PNAS RESEARCH ARTICLE | APPLIED MATHEMATICS OPEN ACCESS

Codiscovering graphical structure and functional relationships within data: A Gaussian Process framework for connecting the dots

Théo Bourdais[a], Pau Batlle[a], Xianjin Yang[a], Ricardo Baptista[a], Nicolas Rouquette[b], and Houman Owhadi[a,1]

Affiliations are included on p. 8.

Edited by Linda Petzold, University of California, Santa Barbara, CA; received February 22, 2024; accepted June 26, 2024

Most problems within and beyond the scientific domain can be framed into one of the following three levels of complexity of function approximation. Type 1: Approximate an unknown function given input/output data. Type 2: Consider a collection of variables and functions, some of which are unknown, indexed by the nodes and hyperedges of a hypergraph (a generalized graph where edges can connect more than two vertices). Given partial observations of the variables of the hypergraph (satisfying the functional dependencies imposed by its structure), approximate all the unobserved variables and unknown functions. Type 3: Expanding on Type 2, if the hypergraph structure itself is unknown, use partial observations of the variables of the hypergraph to discover its structure and approximate its unknown functions. These hypergraphs offer a natural platform for organizing, communicating, and processing computational knowledge. While most scientific problems can be framed as the data-driven discovery of unknown functions in a computational hypergraph whose structure is known (Type 2), many require the data-driven discovery of the structure (connectivity) of the hypergraph itself (Type 3). We introduce an interpretable Gaussian Process (GP) framework for such (Type 3) problems that does not require randomization of the data, access to or control over its sampling, or sparsity of the unknown functions in a known or learned basis. Its polynomial complexity, which contrasts sharply with the super-exponential complexity of causal inference methods, is enabled by the nonlinear ANOVA capabilities of GPs used as a sensing mechanism.

Gaussian Process | Analysis of variance | hypergraph discovery | raw data analysis | functional relationships

The three levels of complexity of function approximation. As illustrated in Fig. 1 *A–C*, Type 1, Type 2, and Type 3 problems can be formulated as completing or discovering hypergraphs where nodes represent variables and edges represent functional dependencies. The graph in Type 1 has only two variables and one unknown function. The graph in Type 2 has multiple variables and (some possibly unknown) functions, and the connectivity of the graph is known. The graph in Type 3 has an unknown connectivity (functional dependencies between variables may be unknown), and this is the focus of this work. Current methods for solving Type 1 and 2 problems include deep learning (DL) methods, which benefit from extensive hardware and software support but have limited guarantees. Despite their prevalence, Type 3 challenges have been largely overlooked due to their inherent complexity. Causal inference methods (1, 2), probabilistic graphs (3, 4), and sparse regression methods (5, 6) offer potential avenues for addressing Type 3 problems. However, it is important to note that their application to these problems necessitates additional assumptions. Causal inference models, for instance, typically assume randomized data and some level of access to the data generation process or its underlying distributions. Sparse regression methods, on the other hand, rely on the assumption that functional dependencies have a sparse representation within a known basis. In this paper, we do not impose these assumptions, and thus, these particular techniques may not be applicable. Furthermore, while the complexity of Bayesian causal inference methods may grow super-exponentially with the number d of variables, the complexity of our method is that of d parallel computations of polynomial complexities bounded between $\mathcal{O}(d)$ (best case) and $\mathcal{O}(d^4)$ (worst case).

Generalizing Gaussian Process Methods. Although Gaussian process (GP) methods are sometimes perceived as a well-founded but old technology limited to curve fitting (Type 1 problems), they have recently been generalized, beyond Type 1 problems,

Significance

Many complex data analysis problems within and beyond the scientific domain involve discovering graphical structures and functional relationships within data. Nonlinear variance decomposition with Gaussian Processes simplifies and automates this process. Other methods, such as artificial neural networks, lack this variance decomposition feature. Information-theoretic and causal inference methods suffer from super-exponential complexity with respect to the number of variables. The proposed technique performs this task in polynomial complexity. This unlocks the potential for applications involving the identification of a network of hidden relationships between variables without a parameterized model at a remarkable scale, scope, and complexity.

Author contributions: T.B., P.B., X.Y., R.B., N.R., and H.O. designed research; performed research; analyzed data; and wrote the paper.

Competing interest statement: Caltech has filed for a provisional patent application.

This article is a PNAS Direct Submission.

[1] To whom correspondence may be addressed. Email: owhadi@caltech.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2403449121/-/DCSupplemental.

Published August 1, 2024.

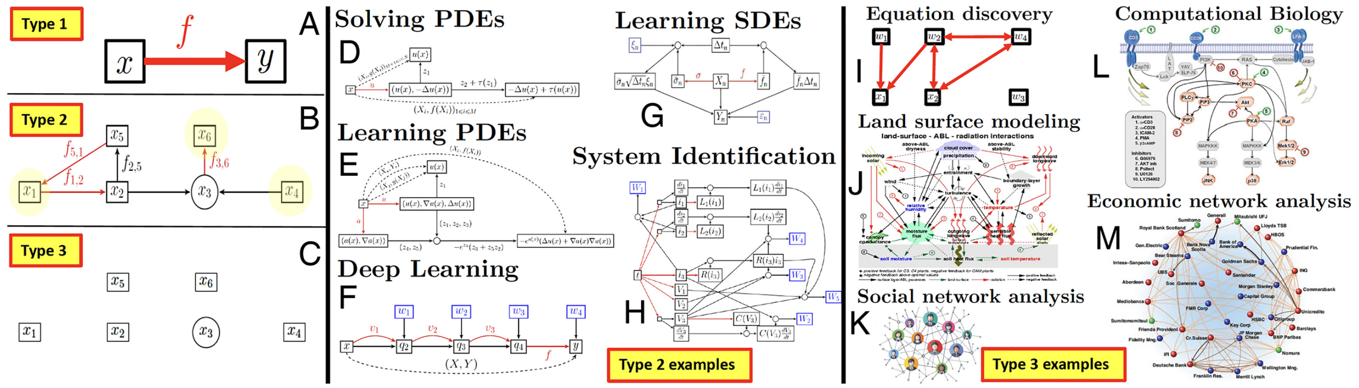

Fig. 1. (*A–C*) The three levels of complexity of function approximation. *A*: Type 1 (regression). *B*: Type 2 (computational graph completion). *C*: Type 3 (computational hypergraph discovery). (*D–H*) Examples of Type 2 problems. *D*: Solving PDEs. *E*: Learning PDEs. *F*: Deep learning. *G*: Learning SDEs. *H*: System identification. (*I–M*) Examples of Type 3 problems. *I*: Equation discovery. *J*: Land surface modeling. *K*: Social network analysis. *L*: Computational biology. *M*: Economic network analysis. Image credit: *D*, Reprinted from ref. 12, with permission from Elsevier; *E-H*, Reprinted from ref. 7, with permission from Springer Nature; *J*, Reprinted from ref. 26, with permission from Elsevier; *K*, Getty/Ani_Ka; *L*, Reprinted from ref. 29, with permission from the American Association for the Advancement of Science; and *M*, Reprinted from ref. 28, with permission from the American Association for the Advancement of Science.

to an interpretable framework [Computational Graph Completion or CGC (7)] for solving Type 2 problems (8–13), all while maintaining the simple and transparent theoretical and computational guarantees of kernel/optimal recovery methods (14, 15). This paper introduces a comprehensive GP framework for solving Type 3 problems, which is interpretable and amenable to analysis. This framework leverages the uncertainty quantification (UQ) properties of GP methods, which do not have an immediate natural counterpart in DL methods. It is based on a kernel generalization (16, 17) of variance-based sensitivity analysis guiding the discovery of the structure of the hypergraph. Here, variables are linked via GPs, and those contributing to the highest data variance unveil the hypergraph's structure. This GP variance decomposition of the data leads to signal-to-noise and a Z-score that can be employed to determine whether a given variable can be approximated as a nonlinear function of a subset of other variables.

The Scope of Type 1, 2, and 3 Problems. "Civilization advances by extending the number of important operations we can perform without thinking about them" (18). In line with this perspective the scope of Type 1, 2, and 3 problems is immense. Numerical approximation (15, 19–21), Supervised Learning, and Operator Learning (22–25) can all be formulated as Type 1 problems, i.e., as approximating unknown functions given (possibly with noisy and infinite/high-dimensional) inputs/output data. The common GP-based solution to these problems is to replace the underlying unknown function by a GP and compute its MAP estimator given available data. Type 2 problems include (Fig. 1 *D–H*) solving and learning (possibly stochastic) ordinary or partial differential equations (9, 12), Deep Learning (8), dimension reduction, reduced-ordered modeling, system identification (7), closure modeling, etc. Indeed, all these problems can be formulated as completing a computational graph (7). In this formulation, variables and functions are represented by the nodes and the edges of the graph whose structure corresponds to the functional dependencies between variables. Some of the functions and variables may be unknown, and by completing, we mean approximating the unknown functions (colored in red in Fig. 1) given samples from the observed variables. The common GP-based solution to Type 2 problems is to simply replace unknown functions by GPs and compute their MAP/MLE estimators given available data and constraints imposed by the structure of the graph (7). While most problems in Computational Sciences and Engineering (CSE) and Scientific Machine Learning (SciML) can be framed as Type 1 and Type 2 challenges, many problems in science can only be categorized as Type 3 problems, i.e., discovering the structure/connectivity of the graph itself from data prior to its completion. Indeed the scope of Type 3 problems extends well beyond Type 2 problems and includes equation discovery (Fig. 1*I*); the modeling of land surface interactions in weather prediction (Fig. 1*J* from ref. 26, discovering possibly hidden functional dependencies between state variables for a finite number of snapshots of those variables); social network analysis (Fig. 1*K* from ref. 27, discovering functional dependencies between quantitative markers associated with each individual in situations where the connectivity of the network may be hidden); economic network analysis (Fig. 1*M* from ref. 28, discovering functional dependencies between the economic markers of different agents or companies, which is significant to systemic risk analysis); and computational biology (Fig. 1*L* from ref. 29, identifying pathways and interactions between genes from their expression levels).

Overview of the Proposed Approach for Type 3 Problems

We first present an algorithmic overview of the proposed GP-based approach for Type 3 problems. For ease of presentation, we consider the simple setting of Fig. 2*A* where we are given N samples on the variables $x_1, \ldots, x_6$. After measurements/collection, these variables are normalized to have zero mean and unit variance. Our objective is to uncover the underlying dependencies between them.

A Signal-to-Noise Ratio to Decide Whether or Not a Node Has Ancestors. Our algorithm's core concept is the identification of ancestors for each node in the graph. Let us explore this idea in the context of a specific node, say x_1, as depicted in Fig. 2*B*. Determining whether x_1 has ancestors is akin to asking if x_1 can be expressed as a function of $x_2, x_3, \ldots, x_6$. In other words, can we find a function f (living in a prespecified space of functions that could be of controlled regularity) such that:

$$x_1 \approx f(x_2, \ldots, x_6)\,? \tag{1}$$

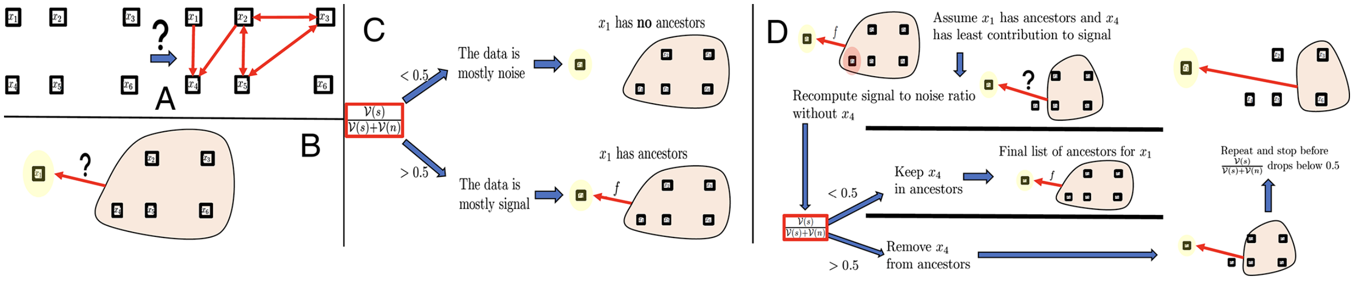
A
x_1
x_2
x_3
x_4
x_5
x_6
?
B
?
C
$\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)}$
< 0.5
The data is mostly noise
x_1 has **no** ancestors
> 0.5
The data is mostly signal
x_1 has ancestors
f
D
f
Assume x_1 has ancestors and x_4 has least contribution to signal
?
Recompute signal to noise ratio without x_4
$\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)}$
< 0.5
Keep x_4 in ancestors
Final list of ancestors for x_1
f
> 0.5
Remove x_4 from ancestors
Repeat and stop before $\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)}$ drops below 0.5

Fig. 2. Ancestors identification in Type 3 problem. (*A*) The Type 3 problem under consideration. (*B*) Can x_1 be approximated as a function of $x_2, \ldots, x_6$? (*C*) Using the signal-to-noise ratio to decide whether x_1 has ancestors. (*D*) Pruning ancestors in decreasing order of contributions to the signal and stopping before the signal-to-noise ratio drops below 0.5.

To answer this question, we regress f with a centered GP $\xi \sim \mathcal{N}(0, \Gamma)$ whose covariance function Γ is an additive kernel of the form $\Gamma = K_s + \gamma\delta(x-y)$, where K_s is a smoothing kernel, $\gamma > 0$ is a regularization parameter, and $\delta(x-y)$ is the white noise covariance operator. This is equivalent to assuming the GP ξ to be the sum of two independent GPs, i.e., $\xi = \xi_s + \xi_n$ where $\xi_s \sim \mathcal{N}(0, K_s)$ is a smoothing/signal GP and $\xi_n \sim \mathcal{N}\big(0, \gamma\delta(x-y)\big)$ is a noise GP. Writing $\mathcal{H}_{K_s}$ for the reproducing Kernel Hilbert space (RKHS) induced by the kernel K_s, this is also equivalent to approximating f with a minimizer of

$$\inf_{f \in \mathcal{H}_{K_s}} \|f\|_{K_s}^2 + \frac{1}{\gamma}\big\|f(X) - Y\big\|_{\mathbb{R}^N}^2, \qquad [2]$$

where $\|\cdot\|_{\mathbb{R}^N}^2$ is the Euclidean norm on $\mathbb{R}^N$, X is the input data on f obtained as an $N \times 5$-matrix whose rows X_i are the samples on $x_2, \ldots, x_6$, Y is the output data on f obtained as an N-vector whose entries are obtained from the samples on x_1, and $f(X)$ is a N-vector whose entries are the evaluations $f(X_i)$. At the minimum

$$\mathcal{V}(s) := \|f\|_{K_s}^2 \qquad [3]$$

quantifies the data variance explained by the signal GP ξ_s and

$$\mathcal{V}(n) := \frac{1}{\gamma}\big\|f(X) - Y\big\|_{\mathbb{R}^N}^2 \qquad [4]$$

quantifies the data variance explained by the noise GP ξ_n (17). This allows us to define the signal-to-noise ratio

$$\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)} \in [0, 1]. \qquad [5]$$

If $\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)} < 0.5$,* then, as illustrated in Fig. 2*C*, we deduce that x_1 has no ancestors, i.e., x_1 cannot be approximated as function of $x_2, \ldots, x_6$. Conversely if $\frac{\mathcal{V}(s)}{\mathcal{V}(s)+\mathcal{V}(n)} > 0.5$, then, we deduce that x_1 has ancestors, i.e., x_1 can be approximated as function of $x_2, \ldots, x_6$.

Selecting the Signal Kernel K_s. This process is repeated by selecting the kernel K_s to be linear ($K_s(x, x') = 1 + \beta_1 \sum_i x_i x_i'$), quadratic ($K_s(x, x') = 1 + \beta_1 \sum_i x_i x_i' + \beta_2 \sum_{i \le j} x_i x_j x_i' x_j'$) or fully nonlinear to identify f as linear, quadratic, or nonlinear. In the case of a nonlinear kernel, we employ:

$$K_s(x, x') = 1 + \beta_1 \sum_i x_i x_i' + \beta_2 \sum_{i \le j} x_i x_j x_i' x_j' + \beta_3 \prod_i (1 + k(x_i, x_i')), \qquad [6]$$

where k is a universal kernel, such as a Gaussian or a Matérn kernel, with all parameters set to 1, and β_i assigned the default value 0.1. We select K_s as the first kernel that surpasses a signal-to-noise ratio of 0.5. If no kernel reaches this threshold, we conclude that x_1 lacks ancestors.

Pruning Ancestors Based on Signal-to-Noise Ratio. Once we establish that x_1 has ancestors, the next step is to prune its set of ancestors iteratively. We remove nodes with the least contribution to the signal-to-noise ratio and stop before that ratio drops below 0.5 as illustrated in Fig. 2*D*. To describe this, assume that K_s is as in Eq. **6**. Then K_s is an additive kernel that can be decomposed into two parts:

$$K_s = K_1 + K_2, \qquad [7]$$

*We will later present a version with a more sophisticated method for pruning, but we keep the 0.5 threshold in this example for simplicity.

where $K_1 = 1 + \beta_1 \sum_{i \neq 1,2} x_i x_i' + \beta_2 \sum_{i \leq j, i,j \neq 1,2} x_i x_j x_i' x_j' + \beta_3 \prod_{i \neq 1,2} (1 + k(x_i, x_i'))$ does not depend on x_2 and $K_2 = K_s - K_1$ depends on x_2. This decomposition allows us to express f as the sum of two components:

$$f = f_1 + f_2, \qquad [8]$$

where f_1 does not depend on x_2, f_2 depends on x_2 and $(f_1, f_2) = \text{argmin}_{(g_1,g_2) \in \mathcal{H}_{K_1} \times \mathcal{H}_{K_2} \text{ s.t. } g_1+g_2=f} \|g_1\|_{K_1}^2 + \|g_2\|_{K_2}^2$. Furthermore, $\|f\|_{K_s}^2 = \|f_1\|_{K_1}^2 + \|f_2\|_{K_2}^2$, and $\frac{\|f_2\|_{K_1}^2}{\|f\|_{K_s}^2} \in [0, 1]$ quantifies the contribution of x_2 to the signal data variance. Following the procedure illustrated in Fig. 2*D*, if, for example, x_4 is found to have the least contribution to the signal data variance, we recompute the signal-to-noise ratio without x_4 in the set of ancestors for x_1. If that ratio is below 0.5, we do not remove x_4 from the list of ancestors, and x_2, x_3, x_4, x_5, x_6 is the final set of ancestors of x_1. If this ratio remains above 0.5, we proceed with the removal. This iterative process continues, and we stop before the signal-to-noise ratio drops below 0.5 to identify the final list of ancestors of x_1. The most efficient version of our proposed algorithm does not use a threshold of 0.5 on the signal-to-noise ratio to prune ancestors, but it rather employs an inflection point in the noise-to-signal ratio $\frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q)$ as a function of the number q of ancestors (Fig. 3*D*). To put it simply, after ordering the ancestors in decreasing contribution to the signal, the final number q of ancestors is determined as the maximizer of $\frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q+1) - \frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q)$.

Computational Complexity. We will now present a detailed analysis of the computational demands of the proposed method as a function of the number of variables, denoted as d, and the number of samples, N, pertaining to these variables. In the worst case, the proposed approach necessitates, for each of the d variables: for $i = 1, \ldots, d-1$, regressing a function mapping $d - i$ variables to the variable of interest and performing a mode decomposition, as exemplified in Eq. **8**, to identify the variable with the minimal contribution to the signal. Since these two steps have the same cost, it follows that, in the worst case, the total computational complexity of the proposed method is $\mathcal{O}(\mathbf{d^2N^3})$ which corresponds to product of the number of double-looping operations, d^2, and the cost of kernel regression from N samples which, without acceleration, is N^3 (i.e., the cost of inverting a $N \times N$ dense kernel matrix). However, if kernel scalability techniques are utilized, such as when the kernel has a rank k (for example, $k = d$ if the kernel is linear) or is approximated by a kernel of rank k (e.g., via a random feature map), then this worst-case bound can be reduced to $\mathcal{O}(\mathbf{d^2Nk^2})$ by reducing the complexity of each regression step from $\mathcal{O}(N^3)$ to $\mathcal{O}(Nk^2)$. Note that the statistical accuracy of the proposed approach requires that $N > d$ if the dependence of the unknown functions on their inputs is not sparse. Moreover, in the absence of kernel scalability techniques, the worst-case memory footprint of the method is $\mathcal{O}(N^2)$ due to the necessity of handling dense kernel matrices. However, once the functional ancestors of each variable are determined, these matrices can be discarded. Consequently, only one such matrix needs to be retained in memory at any given time.

Results

The following examples and experiments illustrate the proposed approach.

The Fermi–Pasta–Ulam–Tsingou System. The Fermi–Pasta–Ulam–Tsingou (FPUT) system (30) is a prototypical chaotic dynamical system. It is composed of M masses indexed by $j \in \{0, \ldots, M-1\}$ with equilibrium position jh with $h = 1/M$. Each mass is tethered to its two adjacent masses by a nonlinear spring, and the displacement of the mass x_j adheres to the equation:

$$\ddot{x}_j = \frac{c^2}{h^2}(x_{j+1} + x_{j-1} - 2x_j)\left(1 + \alpha(x_{j+1} - x_{j-1})\right), \qquad [9]$$

where $\alpha(x) = x^2$, $c = 1$ and $M = 10$. We use fixed boundary conditions by adding two more masses, with $x_{-1} = x_M = 0$. We take a total of 1,000 snapshots from multiple trajectories and the observed variables are the positions, velocities, and accelerations of all the underlying masses. In the graph discovery phase, every other node is initially deemed a potential ancestor for a specified node of interest. We then proceed to iteratively remove the node with the least signal contribution. The step resulting in the largest surge in the noise-to-signal ratio is inferred as one eliminating a crucial ancestor, thereby pinpointing the final ancestor set. Fig. 3*C* shows a plot of the noise-to-signal ratio $\frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q)$ as a function of the number q of proposed ancestors for the variable $\ddot{x}_7$ and with Z-test quantiles (in the absence of signal, the noise-to-signal ratio should fall within the shaded area with probability 0.9). Removing a node essential to the equation of interest causes the noise-to-signal ratio to markedly jump from approximately 25% to 99%. Fig. 3*D* shows a plot of the noise-to-signal ratio increments $\frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q) - \frac{\mathcal{V}(n)}{\mathcal{V}(s)+\mathcal{V}(n)}(q-1)$ as a function of the number q of ancestors for the variable $\ddot{x}_7$. Note that the increase in the noise-to-signal ratio is significantly higher compared to previous removals when an essential node was removed. Therefore, while solely relying on a fixed threshold to decide when to cease the removals might prove challenging, evaluating the increments in noise-to-signal ratios offers a clear guideline for efficiently and reliably pruning ancestors. The recovered full graph, depicted in Fig. 3*A*, is remarkably accurate despite the nonlinear nature of the model and the fact that our prior only encodes that the nonlinearity is smooth. Therefore, our algorithm does not require a dictionary or extensive knowledge of the structure of the unknown functions. Notably, velocity variables are accurately identified as nonessential and omitted from the ancestors of position and acceleration variables. Fig. 3*B*, which omits velocity variables for clarity, further elucidates the accurate recovery of dependencies. The dependencies are the simplest and clearest possible. They match exactly those of the original equations except for the boundary particles for which we recover valid equivalent equations.

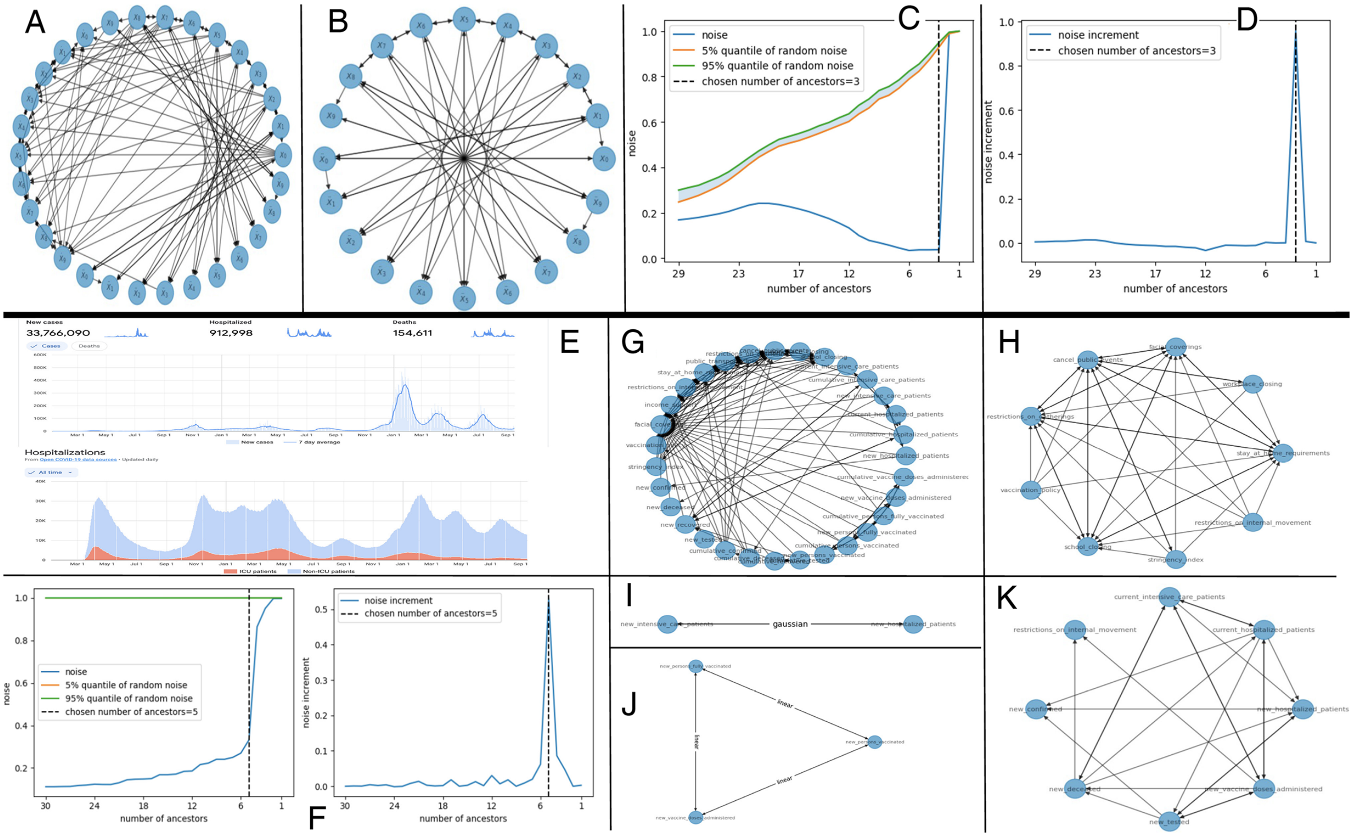

Fig. 3. (*A–D*) The Fermi–Pasta–Ulam–Tsingou system. (*E–K*) The Google COVID-19 open data.

The Google COVID-19 Open Data. Consider the COVID-19 data from Google.[†] We focus on a single country, France, to ensure consistency in the data and avoid considering cross-border variations that are not directly reflected in the data. We select 31 variables that describe the state of the country during the pandemic, spanning over 500 data points, with each data point corresponding to a single day. These variables are categorized as the following datasets: 1) Epidemiology dataset: Includes quantities such as new infections, cumulative deaths, etc. 2) Hospital dataset: Provides information on the number of admitted patients, patients in intensive care, etc. 3) Vaccine dataset: Indicates the number of vaccinated individuals, etc. 4) Policy dataset: Consists of indicators related to government responses, such as school closures or lockdown measures, etc. Some of these variables are illustrated in Fig. 3*E*. The problem is then to analyze this data and identify possible hidden functional relations between these variables. Fig. 3*F* shows the noise-to-signal ratio (and its increments) as function of the number of ancestors of the "cumulative number of hospitalized patients" variable. Even for this real dataset, the proposed approach gives a clear signal for stopping the pruning process. Fig. 3*G* shows the full recovered graph, which is highly clustered. Fig. 3*H* shows the cluster corresponding to the variable "schools closing" revealing that the government either implemented multiple restrictive measures simultaneously or lifted them in unison (except for mask mandates that were on the verge of being identified as noise). The vaccination cluster (Fig. 3*J*) reveals a linear relationship between variables (signaling redundant information) and the hospitalization cluster (Fig. 3*I*) reveals a nonlinear one. Eliminating redundant nodes leads to the sparse graph shown in Fig. 3*K*, which is interpretable and amenable to (both quantitative and qualitative) analysis,

Chemical Reaction Network. In this example, we consider the recovery of a chemical reaction network from concentration snapshots. The reaction network, illustrated in Fig. 4*A*, is that of the hydrogenation of ethylene (C_2H_4) into ethane (C_2H_6). The problem is that of recovering the underlying chemical reaction network from snapshots (illustrated in Fig. 4*B*) of concentrations $[H_2]$, $[H]$, $[C_2H_4]$, and $[C_2H_5]$ and their time derivatives. $\frac{d[H_2]}{dt}$, $\frac{d[H]}{dt}$, $\frac{d[C_2H_4]}{dt}$ and $\frac{d[C_2H_5]}{dt}$. The proposed approach leads to a perfect recovery of the computational graph (shown in Fig. 4*C*) and a correct identification of quadratic functional dependencies between variables.

Algebraic Equations. Fig. 4 *A–D* illustrates the application of the proposed approach to the recovery of functional dependencies from data satisfying hidden algebraic equations. In all these examples, we have $d = 6$ or $d = 7$ variables and $N = 1{,}000$ samples from those variables. For $d = 6$ the variables are $w_1, w_2, w_3, w_4, x_1, x_2$. For $d = 7$ the variables are $w_1, w_2, w_3, w_4, x_1, x_2, x_3$. The samples from the variables w_1 to w_4 are i.i.d. $\mathcal{N}(0, 1)$ random variables, and the samples from x_1, x_2 (and x_3 for $d = 7$) are functionally dependent on the other variables. In the first example, $d = 6$ and the samples from x_1 and x_2 satisfy the equations $x_1 = w_1$ and $x_2 = w_2$. The algorithm selects the linear kernel, and Fig. 4*A* shows the recovered graph (which is exact). In the second example, $d = 7$ and the samples from x_1, x_2, and x_3 satisfy the equations $x_1 = w_1$, $x_2 = x_1^2 + 1 + 0.1w_2$, and $x_3 = w_3$. The algorithm selects the quadratic

[†] The dataset can be accessed at https://health.google.com/covid-19/open-data/raw-data.

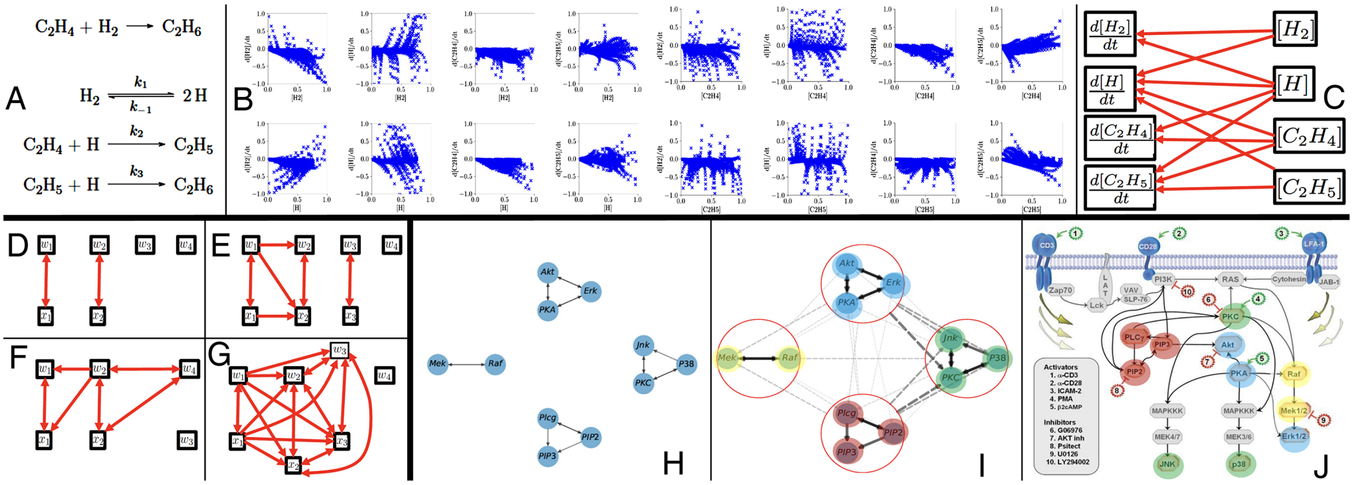

Fig. 4. (*A*–*C*) Chemical reaction network. (*D*–*G*) Algebraic equations. (*H*–*J*) Cell signaling network (Image credit: *J*, Reprinted from ref. 29, with permission from the American Association for the Advancement of Science).

kernel and Fig. 4*B* shows the recovered graph (which is exact). Even though x_2 can trace back its origin to either x_1 and w_2 or w_1 and w_2, the algorithm recognizes x_1, w_1, and w_2 as its ancestors underscoring the importance of eliminating redundant variables when aiming at deriving the sparsest graph. In the third example, $d = 6$ and the samples from x_1 and x_2 satisfy the equations $x_1 = w_1 w_2$ and $x_2 = w_2 \sin(w_4)$. The algorithm selects the nonlinear kernel and Fig. 4*C* shows the recovered graph (which is exact). In the fourth example, $d = 7$ and the samples from x_1, x_2, and x_3 satisfy the equations $x_1 = w_1$, $x_2 = x_1^3 + 1 + 0.1w_2$ and $x_3 = (x_1 + 2)^3 + 0.1w_3$. Although these equations appear to be cubic, the algorithm correctly selects the quadratic kernel and makes an exact recovery of the graph shown in Fig. 4*D* revealing hidden quadratic dependencies between variables.

Cell Signaling Network. Next, we apply the proposed framework to the example illustrated in Fig. 1*L* from ref. 29 and discover a hierarchy of functional dependencies in biological cellular signaling networks. We use single-cell data consisting of the $d = 11$ phosphoproteins and phospholipids levels in the human immune system T cells that were measured using flow cytometry. This dataset was studied from a probabilistic modeling perspective in previous works. While Sachs et al. (29) learned a directed acyclic graph to encode causal dependencies, Friedman et al. (31) learned an undirected graph of conditional independencies between the d molecule levels by assuming the underlying data follows a multivariate Gaussian distribution. The latter analysis encodes acyclic dependencies but does not identify directions. In this work, we aim to identify the functional dependencies without imposing strong distributional assumptions on the data. We simply use $N = 2{,}000$ samples chosen uniformly at random from the dataset consisting of 11 proteins and 7,446 samples of their expressions. We apply the algorithm in two stages. The first stage of the algorithm uses only linear and quadratic kernels and recovers the graph shown in Fig. 4*H*. It consists of four disconnected clusters where the molecule levels in each cluster are closely related by linear or quadratic dependencies (all connections are linear except for the connection between Akt and PKA, which is quadratic). These edges match a subset of the edges found in the gold standard model identified in ref. 29. With perfect noiseless dependencies, one can define constraints that reduce the total number of variables in the system. Second, we learn the connections between groups of variables within each cluster with nonlinear kernels and obtain the graph shown in Fig. 4*I* in which solid arrows indicate strong intracluster connections identified in the first level, and dashed lines indicate weaker connections between nodes and clusters identified in the second level. The width and grayscale intensities of each edge correspond to its signal-to-noise ratio. We emphasize that while causal graph recovery methods rely on the control of the sampling of the underlying variables (i.e., the simultaneous measurement of multiple phosphorylated protein and phospholipid components in thousands of individual primary human immune system cells, and perturbing these cells with molecular interventions), the reconstruction obtained by our method did not use this information and recovered functional dependencies rather than causal dependencies. Interestingly, the information recovered through our method appears to complement and enhance the findings presented in ref. 29 (e.g., the linear and noiseless dependencies between variables in the JNK cluster is not something that could easily be inferred from the graph produced in ref. 29 shown in Fig. 1*J* where we have colored the clusters for comparison).

Comparisons. Using the expected graph reported in ref. 29 as the ground truth (acknowledging that it may not be entirely accurate), we compare the edges our approach incrementally added to the true graph. Fig. 5*A* reports the number of additional edges that have been added and are not present in the ground truth (false positives) and edges removed that are present in the ground truth graph (false negatives). The added edges are based on the two-stage procedure described above, where we first add the ten intracluster connections, followed by intercluster connections. Edges are added in decreasing order of signal-to-noise ratio, starting with the strongest. In the reported results, we do not account for the recovery of the direction of ground-truth edges. We note that, up to direction, all intracluster connections, along with the intercluster connections with the strongest signals are found in the ground truth graph, leading to the initial decrease in false negatives with only one false positive edge (the linear connection P38 → Jnk that is not reported in the true graph). With the addition of the remaining (possibly nonspurious) edges, the number of false negatives drops to one, having recovered all edges, except for the one between PKC and Raf, which is identified to be statistically noninformative in our approach.

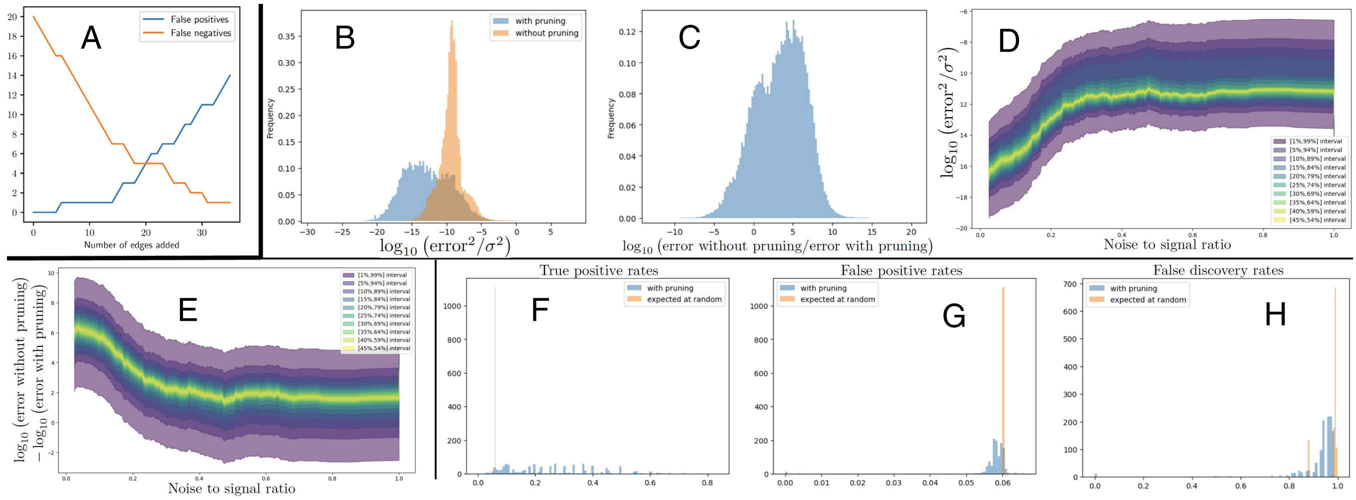

Fig. 5. (*A*) Cell signaling network comparisons. (*B*–*H*) The BCR reaction benchmark.

A Large-Scale Chemical Reaction Network: The BCR Reaction Benchmark. Finally, we stress-test the scalability of our approach by applying it to a large-scale chemical reaction network: the BCR reaction benchmark from ref. 32, which encompasses 1,122 species. The dataset comprises 2,400 snapshots of species concentrations and their corresponding time derivatives. We leveraged JAX's inherent parallelization capabilities (33) to accelerate our computations, allowing for the simultaneous pruning of multiple nodes while abstracting the complexity of parallel execution. While the scaling with respect to the number of data points is straightforward, scaling with the number of variables introduces a trade-off between computational speed and memory footprint. Specifically, the process of identifying the ancestors of various nodes can be expedited by storing a large array for all nodes. Using a DGX workstation equipped with four Nvidia V100 GPUs, each with 32GB of memory, pruning 190 nodes took approximately three days, projecting a total experiment duration of around one month. Nonetheless, we can mitigate this computational burden by optimizing the computation of terms of the form $y^T Ky$ for the specific quadratic kernel identified for this example. We include the details of such optimization in *SI Appendix*, section 8E. By implementing this optimization, the duration of the entire experiment was reduced to just 1 h.

In the first experiment, we simulated five trajectories of the associated system of ODEs, recording 1,000 snapshots per trajectory. Out of these 5,000 snapshots, 2,400 were randomly selected as training data, and 2,600 as testing data. Writing TP, TN, FP, and FN for True/False Positives/Negatives and using the metrics True Positive Rate (TPR=TP/(TP+FN)), False Positive Rate (FPR=FP/(FP+TN)), and false discovery rate (FDR=FP/(TP+FP)), we observed a TPR of 39.9%, an FPR of 16.4%, and an FDR of 97.2% (indicating that 97.2% of predicted positives are false). This high FDR can be attributed to the limited exploration of the full variable range-1,122 in total-by the five trajectories. The trajectories explored a subset of the possible space (near a limit cycle attractor), which led to the recovery of functional dependencies that represent both the chemical reactions and the specific subspace visited. Furthermore, with 1,122 variables, the 630,003 coefficients of the underlying quadratic equations are vastly underdetermined with only 2,400 data points. Despite the high FDR in the recovered graph, as illustrated in Fig. 5 *B* and *C*, the CHD pruning process vastly improves the accuracy (by orders of magnitude) of the estimated functions on the 2,600 unseen snapshots by reducing the dimension of the regression problem whenever possible. We denote y_i as an observed data point, σ^2 as the variance of the observed data, $\hat{y}_i$ for a predicted data point without pruning, and $\bar{y}_i$ for a predicted data point postpruning. Fig. 5*B* illustrates the histogram of the log-normalized squared errors before and after pruning, expressed as $\log_{10}\left(|y_i - \hat{y}_i|^2/\sigma^2\right)$ and $\log_{10}\left(|y_i - \bar{y}_i|^2/\sigma^2\right)$. The 99th percentile of the normalized squared error is less than 10^{-2} for all species. Fig. 5*C* displays the histogram of the log-normalized squared error improvements due to pruning, calculated as $\log_{10}\left(|y_i - \hat{y}_i|^2/|y_i - \bar{y}_i|^2\right)$. Fig. 5 *D* and *E* displays the quantiles of the histograms postpruning, conditioned on the noise-to-signal ratio observed at the final pruning step. These plots reveal a clear trend: a higher noise-to-signal ratio at the time of pruning correlates with increased error and diminished improvements in accuracy.

In a second experiment, we formed the data by randomly sampling concentrations uniformly in $[0, 1]$ (independently across species and snapshots) and recorded the resulting time derivatives. While this sampling increased the variability of the 2,400 snapshots, the model remained vastly underdetermined. The noise-to-signal and bootstrapped (Z-test) ratios remained close to 0.5, suggesting insufficient data for statistically significant variable importance assessments. Nonetheless, as depicted in Fig. 5 *F*–*H*, significant insights can still be gleaned from the activations, showing notable improvements when comparing the histograms of the values of TPR, FPR, and FDR obtained with pruning based on these ratios and pruning at random. This analysis reveals that even with high dimensionality and scarce data, between 10% and 80% of the true ancestors can still be accurately identified.

Discussions

Limitations. In its present form, the proposed approach is limited by several factors. 1) Without access to the sampling of the data, the direction of some edges may not be identifiable. For instance, the functional relationship $x - 2y = 0$ can be represented as both $y = 2x$ ($x \to y$) and $x = y/2$ ($y \to x$). 2) It assumes an additive noise W on the functional relationship $y = f(x) + W$ between the variables x and y. In a fully probabilistic setting, this structure may be nonadditive, i.e., of the form $y = f(x, W)$, which implies discovering a general transition kernel, i.e., a non-Gaussian generative model. Although our method achieves polynomial complexity, in settings where one has access to the distribution of the data, the price to pay, when compared with information-theoretic methods, is a reduction in generality imposed by the stronger assumption made on the data-generating process. Furthermore, the price to pay for the weaker data requirements (i.e., the absence of interventional data) is that our method recovers functional relationships rather than causal ones or conditional dependencies. 3) If the (noisy) functional relationship $y = f(x) + W$ is associated with a nonregular (e.g., discontinuous) function f then the kernels discussed above (linear, quadratic, and fully nonlinear) will be misspecified and may lead to false negatives. The kernel selection and hyperparameter tuning problems in misspecified settings require further work. 4) As demonstrated in the BCR reaction application, while the method scales well computationally with an increase in the number of variables, it may still be impacted by the curse of dimensionality. This occurs particularly if the dataset only covers a limited subset of the full range of variable values. Given the results displayed in Fig. 5 *B*–*H*, we suspect that this impact could be mitigated by adopting more advanced strategies in place of our current top–down pruning method. Such strategies could involve grouping variables and integrating both top–down and bottom–up iterative approaches.

Conclusions. We have developed a comprehensive GP framework for solving Type 3 (hypergraph discovery) problems, which is interpretable and amenable to analysis. The breadth and complexity of Type 3 problems significantly surpass those encountered in Type 2 (hypergraph completion), and the initial numerical examples we present serve as a motivation for the scope of Type 3 problems and the broader applications made possible by this approach. Our proposed algorithm is designed to be fully autonomous, yet it offers the flexibility for manual adjustments to refine the graph's structure recovery. We emphasize that our proposed approach is not intended to supplant causal inference methods (34); see *SI Appendix*, section 4C for a complete overview. Instead, it aims to incorporate a distinct kind of information into the graph's structure, namely, the functional dependencies among variables rather than their causal relationships. Additionally, our method eliminates the need for a predetermined ordering of variables, a common requirement in acyclic probabilistic models where determining an optimal order is an NP-hard problem usually tackled using heuristic approaches. Furthermore, our approach can actually be utilized to generate such an ordering by quantifying the strength of the connections it recovers. The Uncertainty Quantification properties of the underlying GPs are integral to the method and could also be employed to quantify uncertainties in the structure of the recovered graph. We also observe that forming clusters from highly interdependent variables helps to obtain a sparser graph. Additionally, the precision of the pruning process is enhanced by avoiding the division of node activation within the cluster among its separate constituents. We employed this strategy in the recovery of the gene expression graph in Fig. 4*I*. Given the polynomial complexity of our method, promising avenues for future work include applications to large datasets in genomics and in systems biology, particularly in the reconstruction and intervention of metabolic pathways. These applications benefit from the ability to handle large-scale datasets efficiently, enabling the analysis of complex biological networks.

Data, Materials, and Software Availability. Data and Python Code data [the code for the algorithm and its application to various examples are available for download (and as an installable python library/package)] have been deposited in Github (https://github.com/TheoBourdais/ComputationalHypergraphDiscovery).

ACKNOWLEDGMENTS. H.O., T.B., P.B., X.Y., and R.B. acknowledge support from the Air Force Office of Scientific Research under MURI Award No. FA9550-20-1-0358 (Machine Learning and Physics-Based Modeling and Simulation). Additionally, H.O., T.B., and P.B. acknowledge support by the Department of Energy under Award No. DE-SC0023163 (SEA-CROGS: Scalable, Efficient and Accelerated Causal Reasoning Operators, Graphs and Spikes for Earth and Embedded Systems) and by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the NASA. H.O. and P.B. further acknowledge support by Beyond Limits (Learning Optimal Models) through The Caltech Center for Autonomous Systems and Technologies. T.B. acknowledges support from the Kortschak Scholar Fellowship Program. N.R. acknowledges support from the JPL Researchers on Campus Program. H.O. is grateful for the Department of Defense Vannevar Bush Faculty Fellowship. We are grateful to two referees for their insightful comments and valuable suggestions.

Author affiliations: [a]Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125; and [b]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

1. S. L. Morgan, C. Winship, *Counterfactuals and Causal Inference* (Cambridge University Press, 2015).
2. M. Glymour, J. Pearl, N. P. Jewell, *Causal Inference in Statistics: A Primer* (John Wiley & Sons, 2016).
3. O. Stegle, D. Janzing, K. Zhang, J. M. Mooij, B. Schölkopf, Probabilistic latent variable models for distinguishing between cause and effect. *Adv. Neural. Inf. Process. Syst.* **23**, 1687–1695 (2010).
4. D. Lopez-Paz, K. Muandet, B. Schölkopf, I. Tolstikhin, "Towards a learning theory of cause–effect inference" in *International Conference on Machine Learning* (PMLR, 2015), pp. 1452–1461.
5. A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs. *J. Comput. Phys.* **230**, 3015–3034 (2011).
6. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* **13**, 3932–3937 (2016).
7. H. Owhadi, Computational graph completion. *Res. Math. Sci.* **9**, 1–33 (2022).
8. H. Owhadi, Do ideas have shape? Idea registration as the continuous limit of artificial neural networks. *Physica D* **444**, 133592 (2023).
9. Y. Chen, B. Hosseini, H. Owhadi, A. M. Stuart, Solving and learning nonlinear PDEs with Gaussian processes. *J. Comput. Phys.* **447**, 110668 (2021).
10. P. Batlle, Y. Chen, B. Hosseini, H. Owhadi, A. M. Stuart, Error analysis of kernel/GP methods for nonlinear and parametric PDEs. arXiv [Preprint] (2023). https://arxiv.org/abs/2305.04962 (Accessed 5 August 2023).
11. Y. Chen, H. Owhadi, F. Schäfer, Sparse Cholesky factorization for solving nonlinear PDEs via Gaussian processes. *Math. Comput.*, 10.1090/mcom/3992 (2024).
12. M. Darcy, B. Hamzi, G. Livieri, H. Owhadi, P. Tavallali, One-shot learning of stochastic differential equations with data adapted kernels. *Physica D* **444**, 133583 (2023).
13. B. Hamzi, H. Owhadi, Y. Kevrekidis, Learning dynamical systems from data: A simple cross-validation perspective. Part IV: Case with partial observations. *Physica D* **454**, 133853 (2023).
14. C. A. Micchelli, T. J. Rivlin, Eds., "A survey of optimal recovery" in *Optimal Estimation in Approximation Theory* (Springer, 1977), pp. 1–54.
15. H. Owhadi, C. Scovel, *Operator Adapted Wavelets, Fast Solvers, and Numerical Homogenization, from a Game Theoretic Approach to Numerical Approximation and Algorithm Design. Cambridge Monographs on Applied and Computational Mathematics* (Cambridge University Press, 2019).
16. G. Wahba, An introduction to smoothing spline ANOVA models in RKHs, with examples in geographical data, medicine, atmospheric sciences and machine learning. *IFAC Proc. Vol.* **36**, 531–536 (2003).
17. H. Owhadi, C. Scovel, G. R. Yoo, *Kernel Mode Decomposition and the Programming of Kernels* (Springer, 2021).
18. A. N. Whitehead, *An Introduction to Mathematics* (Williams and Norgate, London, 1911).
19. H. Owhadi, C. Scovel, F. Schäfer, *Statistical numerical approximation. Not. AMS* **66**, 1608–1617 (2019).
20. F. Schäfer, T. J. Sullivan, H. Owhadi, Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *Multiscale Model. Simul.* **19**, 688–730 (2021).
21. F. Schäfer, M. Katzfuss, H. Owhadi, Sparse Cholesky factorization by Kullback–Leibler minimization. *SIAM J. Sci. Comput.* **43**, A2019–A2046 (2021).
22. B. Hamzi, R. Maulik, H. Owhadi, Simple, low-cost and accurate data-driven geophysical forecasting with learned kernels. *Proc. R. Soc. A* **477**, 20210326 (2021).

23. B. Hamzi, H. Owhadi, Learning dynamical systems from data: A simple cross-validation perspective. Part I: Parametric kernel flows. *Physica D* **421**, 132817 (2021).
24. F. Schäfer, H. Owhadi, Sparse recovery of elliptic solvers from matrix-vector products. *SIAM J. Sci. Comput.* **46**, A998–A1025 (2023).
25. P. Batlle, M. Darcy, B. Hosseini, H. Owhadi, Kernel methods are competitive for operator learning. *J. Comput. Phys.* **496**, 112549 (2023).
26. P. A. Dirmeyer, P. Gentine, M. B. Ek, G. Balsamo, "Land surface processes relevant to sub-seasonal to seasonal (S2S) prediction" in *Sub-Seasonal to Seasonal Prediction*, A. W. Robertson, F. Vitart, Eds. (Elsevier, 2019), pp. 165–181.
27. J. H. Gittell, H. N. Ali, *Relational Analytics: Guidelines for Analysis and Action* (Routledge, 2021).
28. F. Schweitzer *et al*., Economic networks: The new challenges. *Science* **325**, 422–425 (2009).
29. K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, G. P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**, 523–529 (2005).
30. R. S. Palais, The symmetries of solitons. *Bull. Amer. Math. Soc. (N.S.)* **34**, 339–403 (1997).
31. J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432–441 (2008).
32. T. E. Loman *et al.*, "Catalyst: Fast and flexible modeling of reaction networks." *PLoS Comput. Biol.*, 10.1371/journal.pcbi.1010722 (2023).
33. J. Bradbury *et al.*, JAX: Composable transformations of Python+NumPy programs (Version JAX[GPU] 0.4.28). Github. http://github.com/google/jax. Accessed 29 May 2024.
34. J. Pearl, *Causality* (Cambridge University Press, 2009).