U_s has a pure point spectrum with k points and U_r has a pure continuous spectrum), then

$$U^t u(f,x) = u^*(f,x) + \sum_{j=1}^{k} \lambda_j^t \phi_j(f) s_j(x) + U_r^t u(f,x) \tag{14}$$

can be written, where $u^*(x)$ represents the time-averaged part of the field, which corresponds with $\lambda = 1$ (see the GLA in [57]). The continuous spectrum component is usually discarded/neglected since it represents the part of the field that is genuinely aperiodic (or chaotic) in time, which could be modeled as a stochastic process [57]. From this expansion, the following proposition can be proved.

Proposition 5

Let $u(f, x)$ and $\hat{u}(f, x)$ be observables, as in Proposition 4, and let $s_j(x)$ be the Koopman modes associated with the projection of the former onto the eigenfunctions ϕ_j of U at λ_j. The Koopman modes associated with $\hat{u}$ are

$$\hat{s}_j(x) = \frac{\phi_j(\hat{f})}{\phi_j(f)} s_j(x). \tag{15}$$

Proof

Using the spectral expansion and the definition of the observables,

$$\begin{aligned} U^t \hat{u}(f,x) &= U^t u(\hat{f},x)\hat{u}^*(f,x) + \sum_{j=1}^{k} \lambda_j^t \phi_j(f)\hat{s}_j(x) + U_r^t \hat{u}(f,x) \\ &= u^*(\hat{f},x) + \sum_{j=1}^{k} \lambda_j^t \phi_j(\hat{f}) s_j(x) + U_r^t u(\hat{f},x). \end{aligned}$$

Since this must be true for all t, the terms must match, and the hypothesis follows. ■

Note that this means the modes of the two observables are identical in shape (differing only by a multiplicative constant). In the exceptional case that $\phi_j(f) = 0$ the GLA method is unable to compute $s_j(x)$ anyway, so that is not an issue for this proposition. The final step in this analysis is to connect the Koopman modes with the spectral decomposition of this model's $\hat{A}$ operator in the dual space.

Theorem 4

In an E-GP or a kernel observer model of the dynamical system $f_{\tau+1} = \mathcal{A} f_\tau$, the following statements are true:

1) The eigenvalues of $\hat{A}$ are a subset of the eigenvalues of the Koopman operator.
2) The eigenfunctions of the Koopman operator in the dual space correspond with a subset of the eigenfunctions of the Koopman operator on f_0; that is, $\hat{\phi}_j(w_\tau) = \phi_j(f_\tau)$.
3) If the observable $u(f,x)$ is the evaluation operator $u(f,x) := f(x)$, then the eigenvectors v_j of $\hat{A}$ are related to the Koopman modes of the observable by

$$s_j(x) = \frac{\phi_j(f)}{\phi_j(\hat{f})}\hat{\Psi}(x)\cdot v_j, \tag{16}$$

where $\hat{\Psi}(x)$ is the feature map of the model as a row vector.

Proof

It should be clear that as long as the projection onto the approximate feature space is well defined, $P(f_\tau) := w_\tau$ is an observable. If $\phi_j(f) = \langle P(f), q_j \rangle$, where q_j are eigenvectors of the adjoint $\hat{A}^*$ (that is, $\hat{A}^* q_j = \bar{\lambda}_j q_j$), then ϕ_j is an eigenfunction of the Koopman operator in $\mathcal{H}$, since $U\phi_j(f_\tau) = \langle P(f_{\tau+1}), q_j \rangle = \langle w_{\tau+1}, q_j \rangle = \langle \hat{A} w_\tau, q_j \rangle = \langle w_\tau, \hat{A}^* q_j \rangle = \lambda_j \langle w_\tau, q_j \rangle = \lambda_j \phi_j(f_\tau)$. This shows the following:

1) The eigenvalues of $\hat{A}$ correspond with that of the Koopman operator.
2) The eigenfunctions of the Koopman operator in the dual space, known to be $\hat{\phi}_j(\cdot) = \langle \cdot, q_j \rangle$, correspond with the eigenfunctions of the Koopman operator in $\mathcal{H}$. In the case that $u(f,x) := f(x)$, there exists the relationship $\hat{u}(f,x) = \hat{f}(x) = \hat{\Psi}(x)\cdot w(f)$ according to the formulation of the model. It is well known that the spectral expansion of a finite linear system is

$$w_t = \sum_{j=1}^{M} \lambda_j^t \hat{\phi}_j(w_0) v_j,$$

where v_j are the eigenvectors of the transition matrix. Therefore,

$$\begin{aligned} U^t \hat{u}(f_0,x) &= U^t(\hat{\Psi}(x)\cdot w(f_0)) \\ &= \hat{\Psi}(x)\cdot w_t \\ &= \hat{\Psi}(x)\cdot\left(\sum_{j=1}^{M} \lambda_j \hat{\phi}_j(w_0) v_j\right) \\ &= \sum_{j=1}^{M} \lambda_j \phi_j(f_0)\hat{\Psi}(x)\cdot v_j. \end{aligned}$$

Comparing this to the spectral expansion of $\hat{u}(f,x)$ (and letting $\lambda_1 = 1$),

3) $\hat{u}^*(f,x) = \hat{\Psi}(x)\cdot v_1$, $\hat{s}_j(x) = \hat{\Psi}(x)\cdot v_j$, and there is no continuous spectrum component. The hypothesis now follows from Proposition 5. ■

Ultimately, a direct connection between the spectral decomposition of the transition matrix in the dual space of the approximate feature space and the Koopman modes, eigenfunctions, and eigenvalues was established.

Corollary 1

Suppose that given $\epsilon > 0$, an approximate feature space $\hat{\mathcal{H}}$ was determined such that for all $f \in \mathcal{H}$, there exists $\hat{f} \in \hat{\mathcal{H}}$ such that $\|f - \hat{f}\| < \epsilon$. If $\phi_j(f) \neq 0$, there exists an approximate feature space such that $\hat{\Psi}(x)\cdot v_j$ is arbitrarily close to the Koopman mode $s_j(x)$ (where v_j is an eigenvector of the dual space transition matrix $\hat{A}$).

Proof

Since ϕ_j is continuous, then $|\phi_j(f) - \phi_j(\hat{f})|$ can be made arbitrarily small. This means that if $\phi_j f \neq 0$, then $(\phi_j(f)/\phi_j(\hat{f}))$ can be made arbitrarily close to one. From Theorem 4 and the fact that $s_j \in \mathcal{H}$ means s_j is bounded, it is concluded that $\hat{\Psi}(x)\cdot v_j$ can be made arbitrarily close to $s_j(x)$ everywhere in Ω. ■

This implies that if a close-enough approximate feature space can be determined, then the eigenvectors of the transition matrix in the dual space correspond exactly with the Koopman modes in the input space.

INVARIANT SUBSPACES

One way of viewing the invariant subspaces concept is to say that information contained in an invariant subspace never leaves that invariant subspace. It is hypothesized that the kernel centers associated with the invariant subspaces of a spatiotemporally evolving system are generally associated with spatial regions in the domain (and not just homogeneously spread all over and mixed with the other invariant subspaces). This hypothesis makes sense both physically and mathematically. In physics, the *principle of locality* states that an object is only directly influenced by its immediate surroundings [59]. If this is the case (and there is no reason to believe that it is not, apart from certain quantum dynamics situations) and the E-GP model accurately captures the physics of the system, then information (measurable phenomena) may continuously travel only from one point in the domain to another. Mathematically, since a value at any one point in the domain is influenced by the weights of multiple nearby centers, proximate centers are expected to be dynamically connected unless separated by "plains" (where the values are indistinguishable from noise). In an ideal case, the Jordan form of an $n \times n$ matrix $\hat{A}$ is block diagonal and therefore gives a decomposition of the n-dimensional Euclidean space into clearly separated invariant subspaces of $\hat{A}$. The cyclic index, which can be found by counting the geometric multiplicities of eigenvalues in $\hat{A}$, gives the number of invariant subspaces.

In practice, data-driven approximations of $\hat{A}$ rarely give such easily interpretable properties. This fact drives the need for an algorithm that can divide the system into invariant subspaces, even when the boundaries between such spaces are not mathematically precise. This algorithm can be thought of as a soft Jordan decomposition. Each block in the normal Jordan form has a set of corresponding eigenvectors and an eigenvalue with geometric multiplicity. When the former is transformed back into the domain space, complex-valued functions are obtained that are the Koopman modes of the system. These provide an image of what kind of structures are seen in the dynamics. The eigenvalues describe the frequency with which these structures oscillate between their real and imaginary forms as well as their exponential growth or decay in magnitude.

k-INVARIANT SUBSPACE CLUSTERING

In response to the need for an algorithm that can separate invariant subspaces in the linear model of a system

generated by data, *k-invariant subspaces clustering* was developed. The intuition behind this algorithm is to replace the Euclidean distance in the well-known *k*-means algorithm with a different metric of "nearness," namely one corresponding with the dynamic connections in the space. The $\hat{A}$ matrix provides easy access to these. Its rows $\hat{A}_{i*}$ indicate which centers inform the ith value of $w_{\tau+1}$, and its columns $\hat{A}_{*j}$ indicate which centers will be informed by the jth value of w_τ. However, these values tend to be biased toward the eigenmodes with higher frequencies (that is, those whose eigenvalues have a greater polar angle) and those which grow exponentially. To control for this, the eigenvalues of the matrix were modified as follows: 1) zero any eigenvalue that is clearly inside (not on) the unit circle, 2) unitize the remaining eigenvalues, and 3) adjust the frequency of the remaining eigenvalues on the unit circle to either $\pm(\pi/4)$. If the eigendecomposition of the original was $\hat{A} = UDU^{-1}$, then construct a new matrix with the modified eigenvalues $\bar{D}$ as $\bar{A} = U\bar{D}U^{-1}$.

Much like the pairwise-squared deviations of the points formulation of the *k*-means clustering problem, the problem can be written as

$$\underset{S}{\operatorname{argmax}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\substack{x_i \neq x_j \\ x_j, x_i \in S_i}} \bar{A}_{ij}^2.$$

This problem can be solved with Algorithm 3. **<AU: Algorithms were renumbered to be sequential in the text. Please check that the statement is okay.>** Note the differences between this and the *k*-means clustering algorithm. First, we are *maximizing* since the terms represent influence rather than distance. Second, in the possible case that $|S_k| = 0$, the total value is set to zero to avoid division by zero. Third, exclude the centers' influence on themselves from the score by taking $S_k \setminus \{i\}$. This algorithm can be repeated as many times as desired, with different random initial conditions, and the result that produces the highest score is retained.

EMPIRICAL RESULTS

We begin with numerical results obtained for determining sensing locations through the application of kernel observers to the prediction of spatiotemporal functions. The highlight here is a set of results related to sensing global ocean temperatures by using information from just a few locations, employing *Advanced Very-High-Resolution Radiometer* (*AVHRR*) satellite data. This is followed by an application of the E-GP in predicting spatiotemporal functions. Results were included that predicted weed growth in simulated agricultural fields. The application of the E-GP to model the flow past a cylinder at different Reynolds numbers is explored in "Learning Fluid Flows With Evolving Gaussian Processes." The section concludes with a comparison of eigenvalues and Koopman modes computed from the E-GP to those of DMD.

ALGORITHM 3 The *k*-invariant subspaces algorithm.

while clusters have changed **do**
  for each center i **do**
    Find cluster k that maximizes the score $\frac{1}{|S_k|}(\|\bar{A}_{i,S_k\setminus\{i\}}\|^2 + \|\bar{A}_{S_k\setminus\{i\},i}\|^2)$
    Reassign center i to cluster with highest score
  end for
end while
return clusters

Sensing Locations for Synthetic Data Sets

The goal of this experiment is to investigate the dependency of the observability of system (3) on the shaded observation matrix and the lower bound presented in Proposition 2. The domain is fixed on the interval $\Omega = [0, 2\pi]$. First, select sets of points $C^{(i)} = \{c_1, \ldots, c_{M_i}\}$, $c_j \in \Omega$, and $M = 50$, and construct a dynamics matrix $A = \Lambda \in \mathbb{R}^{M\times M}$, with cyclic index 5. Select the RBF kernel $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$, $\sigma = 0.02$. Generating samples $\mathcal{X} = \{x_1, \ldots, x_N\}$, $x_i \in \Omega$ randomly, compute the ℓ-shaded property and observability for this system. Figure 9(a)

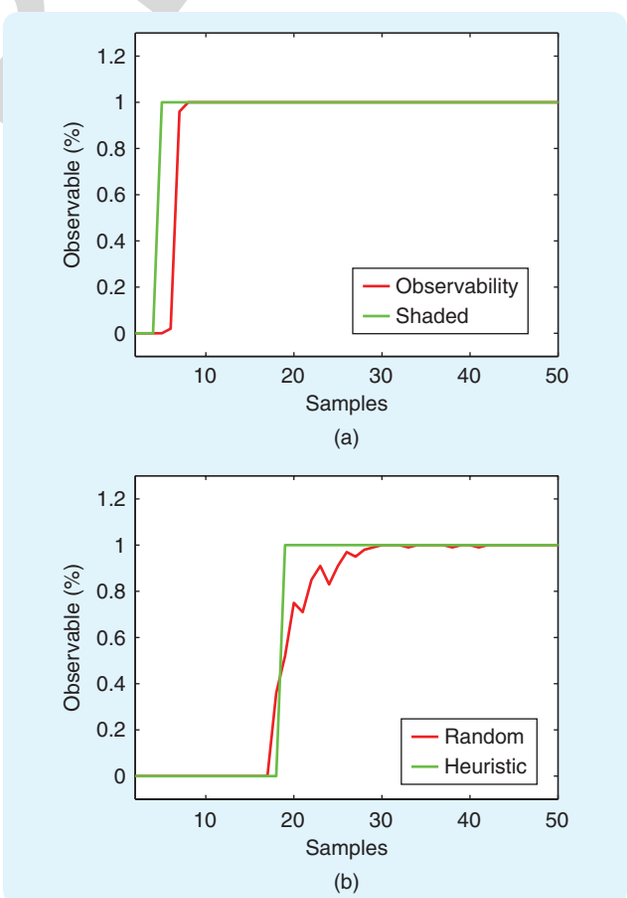

FIGURE 9 Kernel observability results. (a) Shaded versus observability. (b) Heuristic versus random.

shows how shadedness is a necessary condition for observability, validating Proposition 1: the slight gap between shadedness and observability, here, can be explained due to numerical issues in computing the rank of O_Υ.

Next, consider a system with a cyclic index $\ell = 18$ to verify random-sensor placement results. The measurement operator K was constructed using the heuristic in Remark 2 (Algorithm 2) and random-sensor selection to generate the sampling locations $\mathcal{X}$. These results are presented in Figure 9(b). The plot for random sampling, which has been averaged across 100 runs, resembles a cumulative distribution function of an exponential distribution $F(X = x) = 1 - \exp(-\lambda x)$. This verifies the claim made in Theorem 3, as the probability of becoming unobservable decays exponentially with the number of placed sensors. Also, fitting an exponential distribution, it was determined that the mean λ^{-1} comes close to the ratio ς/p, which is the expected number of randomly placed sensors required for observability, as per Theorem 2. Note that observability is not achieved if the number of samples $N < \ell$, which is experimental evidence of the result in Proposition 2.

ALGORITHM 4 The kernel observer (transition learning).

Input: Kernel k, basis centers C, final time step T.
while $\tau \leq T$ **do**
1) Sample data $\{y_\tau^i\}_{i=1}^M$ from f_τ.
2) Solve for $\hat{w}_\tau$ by using least squares on $y_\tau = K\hat{w}_\tau$.
3) Store weights $\hat{\mathcal{W}}_\tau$ in matrix $\mathcal{W} \in \mathbb{R}^{M\times T}$.
end while
To infer $\hat{A}$, define matrix $\Phi = \mathcal{W}^T\mathcal{W}$. Then:
for $i = 1$ to M **do**
At step i, solve system

$$\hat{A}^{(i)} = ((\Phi + \lambda I)^{-1}(\mathcal{W}^T\mathcal{W}^{(i)}))^T, \tag{17}$$

where $\hat{A}^{(i)}$ and $\mathcal{W}^{(i)}$ are the ith columns of $\hat{A}$ and $\mathcal{W}^{(i)}$, respectively.
end for
Compute the covariance matrix $\hat{B}$ of the observed weights $\mathcal{W}$.
Output: estimated transition matrix $\hat{A}$, predictive covariance matrix $\hat{B}$.

ALGORITHM 5 The kernel observer (monitoring and prediction).

Input: Kernel k, basis centers C, estimated system matrix $\hat{A}$, estimated covariance matrix $\hat{B}$.
Compute Observation Matrix: Compute the cyclic index ℓ of $\hat{A}$, and compute K.
Initialize Observer: Use $\hat{A}, \hat{B}$, and K to initialize a state observer [such as a Kalman filter] on $\hat{\mathcal{H}}$.
while measurements available **do**
1) Sample data $\{y_\tau^i\}_{i=1}^N$ from f_τ.
2) Propagate Kalman filter estimate $\hat{w}_\tau$ forward to time $\tau + 1$, correct using measurement feedback with $\{y_{\tau+1}^i\}_{i=1}^N$.
3) Output predicted function $\hat{f}_{\tau+1}$ of Kalman filter.
end while

Comparison With Nonstationary Kernel Methods on Real-World Data

Three real-world data sets were used to evaluate and compare the kernel observer with the two different lines of approach for nonstationary kernels discussed in the related work. The process convolution with a local smoothing kernel (PCLSK) and the latent extension of the input space (LEIS) are compared with the nonstationary space-time variable latent length **<AU: Please check that NOSTILL is spelled out correctly.>** GP in [8] and [37], respectively, using Intel Berkeley, Irish wind, and ozone data sets. The model inference for the kernel observer involved the following three steps:

1) Selecting the Gaussian RBF kernel $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$, a search for the ideal σ is performed for a sparse GP model (with a fixed basis vector set C selected using the method in [56]). For the data set discussed in this section, the number of basis vectors was equal to that of the sensing locations in the training set, with the domain for the input set defined across $\mathbb{R}^2$.
2) Having obtained σ, GP inference is used to generate weight vectors for each time step in the training set, resulting in the sequence $w_\tau, \tau \in \{1,\ldots,T\}$.
3) Matrix least squares is applied to this sequence to infer $\hat{A}$ (Algorithm 4).

For prediction in the autonomous setup, $\hat{A}$ is used to propagate the state w_τ forward to make estimates with no feedback, and in the observer setup, a Kalman filter (Algorithm 5), with N determined using Proposition 2 (and locations picked randomly), is used to propagate w_τ forward to make forecasts. This is also compared with a baseline GP (denoted by *original GP*), which is the sparse GP model trained using all the available data.

The first data set, the Intel Berkeley research lab temperature information, consists of 50 wireless temperature sensors in an indoor laboratory region spanning 40.5 m in length and 31 m in width (http://db.csail.mit.edu/labdata/labdata.html). The training data consists of temperature information from March 6, 2004 at intervals of 20 min (beginning 12:20 a.m.), which totals to 72 time steps. Testing is performed across another 72 time steps, beginning 12:20 p.m. on the same day. Out of 50 locations, 25 were uniformly selected for training and testing purposes. Results of the prediction error are shown as a box plot in Figure 10(a) and as a time series in Figure 10(b). Note that, in the figure, *Auto* refers to autonomous setup. The cyclic index of $\hat{A}$ was determined to be two, so N was set to two for the kernel

observer with feedback. Note that even the autonomous kernel observer outperforms the PCLSK and the LEIS, overall, and the kernel observer with feedback with $N = 2$ significantly outperforms all the other methods (which is why we did not include results with $N > 2$).

The second data set contains wind information consisting of daily average wind speed measurements collected, from 1961 to 1978, at 12 meteorological stations in the Republic of Ireland (http://lib.stat.cmu.edu/datasets/wind.desc). The prediction error is given as a box plot in Figure 11(a) and as a time series in Figure 11(b). Again, the cyclic index of $\hat{A}$ was determined to be two. In this case, the autonomous kernel observer's performance is comparable to the PCLSK and LEIS, while the kernel observer with feedback with $N = 2$ again outperforms all the other methods. Finally, the third data set records the ozone concentration (in parts per billion) measured at 60 stations across the country by the United States Environmental Protection Agency [60]. Due to missing measurements, only data from 1997 to 2013 are selected for training and evaluation. Each station averaged the ozone concentration during a period of three months, resulting in four quarters per year. Out of 60 sensor locations, 30 were uniformly selected for training, and the remaining stations were chosen for testing purposes. The prediction error results are presented as a box plot in Figure 12(a) and as a time series in Figure 12(b). Here, the cyclic index of $\hat{A}$ was determined to be one. In this case, the performance of the autonomous kernel observer is comparable to the PCLSK and LEIS, with the kernel observer with feedback with $N = 1$ performing the best. Table 1 reports the total training and prediction times associated with the PCLSK, LEIS, and kernel observer. It was observed that 1) the kernel observer is an order of magnitude faster than the competing methods and 2) even for small sets, competing methods did not scale well.

Prediction of Global Ocean Surface Temperature

The feasibility of this approach was analyzed on a very large data set from the National Oceanographic Data Center:

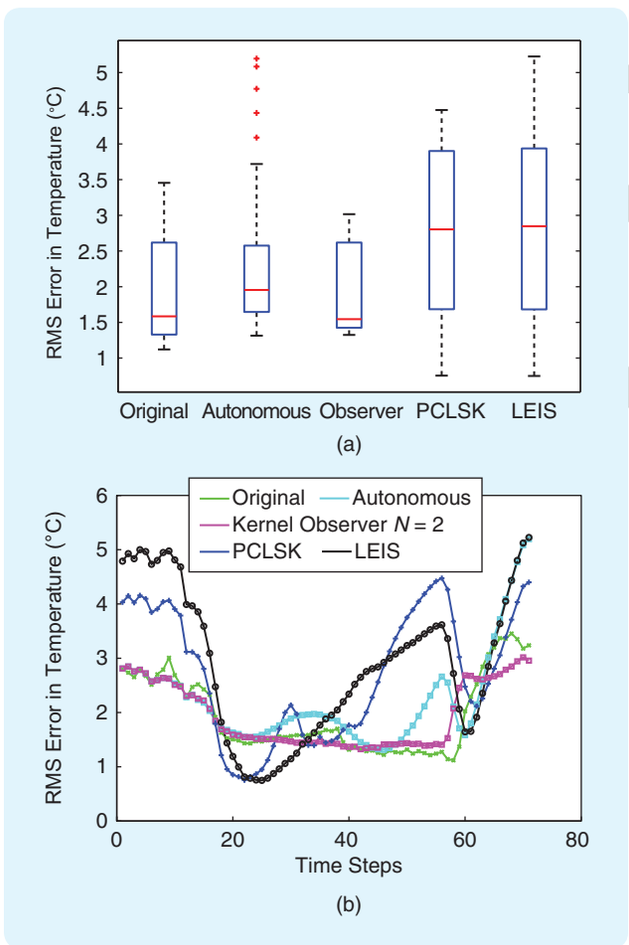

FIGURE 10 A comparison of the kernel observer versus the process convolution with a local smoothing kernel (PCLSK) and the latent extension of the input space (LEIS) on the Intel Berkeley data set. The error as (a) a box plot and (b) a time series. RMS: root mean square.

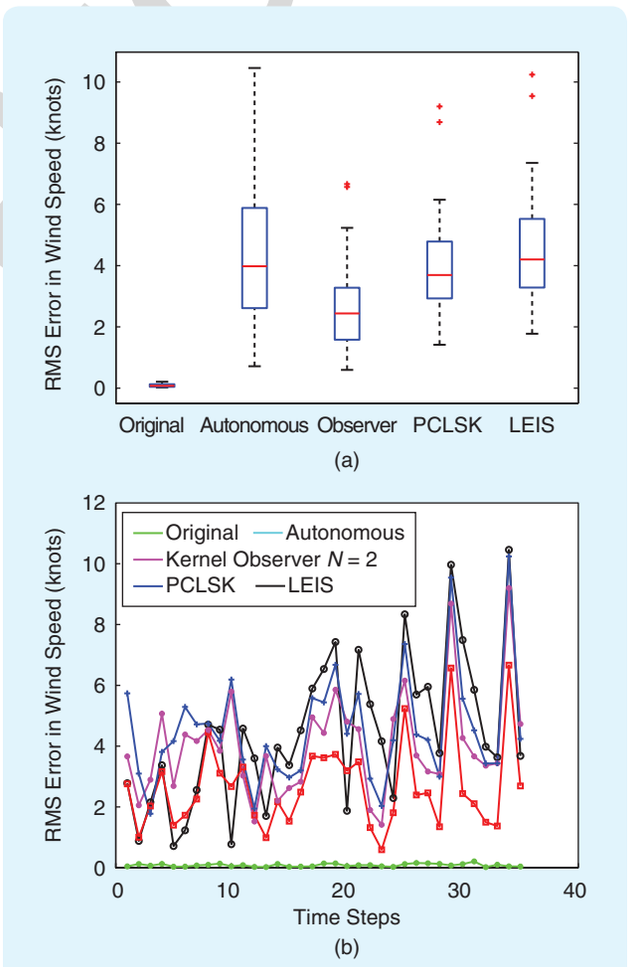

FIGURE 11 A comparison of the kernel observer versus the process convolution with a local smoothing kernel and the latent extension of the input space on the Irish wind data set. The error as (a) a box plot and (b) a time series. RMS: root mean square.

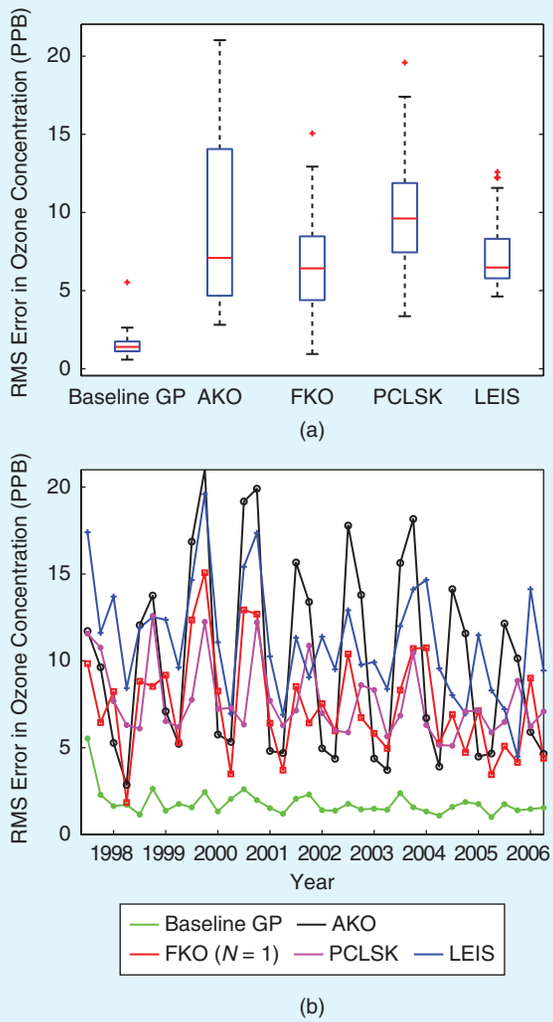

FIGURE 12 A comparison of the kernel observer versus the process convolution with a local smoothing kernel and the latent extension of the input space using the ozone data set. The error as (a) a box plot and (b) a time series. RMS: root mean square; PPB: parts per billion; GP: Gaussian process; AKO: autonomous kernel observer; FKO: feedback kernel observer; PCLSK: process convolution with a local smoothing kernel; LEIS: latent extension of the input space.

TABLE 1 Total training and prediction times for Figures 10 and 11.

	Intel Berkeley	Irish Wind	Ozone
Data size (bases/time steps)	25/72	12/36	30/68
Kernel observer	2.1 s	0.1 s	1.61 s
Process convolution with a local smoothing kernel	121.4 s	7 s	91.9 s
Latent extension of the input space	43.8 s	2.8 s	37.41 s

the 4-km *AVHRR* Pathfinder Project, which monitors the global ocean surface temperature [Figure 13(a) provides the raw data]. This data set is challenging, with measurements at more than 37 million possible coordinates but with only 3–4 million measurements available per day (which means a lot of missing data). The goal was to learn the day and night temperature models by using data from 2011 and then monitor the system through 2012. Success in monitoring would demonstrate two things: 1) the modeling process can capture spatiotemporal trends that generalize across years and 2) the observer framework facilitates inferring the state using a number of measurements that is an order of magnitude fewer than the available total. Note that the data set and the nonstationary kernel methods were not compared due to their size and high computational requirement.

To build the autonomous kernel observer and general kernel observer models, the same procedure outlined in the "Comparison With Nonstationary Kernel Methods on Real-World Data" section was followed but with $C = \{c_1, \ldots, c_M\}$, $c_j \in \mathbb{R}^2$, and $|C| = 300$. The Kalman filter for the general kernel observer model used $N \in \{250, 500, 1000\}$ at random locations to track the system state, given a random initial condition w_0. As a fair baseline, the observers are compared to training a sparse GP model (labeled *original*) on approximately 400,000 measurements per day. Figure 13(b) estimates the global ocean surface temperatures obtained using the autonomous kernel observer.

In Figure 13, (a) and (d) compare the autonomous and feedback approach with 1000 samples to the baseline GP. It is shown that the autonomous method does well in the beginning but then incurs an unacceptable amount of error when the time series goes into 2012 (that is, where the model has not seen any training data), whereas the feedback kernel observer does well throughout. In Figure 13, parts (e) and (f) compare the root mean square error of estimated values from the real data. This figure shows the trend of the observer getting progressively better state estimates as a function of the number of sensing locations N. Note that the performance of training a GP was checked with only 1000 samples as a control. However, the average error was approximately 10 K, that is, much worse than the feedback kernel observer. The time required for using the kernel observer is significantly less than retraining the model at every time step [see Figure 13(g) and (h)].

Weather Anomaly in 2012

The poor performance of the autonomous kernel observer in 2012 was further investigated, as illustrated in Figure 13(c) and (d). Clearly, the prediction error blows up at the start of May 2012, indicating that the autonomous model trained using data from 2011 does well in capturing the annual weather dynamics up to that month. Attention is directed to the weather in May 2012, as changes in ocean temperatures are directly related to the weather. As assumed, severe weather conditions were reported on the east coast of the

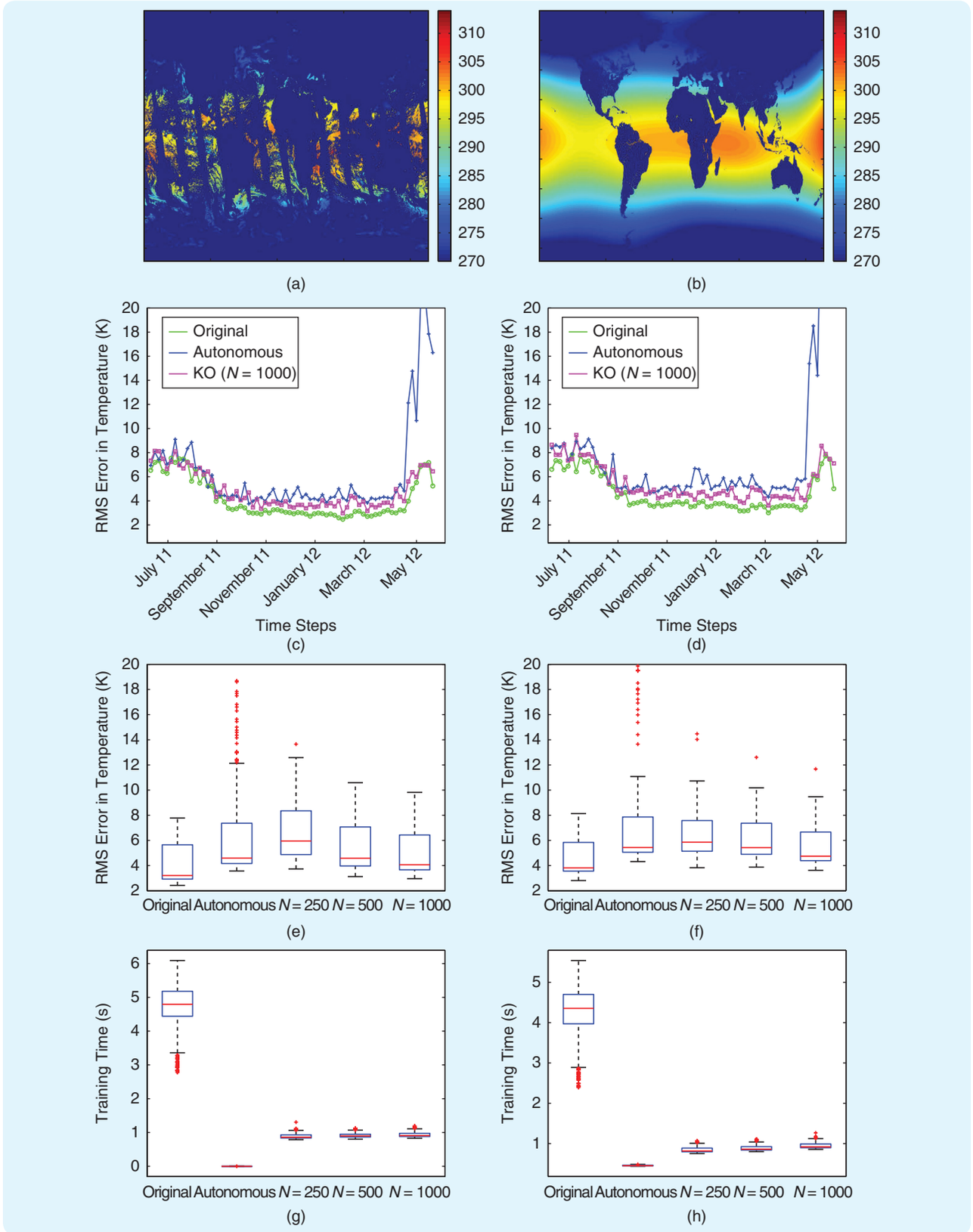

FIGURE 13 The performance of the kernel observer using *Advanced Very-High-Resolution Radiometer* satellite 2012 data with different numbers of observation locations. (a) The raw satellite data. (b) The autonomous kernel observer (KO) estimate. (c) The day error (time series). (d) The night error (time series). (e) The day error (box plot). (f) The night error (box plot). (g) The day estimation time. (h) The night estimation time. RMS: root mean square.

United States, and this anomaly continued until June. The apparent poor performance of the autonomous kernel observer can be a useful indicator in detecting anomalous behavior, as was the case with the ocean temperature in May 2012 (which deviated from the normal weather dynamics observed in 2011, when no severe conditions were reported).

The locations at which the prediction error was two standard deviations above the mean error were further identified. They are plotted in Figure 14. The error locations from May 6 to 7 coincided with the severe weather onset on the east coast, while the locations on May 28 and 29 were along the track of Tropical Storm Beryl. This shows that the autonomous kernel observer model captures the dynamics of spatiotemporal evolution and has the potential to identify current behavior that is anomalous to what is observed in the training set.

Predicting the Evolution of Weed Density in Agricultural Fields

As a proof of concept for applying our methods for learning dynamics and/or inferring the state of spatiotemporally evolving systems, the problem of predicting weed growth in agricultural fields was examined. Past work presented a detailed analysis of weed growth models in which measurements of the seed bank density for various species were conducted [61]–[64]. To generate data for testing our methods, a weed growth simulation model was implemented whose rate of seedling emergence agreed with that found in the research [1]. A Poisson process was utilized to simulate the temporal evolution of emergence events. This assumption is reasonable across the short time scales during which robots may fully weed a field.

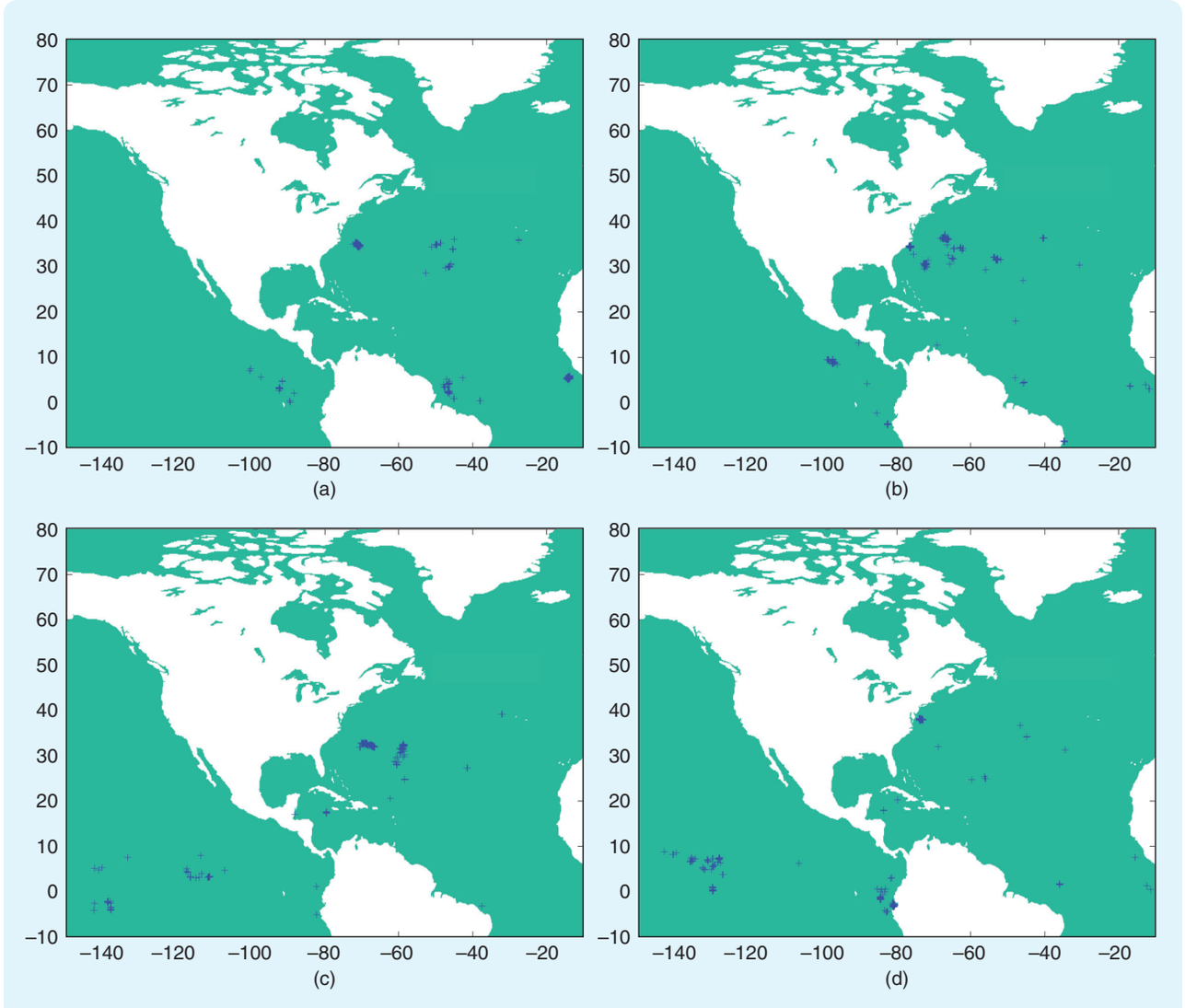

FIGURE 14 Locations of weather anomaly obtained based on the error between the actual temperature and the prediction of the autonomous kernel observer. The landmass is shown in white, and the ocean is in green. Marked locations have an error greater than two standard deviations above the mean error. (a) May 6, 2012. (b) May 7, 2012. (c) May 28, 2012. (d) May 29, 2012.

Other work in the field of crop science [65], [66] has shown that the spatial variation in the seed bank density for some species of weeds may be modeled via the Gini coefficient of concentration (GCC). This model is accurate for common waterhemp (*Amaranthus tuberculatus*), and thus it was found to be useful in this simulation of the spatial distribution of the seed bank density. See [1], [67], and [68] regarding the relationship between seed bank emergence patterns and environmental conditions, such as temperature and moisture. The weed growth simulation (Figure 15) **<AU: Please check that the citation of Figure 15 is appropriate.>** is based on Bernoulli random variables, operating on a matrix of cells (each 0.8 m^2) comprising a gridded field of 0.4 hectares, or a cellular automata model [69].

Seeds emerge from a limited bank, forming a binomial distribution through time. The parameters that are used, which are summarized in Table 2, are aligned with the growth model for common waterhemp determined in [65]. The initial density of the seed bank in each cell is S_0, on average (between 600 and 1560 seeds per cell). However, at the start of the simulation, the seed bank density in each cell $S_0(x, y)$ is chosen so that the GCC between all the cells (used to ensure that the relative density of the weeds aligns with that seen in real experiments) is from 0.31 to 0.35, as experimentally determined in [65]. The field is first divided into 50 patches of weeds, with centers chosen

TABLE 2 Seed bank density parameters.

Parameter	Gini Coefficient of Concentration	S_0 (Seeds/Cell)	Number of Patches	Patch Size (Cells in X and Y)
Range	[0.31,0.35]	[600,1560]	50	[0,20]

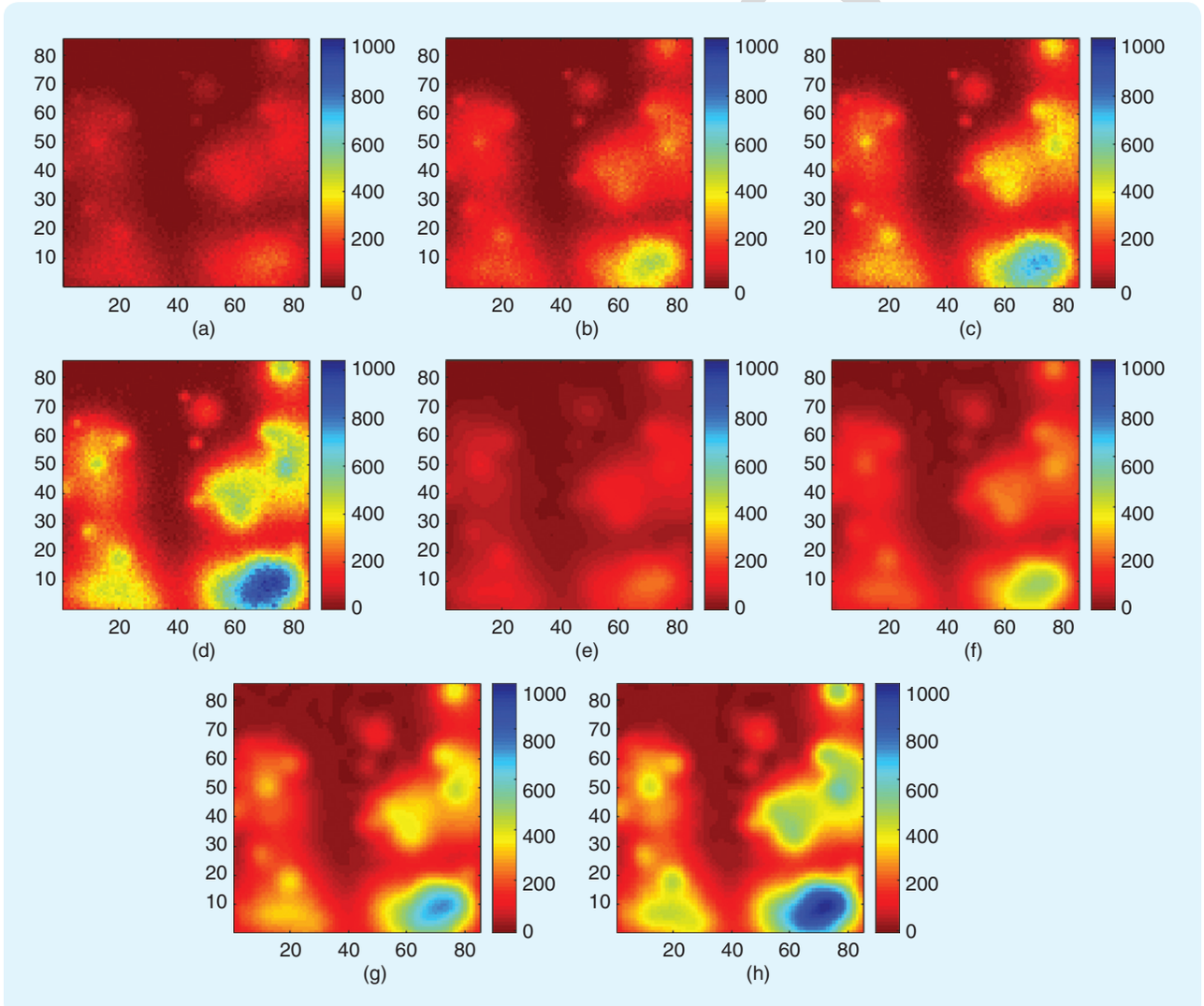

FIGURE 15 A visualization of weed density growth during 80 days. (a)–(d) The original visualization. (e)–(h) The evolving Gaussian process. (a) Snapshot 2. (b) Snapshot 28. (c) Snapshot 54. (d) Snapshot 80. (e) Snapshot 2. (f) Snapshot 28. (g) Snapshot 54. (h) Snapshot 80.

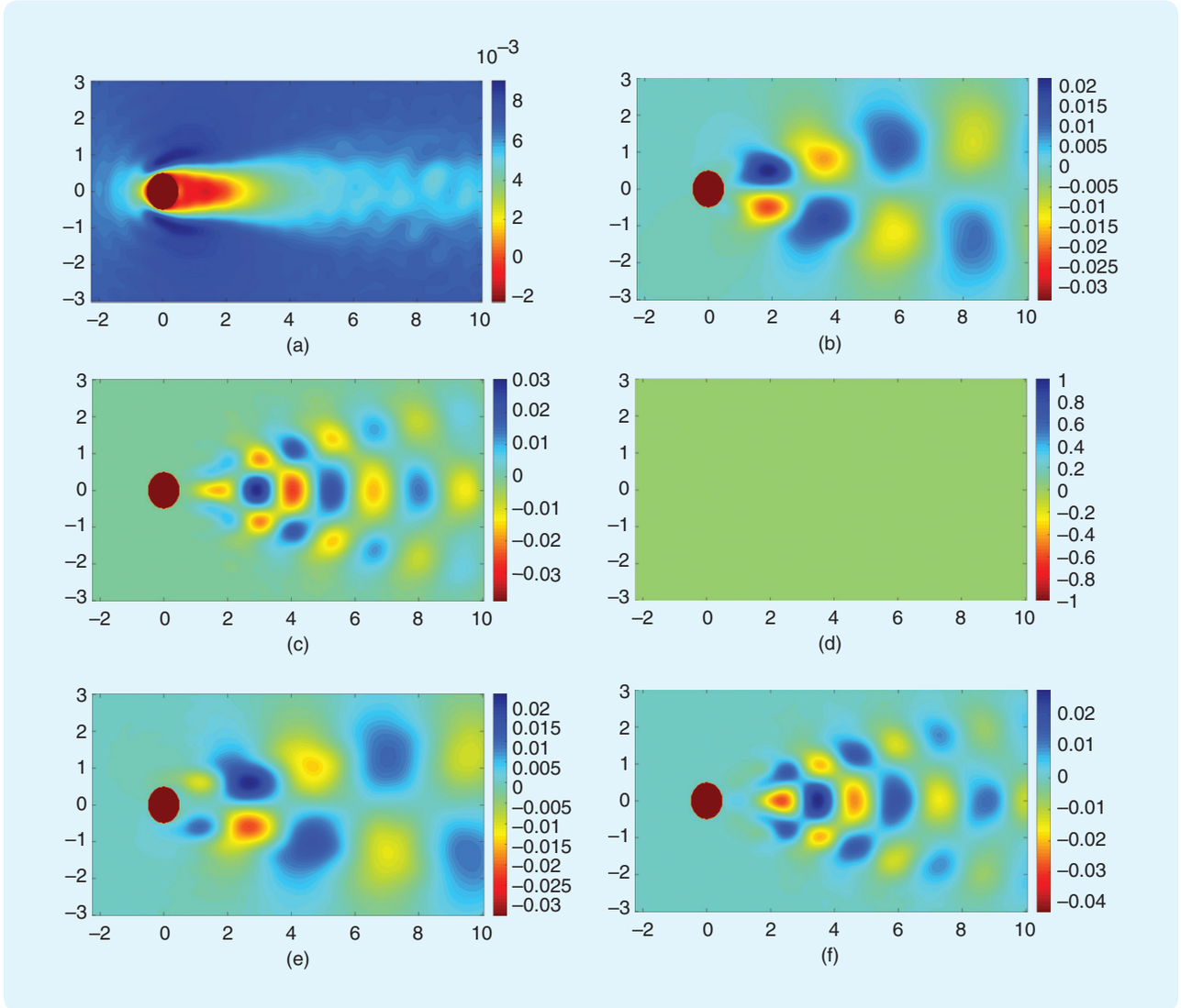

FIGURE 16 The primary Koopman modes of a flow past a cylinder at Reynolds number 100. (a)–(f) The evolving Gaussian process. (g)–(l) The dynamic mode decomposition. (a) 0°, real component. (b) 6°, real component. (c) 12°, real component. (d) 0°, imaginary component. (e) 6°, imaginary component. (f) 12°, imaginary component. (*Contined*)

uniformly at random and with sizes from zero to 20 cells in each dimension. Each cell has an initial density between zero and 20. At the initialization of the simulation, a certain number of days, d_0, is allowed to elapse before weeding starts. The number of emerging weeds in each cell N_{emerge} is a randomly generated Poisson variable with mean $\lambda(x, y, t)$. The weed density in each cell $\zeta(x, y, t)$ rises as seeds emerge from the bank. The maximum weed height at each cell $\delta(x, y, t)$ increases from zero at a fixed rate of Γ in/day.

The main challenge with weeding robots is that they have access only to sparse data about plant density and height and no information about the seed bank. In fact, the robots can sense only the row they are in and potentially the adjacent rows on either side. They have no information about locations in the field that have not been visited. Hence, they must try to predict the global weed density given sparse information. This is a great application for the kernel observer techniques studied in this article. To demonstrate the proof of concept, a kernel observer model was trained using the weed density data generated by our simulations so that robots in a field can quickly globally infer the state of the field from partial measurements based on a few rows. RBF kernels were used, with a bandwidth that was approximately five cells (4.5 m) wide. These were centered throughout the domain by using the algorithm in [70].

After training a GP on each snapshot of the data, a weight vector trajectory was obtained. Linear least squares was used to determine the best linear transition in the weight space for this trajectory. Images were predicted by feeding forward the initial condition through this linear

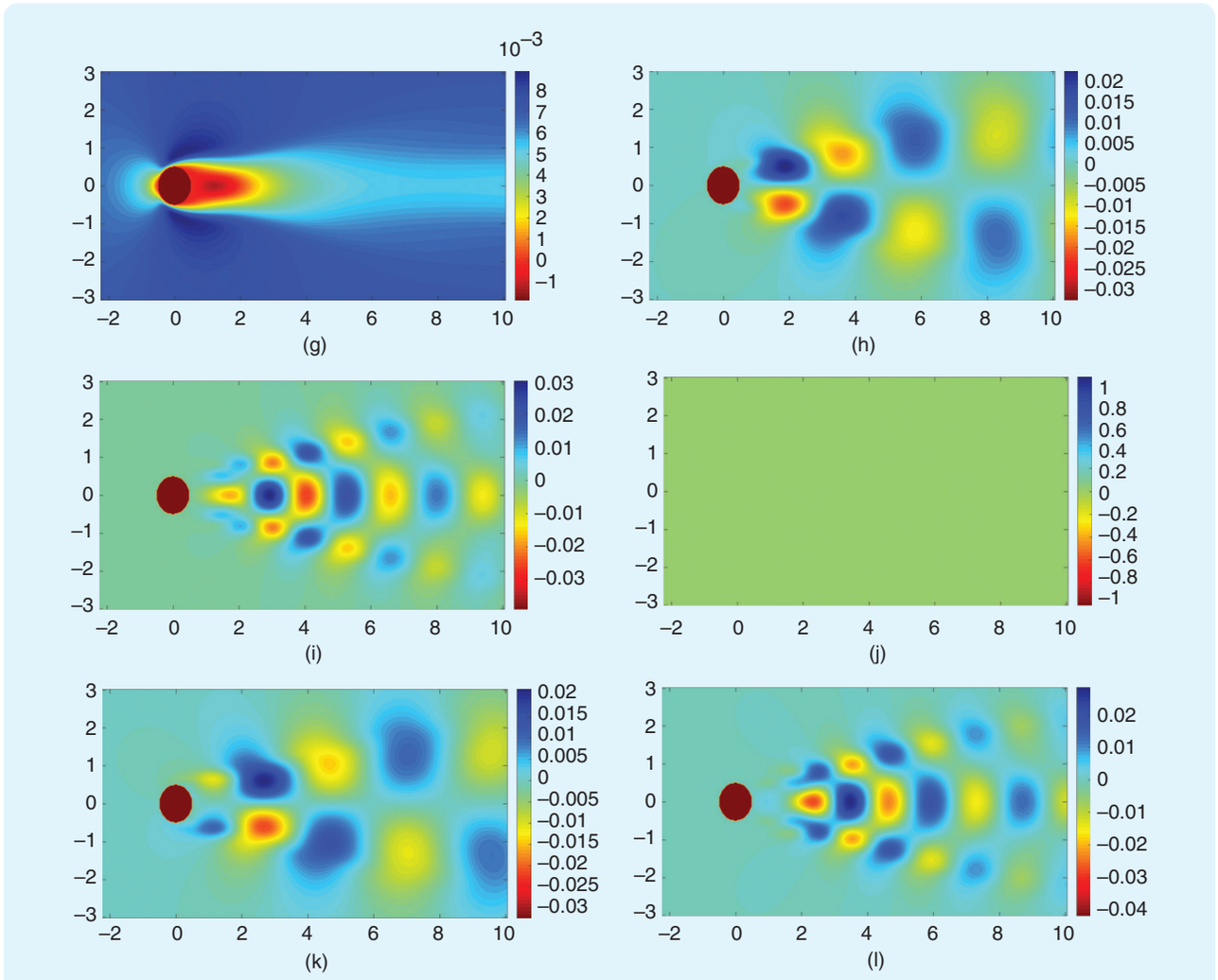

FIGURE 16 (Contined) The primary Koopman modes of a flow past a cylinder at Reynolds number 100. (g) 0°, real component. (h) 6°, real component. (i) 12°, real component. (j) 0°, imaginary component. (k) 6°, imaginary component. (l) 12°, imaginary component.

transition as a comparison with the original data. **<AU: Please check that the preceding edited sentence conveys the intended meaning.>** The model was able to approximate the weed density growth data very well, with the percent error averaging 5%. These results show the feasibility of modeling weed growth using E-GPs. Ongoing work involves a massive field campaign to collect the weed density data required to teach these models. Such data sets are currently not available, and the TerraSentia robots discussed in "Key Control Problems in Agriculture" are being used to collect this data.

Comparison of Eigenvalues and Koopman Modes With Dynamic Mode Decomposition

To empirically verify our theoretical results, the eigenvalues and Koopman modes derived from $\hat{A}$ in this E-GP model are compared with the eigenvalues and Koopman modes derived by the well-known DMD algorithm, using data from fluid flowing past a cylinder at different Reynolds numbers. The results are presented in Figures 16 and 17. As can be seen, the modes generated by the E-GP match those generated by DMD at least as well as the E-GP predictions match the original data.

CONCLUSION

Machine learning techniques, GP regression, and deep learning are providing increasingly more powerful ways of learning predictive models. Since these techniques are limited by the data sets that they are trained with, their predictions are not always reliable when forecasting real-world, complex spatiotemporally evolving phenomena that have high variability. Years of weather data cannot be a reliable predictor of the current weather, nor can data from one farm directly relate to another. Because of this, when designing complex cyberphysical systems, such as teams of mechanically weeding robots, engineers have devised ways to supplement machine learning model predictions with real measurements. This is not always immediate

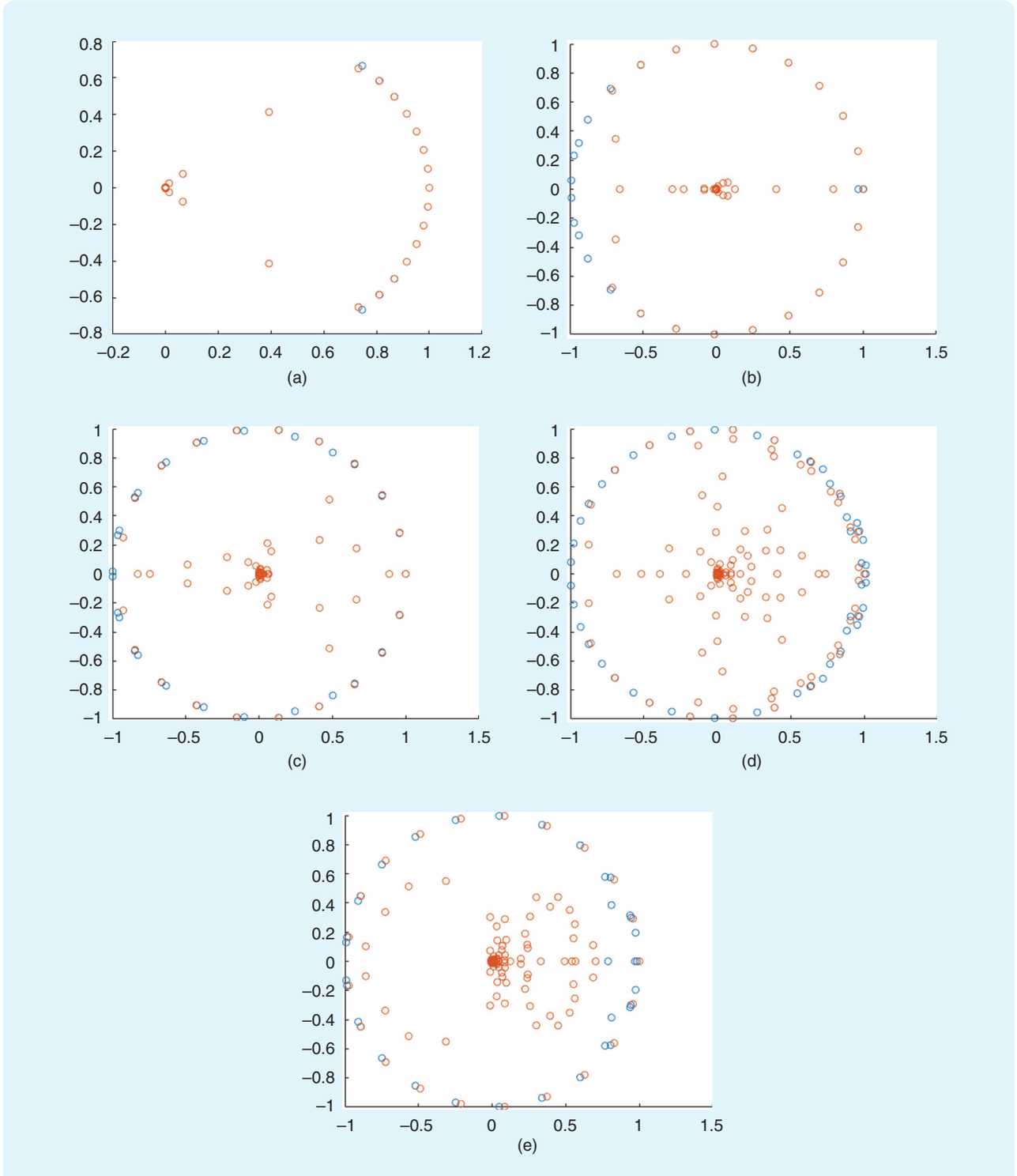

FIGURE 17 An eigenvalue comparison between the evolving Gaussian process and the dynamic mode decomposition at different Reynolds numbers. (a) Re = 100. (b) Re = 300. (c) Re = 600. (d) Re = 800. (e) Re = 1000.

when one works in the abstract feature spaces utilized in machine learning models.

This tutorial presented kernel observers and their extension, E-GPs, which are formed by embedding a linear dynamical system in the RKHSs generated by the GP model. The parameters of the linear dynamical system can be trained to approximately predict the evolution. This leads to powerful and generalizable models, as

demonstrated by results for predicting solutions to the Navier–Stokes equations. When supplemented with a predict-and-correct strategy (such as a Kalman filter embedded with the linear dynamical system in the RKHS), the dynamical state of the system can be kept on track with real sensor measurements. This creates a very powerful prediction system that is capable of estimating nonlinear evolution through minimal sensor measurements.

It was shown that the geometric properties of the linear dynamical system can be utilized to determine sensor numbers and locations. Looking forward, it would be interesting to extend the idea to include nonlinear dynamical systems in the feature space, which would improve the capability of the E-GP model. At present, the results for fairly complex data sets demonstrate that feedback with sensors creates a robust monitoring system with a simple linear model. The theoretical limits of the tradeoff between improving the model versus increasing the sensing in the context of E-GPs is also an interesting avenue of study. Finally, note that deep neural networks have also been applied to the spatiotemporal modeling problem with significant empirical success [71]. However, because these methods are 1) nonlinear in their parameters and 2) seem to lack the spatial encoding properties of some types of kernels, generalizing the analysis considered here for those systems is an interesting open question. As "Feature Spaces in Machine Learning" indicates, there could be very interesting future work that generalizes the idea of embedding controllers and observers in the advanced feature spaces considered in machine learning models. **<AU: Please note that the "Software" section repeated information presented in the "Article Outline and Relationship to Prior Work by the Authors" section and was deleted to avoid repetition.>**

ACKNOWLEDGMENTS

The work presented in this article was supported in part by the U.S. Air Force Office of Scientific Research under grant FA9550-15-1-0146 and by the United States Department of Agriculture/National Science Foundation under cyberphysical systems grant 2018-67007-28379, and National Robotics Initiative grant 2019-67021-28989. We thank EarthSense (www.earthsense.co) for providing approval to use Figures 1 and S2. We also thank Prof. Stephen Long, Prof. Carl Bernacchi, Prof. Michael Gore, and Prof. Edward Buckler for insight about the phenotyping bottleneck and simulation of plants (crops in silico); Prof. Adam Davis for input regarding the herbicide-resistant weed crisis; and Prof. Sarah Lovell and Dr. Chinmay Soman for input concerning sustainable agricultural production systems and perennial polycultures. We also thank the members of the University of Illinois at Urbana-Champaign Center for Digital Agriculture for their input.

AUTHOR INFORMATION

Joshua E. Whitman received the B.S. degree in mechanical engineering and mathematics from Oklahoma State University in 2015 and the M.S. degree in mechanical engineering from the University of Illinois in 2018. He is a Ph.D. student at the University of Illinois at Urbana-Champaign. His research interests include the fields of machine learning, controls, and autonomy, and his work focuses on developing new methods for learning, monitoring, and controlling the dynamics of complex spatiotemporally evolving systems.

Harshal Maske received the B.S. and M.S. integrated degree in mechanical engineering from the Indian Institute of Technology Kharagpur in 2009 and the Ph.D. degree in 2018 from the University of Illinois at Urbana-Champaign. He is a research engineer at Ford. Prior to joining the company, he was a research intern at Mitsubishi Electric Research Laboratory. He worked for three years at India's Defense Research and Development Organization (2009–2012) and for one year at Deere & Company (2012–2013).

Hassan A. Kingravi received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology in 2014 and was a postdoctoral researcher in Prof. Girish Chowdhary's group at Oklahoma State University. He is a senior data scientist at MailChimp, where he conducts research on machine learning and builds systems for fighting fraud. His research interests include the interplay of control theory, signal processing, and machine learning for spatiotemporal data.

Girish Chowdhary (girishc@illinois.edu) received the Ph.D. degree in aerospace engineering in 2010 from Georgia Institute of Technology. He is an assistant professor and Donald Biggar Willet Faculty Fellow at the University of Illinois at Urbana-Champaign (UIUC), where he is affiliated with the Departments of Electrical and Computer Engineering, Agricultural and Biological Engineering, Computer Science, and Aerospace Engineering. He is a member of the UIUC Coordinated Science Laboratory and director of the UIUC Distributed Autonomous Systems Laboratory and the Field Robotics Engineering and Science Hub. He was a postdoctoral researcher at the Massachusetts Institute of Technology Laboratory for Information and Decision Systems from 2011 to 2013 and an assistant professor in the Department of Mechanical and Aerospace Engineering, Oklahoma State University, from 2013 to 2016. Prior to joining Georgia Tech, he worked with the German Aerospace Center Institute of Flight Systems from 2003 to 2006. His research interests include theoretical insights and practical algorithms for adaptive autonomy, with applications in field robotics.

REFERENCES

[1] W. McAllistar, D. Osipychev, G. Chowdhary, and A. Davis, "Multi-agent planning for coordinated robotic weed killing," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, 2018. doi: 10.1109/IROS.2018.8593429. **<AU: Please provide the page range.>**

[2] T. P. Barnett, D. W. Pierce, and R. Schnur, "Detection of anthropogenic climate change in the world's oceans," *Science*, vol. 292, no. 5515, pp. 270–274, 2001. doi: 10.1126/science.1058304.

[3] M. J. Heaton et al., "Spatio-temporal models for large-scale indicators of extreme weather," *Environmetrics*, vol. 22, no. 3, pp. 294–303, 2011. doi: 10.1002/env.1050.
[4] N. Cressie and C. K. Wikle, *Statistics for Spatio-Temporal Data*. Hoboken, NJ: Wiley, 2011.
[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, Dec. 2005.
[6] K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–202, 2001. doi: 10.1109/72.914517.
[7] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
[8] S Garg, A Singh, and F Ramos, "Learning non-stationary space-time models for environmental monitoring," in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, Canada, 2012. **<AU: Please provide the page range.>**
[9] C. Ma, "Nonstationary covariance functions that model space–time interactions," *Statist. Probab. Lett.*, vol. 61, no. 4, pp. 411–419, 2003. doi: 10.1016/S0167-7152(02)00401-7.
[10] 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*, W. Daelemans, B. Goethals, and K. Morik, Eds. New York: Springer-Verlag, 2008, pp. 204–219. **<AU: Please confirm the editor names.>**
[11] M. Todescato, A. Carron, R. Carli, G. Pillonetto, and L. Schenato, "Efficient spatio-temporal Gaussian regression via Kalman filtering," 2017, arXiv:1705.01485.
[12] G. Chowdhary, H. Kingravi, J. P. How, and P. Vela, "Nonparametric adaptive control using Gaussian processes," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2013, pp. 861–867. doi: 10.1109/CDC.2013.6759990.
[13] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*. Cambridge, MA: MIT Press, 2012.
[14] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on Riemannian manifolds with Gaussian RBF kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2464–2477, 2015. doi: 10.1109/TPAMI.2015.2414422.
[15] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
[16] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The Kriged Kalman filter," *Test*, vol. 7, no. 2, pp. 217–282, 1998. doi: 10.1007/BF02565111.
[17] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bull. Amer. Math. Soc.*, vol. 39, no. 1, pp. 1–49, 2002. doi: 10.1090/S0273-0979-01-00923-5.
[18] H. A. Kingravi, G. Chowdhary, P. A. Vela, and E. N. Johnson, "Reproducing kernel Hilbert space approach for the online update of radial bases in neuro-adaptive control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1130–1141, 2012. doi: 10.1109/TNNLS.2012.2198889.
[19] M. Liu, G. Chowdhary, B. Castra da Silva, S. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Syst. Mag.*, vol. 38, no. 5, pp. 53–86, Oct. 2018. doi: 10.1109/MCS.2018.2851010.
[20] C. E. Rasmussen, "Gaussian processes for machine learning," Max Planck Institute for Biological Cybernetics, Tech. Rep., 2006. **<AU: Please provide the report number or URL and location for the associated organization.>**
[21] H Kingravi, H Maske, and G Chowdhary, "Kernel observers: Systems theoretic modeling and inference of spatiotemporally varying processes," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, Barcelona, Spain, 2016, pp. 3990–3998.
[22] J. Whitman, H. Maske, G. Chowdhary, H. Kingravi, C. Lu, and B. Jayaraman, "Modeling nonlinear dynamical fluid flows with evolving Gaussian process models," in *Proc. NIPS Workshop Mach. Learn. Wild*, Barcelona, Spain, 2016. **<AU: Please provide the page range.>**
[23] H. A. Kingravi, H. Maske, and G. Chowdhary, "Kernel controllers: A systems-theoretic approach for data-driven modeling and control of spatiotemporally evolving processes," 2015, arXiv:abs/1508.02086.
[24] J. Whitman and G. Chowdhary, "Learning dynamics across similar spatiotemporally-evolving physical systems," in *Proc. Conf. Robot Learn.*, 2017, pp. 472–481.
[25] H. Maske, H. Kingravi, and G. Chowdhary, "Sensor selection via observability analysis in feature space," in *Proc. Amer. Control Conf.*, to be published. **<AU: Please provide updated information for this reference.>**
[26] C. K. Wikle, "A kernel-based spectral model for non-Gaussian spatio-temporal processes," *Statist. Model.*, vol. 2, no. 4, pp. 299–314, 2002. doi: 10.1191/1471082x02st036oa.
[27] J. R. Stroud, P. Muller, and B. Sanso, "Dynamic models for spatiotemporal data," *J. Roy. Statist. Soc., Ser. B*, vol. 63, no. 4, pp. 673–689, 2001. doi: 10.1111/1467-9868.00305.
[28] J. Hartikainen, "Sequential inference for latent temporal Gaussian process models," Doctoral dissertation, Dept. Biomed. Eng. Comput. Sci., Aalto Univ., Espoo, Finland, 2013.
[29] F. Lindgren, H. Rue, and J. Lindstrom, "An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach," *J. Roy. Statist. Soc., Ser. B*, vol. 73, no. 4, pp. 423–498, 2011. doi: 10.1111/j.1467-9868.2011.00777.x.
[30] T T Ho, P W Fieguth, and A S Willsky, "Multiresolution stochastic models for the efficient solution of large-scale space-time estimation problems," in *Proc. 1996 IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 6, pp. 3097–3100. doi: 10.1109/ICASSP.1996.550531.
[31] F. Pérez-Cruz, S. V. Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 40–50, 2013. doi: 10.1109/MSP.2013.2250352.
[32] S Särkkä and R Piché, "On convergence and accuracy of state-space approximations of squared exponential covariance functions," in *Proc. 2014 IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, pp. 1–6. doi: 10.1109/MLSP.2014.6958890.
[33] D. Higdon, "A process-convolution approach to modelling temperatures in the North Atlantic Ocean," *Environ. Ecol. Statist.*, vol. 5, no. 2, pp. 173–190, 1998. doi: 10.1023/A:1009666805688.
[34] C. Paciorek and M. Schervish, "Nonstationary covariance functions for Gaussian process regression," *Adv. Neural Inform. Process. Syst.*, vol. 16, pp. 273–280, 2004. **<AU: Please provide the issue number or a month.>**
[35] A Singh, F Ramos, H Durrant-Whyte, and W J Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2010, pp. 5490–5497. doi: 10.1109/ROBOT.2010.5509934.
[36] A. M. Schmidt and A. O'Hagan, "Bayesian inference for non-stationary spatial covariance structure via spatial deformations," *J. Roy. Statist. Soc., Ser. B*, vol. 65, no. 3, pp. 743–758, 2003. doi: 10.1111/1467-9868.00413.
[37] T. Pfingsten, M. Kuss, and C. E. Rasmussen. "Nonstationary Gaussian process regression using a latent extension of the input space," 2006. [Online]. Available: http://www. kyb. mpg. de/˜ tpfingst **<AU: Please update the URL link.>**
[38] J. Noh and V. Solo, "Testing for space-time separability in functional MRI," in *Proc. 2007 4th IEEE Int. Symp. Biomed. Imaging, From Nano to Macro*, pp. 412–415. doi: 10.1109/ISBI.2007.356876.
[39] J. Noh and V. Solo, "Space-time separability in FMRI: Asymptotic power analysis and Cramér-Rao lower bounds," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 148–153, 2012. doi: 10.1109/TSP.2012.2226168.
[40] A Carron, M Todescato, R Carli, L Schenato, and G Pillonetto, "Machine learning meets Kalman filtering," in *Proc. 2016 IEEE 55th Conf. Decis. Control (CDC)*, pp. 4594–4599. doi: 10.1109/CDC.2016.7798968.
[41] J Hartikainen and S Särkkä, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *Proc. 2010 IEEE Int. Workshop Mach. Learn. Signal Process.*, pp. 379–384. doi: 10.1109/MLSP.2010.5589113.
[42] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 51–61, 2013. doi: 10.1109/MSP.2013.2246292.
[43] R. N. Miller, "Toward the application of the Kalman filter to regional open ocean modeling," *J. Phys. Oceanogr.*, vol. 16, no. 1, pp. 72–86, 1986. doi: 10.1175/1520-0485(1986)016<0072:TTAOTK>2.0.CO;2.
[44] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton Univ. Press, 2010.
[45] C. Guestrin, A. Krause, and A. Singh, "Near-optimal sensor placements in Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 265–272. doi: 10.1145/1102351.1102385.
[46] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, July 2010. doi: 10.1017/S0022112010001217.
[47] C. W. Rowley, M. O. Williams, and I. G. Kevrekidis, "A kernel-based method for data-driven Koopman spectral analysis," *J. Comput. Dyn.*, vol. 2, no. 2, pp. 247–265, 2015. doi: 10.3934/jcd.2015005.
[48] S. L. Brunton, J. L. Proctor, and J. Nathan Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical sys-

tems," *Proc. Nat. Acad. Sci.*, vol. 113, no. 15, pp. 3932–3937, 2016. doi: 10.1073/pnas.1517384113.
[49] H. Brezis, *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. New York: Springer-Verlag, 2010.
[50] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, pp. 1177–1184.
[51] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inform. Process. Syst.*, 2001, pp. 682–688.
[52] K. Zhou, J. C. Doyle, and, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall, 1996.
[53] W. Murray Wonham, *Linear Multivariable Control*. New York: Springer-Verlag, 1974.
[54] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," in *Approximation Theory and Spline Functions*, The Netherlands: Springer-Verlag, 1984, pp. 143–145. **<AU: Please provide the editor names.>**
[55] R. Motwani and P. Raghavan, *Randomized Algorithms*. London: Chapman & Hall, 2010.
[56] L. Csatö and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002. doi: 10.1162/089976602317250933.
[57] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annu. Rev. Fluid Mech.*, vol. 45, no. 1, pp. 357–378, 2013. doi: 10.1146/annurev-fluid-011212-140652.
[58] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *J. Fluid Mech.*, vol. 6[illegible]p. 115–127, 2009. doi: 10.1017/S0022112009992059. **<AU: Please provide the issue number or a month.>**
[59] J. Berkovitz, "Action at a distance in quantum mechanics," in *Stanford Encyclopedia of Philosophy*. Stanford, C[illegible]Stanford University, Jan. 26, 2007. [Online]. Available: https://plato.stanford.edu/entries/qm-action-distance/ **<AU: Please confirm added URL and publisher name and location. Also, please provide the date of access.>**
[60] L. Li, X. Zhang, and R. Piltner, "A spatiotemporal database for ozone in the conterminous U.S.," in *Proc. 13th Int. Symp. Temporal Representation Reasoning (TIME'06)*, 2006, pp. 168–176. doi: 10.1109/TIME.2006.3.
[61] D. Nordby, R. Hartzler, and K. Bradley. "Biology and management of waterhemp," Glyphosate, Weeds, and Crops series, Purdue University Extension, West Lafayette, IN, publication GWC-13.12, 2007. [Online]. Available: https://extension.missouri.edu/media/wysiwyg/Extensiondata/Pub/pdf/miscpubs/mx1137.pdf **<AU: Please confirm the location of the organization. Also, please provide the date of access.>**
[62] B. J. Schutte and A. S. Davis, "Do common waterhemp (Amaranthus rudis) seedling emergence patterns meet criteria for herbicide resistance simulation modeling?" *Weed Technol.*, vol. 28, no. 2, pp. 408–417, 2014. doi: 10.1614/WT-D-13-00139.1.
[63] B. A. Sellers, R. J. Smeda, W. G. Johnson, J. Andrew Kendig, and M. R. Ellersieck, "Comparative growth of six Amaranthus species in Missouri," *Weed Sci.*, vol. 51, no. 3, pp. 329–333, 2003. doi: 10.1614/0043-1745(2003)051[0329:CGOSAS]2.0.CO;2.
[64] M. J. Horak and T. M. Loughin, "Growth analysis of four Amaranthus species," *Weed Sci.*, vol. 48, no. 3, pp. 347–355, 2000. doi: 10.1614/0043-1745(2000)048[0347:GAOFAS]2.0.CO;2.
[65] D. Mulugeta and D. E. Stoltenberg, "Seed bank characterization and emergence of a weed community in a moldboard plow system," *Weed Sci.*, vol. 45, no. 1, pp. 54–60, 1997. doi: 10.1017/S000317450009247X.
[66] D. Mulugeta and C. M. Boerboom, "Seasonal abundance and spatial pattern of setaria faberi, chenopodium album, and abutilon theophrasti in reduced-tillage soybeans," *Weed Sci.*, vol. 47, no. 1, pp. 95–106, 1999. doi: 10.1017/S0043174500090718.
[67] A. S. Davis et al., "Seed burial physical environment explains departures from regional hydrothermal model of giant ragweed (Ambrosia trifida) seedling emergence in US Midwest," *Weed Sci.*, vol. 61, no. 3, pp. 415–421, 2013. doi: 10.1614/WS-D-12-00139.1.
[68] R. Werle, L. D. Sandell, D. D. Buhler, R. G. Hartzler, and J. L. Lindquist, "Predicting emergence of 23 summer annual weed species," *Weed Sci.*, vol. 62, no. 2, pp. 267–279, 2014. doi: 10.1614/WS-D-13-00116.1.
[69] B. Chopard and M. Droz, *Cellular Automata*. New York: Springer-Verlag, 1998.
[70] L. Csato and M. Opper, "Sparse representation for Gaussian process models," in *Proc. Adv. Neural Inform. Process. Syst.*, 2001, pp. 444–450.
[71] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 4489–4497.

Machine learning techniques provide a way to capture complex spatiotemporal phenomena that are not easily modeled by first principles alone.

Kernel methods have been applied to spatiotemporal regression problems with varying degrees of success.

The state of research into data-driven generalizing across similar systems with varying parameters is, at best, preliminary.

Functions with complex dynamics can be recovered with fewer sensor placements than functions that have simpler dynamics.

Koopman modes of observables are of interest because they are akin to the eigenvector expansions utilized in linear dynamics.

The main challenge with weeding robots is that they have access only to sparse data about plant density and height and no information about the seed bank.

Years of weather data cannot be a reliable predictor of the current weather, nor can data from one farm directly relate to another.

Geometric properties of the linear dynamical system can be utilized to determine sensor numbers and locations.