On the parameter combinations that matter and on those that do not: data-driven studies of parameter nonidentifiability

Nikolaos Evangelou[a,*,†], Noah J. Wichrowski[b,*,†], George A. Kevrekidis[c], Felix Dietrich [d], Mahdi Kooshkbaghi [e], Sarah McFann [f,g] and Ioannis G. Kevrekidis[a,b,*]

[a]Department of Chemical and Biomolecular Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
[b]Department of Applied Mathematics and Statistics, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
[c]Department of Mathematics and Statistics, University of Massachusetts, 710 N Pleasant St, Amherst, MA 01003, USA
[d]Department of Informatics, Technical University of Munich, Boltzmannstr. 3, Garching 85748, Germany
[e]The Program in Applied and Computational Mathematic, Princeton University, Washington Road, Princeton, NJ 08544, USA
[f]Department of Chemical and Biological Engineering, Princeton University, 50–70 Olden St, Princeton, NJ 08544, USA
[g]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08540, USA
*To whom correspondence should be addressed: Email: yannisk@jhu.edu
Edited By: Karen E. Nelson

Abstract

We present a data-driven approach to characterizing nonidentifiability of a model's parameters and illustrate it through dynamic as well as steady kinetic models. By employing Diffusion Maps and their extensions, we discover the minimal combinations of parameters required to characterize the output behavior of a chemical system: a set of *effective parameters* for the model. Furthermore, we introduce and use a Conformal Autoencoder Neural Network technique, as well as a kernel-based Jointly Smooth Function technique, to disentangle the *redundant* parameter combinations that do not affect the output behavior from the ones that do. We discuss the interpretability of our data-driven effective parameters, and demonstrate the utility of the approach both for behavior prediction and parameter estimation. In the latter task, it becomes important to describe level sets in parameter space that are consistent with a particular output behavior. We validate our approach on a model of multisite phosphorylation, where a reduced set of effective parameters (nonlinear combinations of the physical ones) has previously been established analytically.

Keywords: parameter nonidentifiability, manifold learning, model order reduction, data mining.

Significance Statement:

A mathematical model is *nonidentifiable* if observations of its output behavior do not suffice to uniquely determine the parameter values resulting in these observations. We propose a fully data-driven approach to distinguish those parameter combinations that affect the behavior (*effective* parameter combinations) from those that have no such influence (*redundant* parameter combinations). We also discuss the interpretability of our data-driven effective parameter combinations by mapping them invertibly to candidate sets of physically interpretable ones. Our scheme performs estimation of effective parameter combination values from observations, foliation of parameter space by observation level sets, as well as behavior estimation/prediction from parameters in a seamless, data-driven manner.

Introduction

Model reduction has long been an important endeavor in mathematical modeling of physical phenomena and, in particular, in the modeling of large, complex kinetic networks of the forms that arise in combustion or in cellular signaling (1–3). A rich array of techniques, often based on time-scale separation, exist that can result in a smaller number of *effective state variables* and, consequently, a reduced set of coupled nonlinear differential equations [e.g. Benner et al. (4), Quarteroni et al. (5), and from our work (6–10)]. Yet, it also becomes important to discover, when possible, a smaller number of *effective parameters*. These are (possibly

nonlinear) combinations of the original, usually physically meaningful, model parameters on which the output behavior depends. A universally accepted and practiced approach toward reducing the set of parameters, undertaken before any computation is, of course, dimensional analysis (11).

Beyond dimensional analysis, the issue of parameter nonidentifiability, whether truly structural or approximate, has been the subject of extensive studies for decades, with rekindled interest in recent years (12, 13). Such developments are eloquently summarized in ref. (14). This can be attributed in part to sloppiness/MBAM studies (15, 16); the study of active subspaces (17); the increased
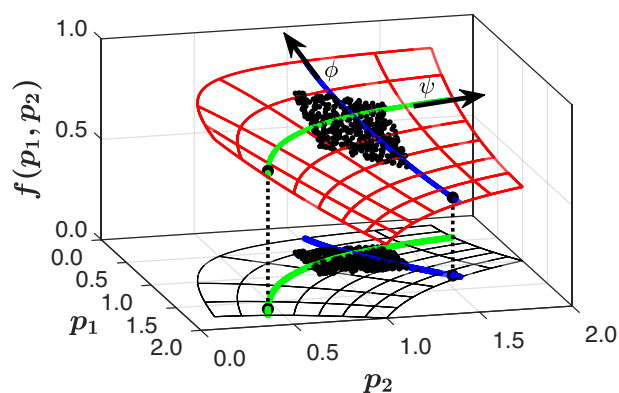
Fig. 1. The function $f(\phi) = \exp(-\phi/2)$, with $\phi = p_1 p_2$, is sampled at a cloud of points and plotted against the two parameters p_1, p_2 (red mesh). Here, ϕ is the *effective parameter*, which we call the "meaningful" parameter combination. The green curve indicates a level set of this effective parameter, for which $f(\phi) - C = 0$, for some constant C (here, $C = 0.75$). The blue curve illustrates the direction orthogonal to each level set of ϕ, parameterized by $\psi = p_1^2 - p_2^2$, which we call the "redundant" parameter combination because it does not affect the output. The projection onto the (p_1, p_2)-plane helps illustrate the level sets of the meaningful and redundant parameter combination(s) in parameter space.

availability and exploitation of symbolic regression packages (18); and, more generally, to recent advances in data science and manifold learning techniques (19, 20). To a large extent, established model reduction techniques hinge on the availability of analytical model equations and operations (e.g. singular perturbation theory-based expansions) on these closed form equations.

This work aspires to synthesize and implement a purely data-driven process for finding reduced effective parameters. The type of models we consider here are systems of coupled, nonlinear, first-order differential equations describing time-evolution of chemical/biochemical reaction networks, but the approach is applicable more generally to the parameterization of input–output relations. Here, the inputs are the parameters, and the outputs are time series of the system observables, such as species concentrations, temperatures, or functions of such quantities.

In Fig. 1, we illustrate a simple model with structurally nonidentifiable parameters. The model output, $f(p_1, p_2) = \exp(-p_1 p_2/2)$, though plotted as a function of the two parameters (p_1, p_2), in fact depends only on their product $\phi = p_1 p_2$. The output data do not suffice to identify or estimate p_1 and p_2 independently: observations can only confine pairs of p_1 and p_2 to a level set, colored green in Fig. 1, of the *effective parameter* ϕ. It is interesting to observe that these level sets are parameterized by the quantity $\psi = p_1^2 - p_2^2$, which is conformal everywhere to $p_1 p_2$, thus making ϕ and ψ an orthogonal system of coordinates (cf. polar or hyperbolic coordinates). A level set for ψ is colored blue in Fig. 1. This is the parameter combination that *does not matter to the output*, one that is "redundant": keeping the output constant while changing ψ traces out the level set $\phi = C$. To trace out the possible values of the output, one could of course fix one parameter (say, p_2) and vary the other(s). In that case, however, the sensitivity of the output to the variation of p_1 depends on the value at which we choose to keep p_2 constant. This variability is avoided when using a conformal orthogonal set of coordinates, such as the one in the figure.

In our illustrative models, the system is available in the form of a "black box" set of ordinary differential equations (ODEs): given parameter values and initial conditions, one can record time

series of the evolution of the system states, or of functions of the system states. But the evolution equations are not explicitly available, so that analytical (possibly perturbative) approaches to reduction of either system states or parameters (outputs or inputs) cannot be undertaken. Given such an input–output model, we start by systematically prescribing a set of numerical experiments for data collection. These data will be processed with manifold learning techniques—here, Diffusion Maps (DMaps) and Geometric Harmonics (GH)—as well as their extensions: output-informed DMaps and Double DMaps GH. Processing the data will:
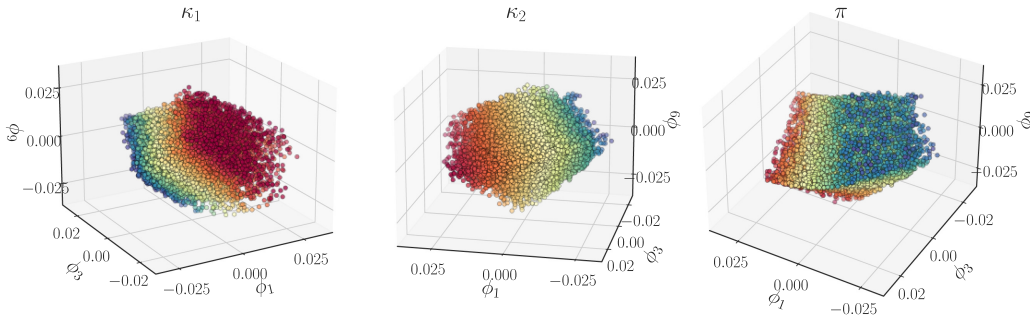
- determine the number of model parameter combinations that matter, i.e. the meaningful effective parameters that affect the model output;
- consequently, determine the number of model parameter combinations that do not matter, the redundant ones;
- interpret the meaningful parameter combinations through computational testing/validation of expert suggestions, or possibly through symbolic regression;
- disentangle the redundant parameter combinations from the meaningful effective ones (21, 22), which is accomplished using deep learning techniques (Conformal Autoencoders) or, alternatively, kernel-based Jointly Smooth Feature extraction (23);
- translate between the data-driven effective parameters and physical ones, which underscores the importance of level sets in parameter space consistent with the same output behavior.

We believe these capabilities constitute a useful toolkit for data-driven reparameterization of models, whether computational or physical/experimental. In the experimental setting, the same toolkit can be applied; one will perturb ("jiggle") all inputs/parameters around a base point, record the richness of the resulting output behavior, and establish (through the same framework) correlations between parameters' richness and output richness—quantify it and parametrize it.

The remainder of the paper is organized as follows: In the section "The MSP model," we will demonstrate and visualize the discovery of the intrinsic dimensionality of the meaningful effective parameter space through our main illustrative example: a six-equation multisite phosphorylation (MSP) kinetic model and its analytical reduction by Yeung et al. (24). In the section "Data-driven parameter reduction," we compare our data-driven effective parameter constructs with those previously obtained analytically and discuss their interpretability, both numerically and through symbolic regression. Finally, we demonstrate the use of these effective parameters in behavior prediction for new physical parameter settings in the section "Effective parameter identification" and (a type of) parameter estimation for previously unobserved behaviors in the section "Behavior estimation." Toward the latter task, in the section "Parameter estimation," we discover and parameterize entire level sets in parameter space that are consistent with this new observed behavior; this requires discovering the redundant parameter combinations. In the section "On the parameter combinations that do not matter," a deep learning architecture (Conformal Autoencoder Networks) as well as an alternative kernel-based *Jointly Smooth Functions (JSFs) extraction* is used for this task of disentangling meaningful effective parameters from redundant ones. We conclude by summarizing the approach and offering a discussion of its potential, shortcomings, and current research directions.

In Supplementary Material Sections S5 and S6, we have included two additional examples. The first comes from a textbook nonidentifiable dynamical system representing a compartmental

model and the second is a steady-state example, which allows us to illustrate how our data-driven framework behaves when transitions between qualitatively distinct behavior regimes arise as one traverses the original parameter space.

Results

The "black box" models that we seek to parameterize in our data-driven work arise mainly from chemical kinetic mechanisms (e.g. Eq. 1), which give rise to systems of ODEs for the evolution in time of the species concentrations as output, depending on several kinetic parameters, possibly including the total quantity of a catalyst or enzyme, as input. In certain parameter regimes, the existence of disparate (fast–slow) time scales allows one to explicitly reduce a detailed kinetic scheme through, e.g. the Bodenstein (25) or Quasi-Steady-State Approximation (QSSA) to an effective reduced one, characterized by new, reduced effective parameters.

The detection of such effective parameters in our scheme will be achieved by using the manifold learning algorithm DMaps (19), for which a more detailed description is given in Section S2A of the Supplementary Material. We will illustrate that, given a systematically collected data set, and with an appropriate *metric*, DMaps can be used for parameter reduction: discovery of effective parameter combinations that affect the output, as well as parameter combinations that do not affect it. The motivation for our work arose from studying the reduction of the following MSP model.

The MSP model

Yeung et al. (24) proposed and analyzed a kinetic model that describes the dual phosphorylation of extracellular signal-regulated kinase (ERK) by an enzyme known as mitogen-activated protein kinase kinase (MEK). Here, ERK can exist in any of three states: S_0, S_1, and S_2, where the subscript indicates the number of times the substrate has been phosphorylated. The MEK enzyme, denoted by E, forms complexes ES_0 and ES_1 with the first two phosphostates. The reaction mechanism for this system is given by

$$E + S_0 \underset{k_{r,1}}{\overset{k_{f,1}}{\rightleftharpoons}} ES_0 \overset{k_{cat,1}}{\rightarrow} ES_1 \overset{k_{cat,2}}{\rightarrow} E + S_2$$

$$k_{f,2} \uparrow\downarrow k_{r,2}$$

$$E + S_1, \tag{1}$$

with the six rate constants comprising our vector of inputs/parameters:

$$\boldsymbol{p} = [k_{f,1}, k_{r,1}, k_{cat,1}, k_{f,2}, k_{r,2}, k_{cat,2}]^\top \in \mathbb{R}^6.$$

The governing system of ordinary differential equations is listed in Section S1A of the Supplementary Material.

Yeung et al. used the QSSA for the species ES_0 and ES_1 along with stoichiometric conservation to approximately reduce the above system: If the assumptions

$$S_{tot} \ll \frac{k_{f,1} + k_{cat,1}}{k_{f,1}}, \qquad S_{tot} \ll \frac{k_{r,2} + k_{cat,2}}{k_{f,2}}$$

reasonably hold, where

$$S_{tot} = [S_0]\big|_{t=0} = [S_0] + [S_1] + [S_2] + [ES_0] + [ES_1],$$

then the initial model reduces to a three-state linear kinetic model that depends on only three effective parameters, which are

combinations of the full model parameters

$$\kappa_1 = [E] \frac{k_{f,1} k_{cat,1}}{k_{r,1} + k_{cat,1}},$$

$$\kappa_2 = [E] \frac{k_{f,2} k_{cat,2}}{k_{r,2} + k_{cat,2}},$$

$$\pi = \frac{k_{cat,2}}{k_{r,2} + k_{cat,2}}. \tag{2}$$

The reduced equations can be found in Section S1B of the Supplementary Material. We will attempt to derive such a reduced parameterization in a data-driven manner.

Data-driven parameter reduction

We select a *base point* in parameter space

$$\tilde{\boldsymbol{p}} = [0.71, 19, 6700, 9200, 0.97, 5200]^\top,$$

which is situated in the region of parameter space where the reduction assumptions hold. We select a reference initial condition $[S_0] = 5$ and $[E] = 0.66$, with the other species not initially present. Numerically integrating the associated system of ODEs, we collected 10,000 dynamic observations of the system output in response to perturbations of each parameter within ±10% of its base value. Note that our random parameter perturbations, are merely a device for sampling the neighborhood of the base point in input/parameter space; a grid of equally spaced points would also suffice.

In the following analysis, we take as our outputs the concentration $[S_2]$ at $t \in \{2, 4, \dots, 20\}$, which yields a 10D observation vector at each parameter setting. For this example, the choice of $[S_2]$ as the observed output is not particularly significant; the temporal response of *any* time-varying chemical species or combination thereof would give the same results [based on Takens' embedding theorem (26)]. We will refer to this particular data set $\mathbf{X}$ as the *transient data*. This data set samples what in the literature is referred to as *the model manifold*, whose dimensionality determines the number of meaningful (effective) parameters (16, 27, 28).

A second data set, $\mathbf{Y}$, was obtained through computational optimization experiments, in which we estimated vectors of six parameter values that best fit the reference transient, we obtained at the base parameter setting. In these experiments, initial conditions were chosen randomly from a log10-uniform distribution, with lower and upper bounds set, respectively, at 10^{-3} and 10^{+3} times the rates estimated by Aoki et al. (29). We performed nonlinear least-square fits of these transients from 1,000 random initial conditions in 6D parameter space, as described in ref. (24). Upon successful completion of these computations, we have 1,000 6D "optimal fits" of the base parameter setting; we call this data set the *optimization data*.

We first computed *output informed* DMaps, with the distance metric described in Section S2A of the Supplementary Material, on the transient data set. The observed outputs in $\mathbf{X}$ for these computations, are used without reference to the values of the analytical parameters provided in Eq. (2); the latter will only be considered later for validation of our data-driven approach: comparing our effective parameters to previously analytically known ones will confirm that our data-driven scheme finds a parameter representation that is equivalent to that proposed in Eq. (2), which we use later only as a means of comparison to confirm that our data-driven scheme finds a representation that is equivalent to that proposed in ref. (24). The number of independent/nonharmonic eigenvectors indicates the effective dimensionality of the model

Fig. 2. The first three independent, nontrivial eigenvectors, ϕ_1, ϕ_3, ϕ_9, colored by (computed) values of the three theoretical effective parameters, κ_1, κ_2, π, respectively, for a transient data set.

manifold. We found three nonharmonic DMaps eigenvectors (30)

$$\boldsymbol{\phi} = (\phi_1, \phi_3, \phi_9) \in \Phi \subset \mathbb{R}^3$$

and deduced that the intrinsic dimensionality of the transient data set, and thus of the model manifold, is three. We then turned to the optimization data set and performed both principal component analysis (PCA) and "regular" DMaps. We found that the intrinsic dimensionality of the optimization data set is also three, whether we estimate it from PCA or from DMaps. These two results corroborate/complement each other, since three plus three equals six, the total number of original parameters.

The dimensionality of the *transient data set* could be estimated from the dimension of the null space of either the sensitivity matrix or the sensitivity Fisher information matrix (14) at the base point. Beyond this estimate, however, our approach discovers a *global* parameterization over the data of the output in terms of $\boldsymbol{\phi} = (\phi_1, \phi_3, \phi_9)$, which are our data-driven effective parameters. These eigenmodes capture the directions, in full parameter space, that matter to the output: the parameter changes that affect the response of our system. Figure 2 illustrates these three leading nonharmonic eigenvectors, colored by the analytical effective parameters of Yeung et al. in Eq. (2). Even though it is difficult to visually appreciate a 3D point cloud through color, we believe one gets a clear visual impression that the data-driven effective parameter set and the analytical effective parameter set are one-to-one with each other. We will quantify this below.

We remind the reader that the DMaps effective parameters, like the analytical ones, will in general correspond to *combinations* of the original parameters. But while the analytical effective parameters are physically explainable [Eq. (2) shows their dependence on the original parameters], no such a priori physical interpretation comes with the proposed data-driven effective parameters. We will address this issue below.

Computing DMaps on the *optimization* data also results in an intrinsically 3D parameterization of the manifold of equivalent optima (Fig. 3). The intrinsic parameters computed for this data set uncover the directions in parameter space that produce (approximately) the same response: the reference trajectory at the base input settings. This dictates how many parameter combinations *do not matter* to the recorded output response. This structural non-identifiability, computed around a selected output response (one in a base setting) is a property of the system in a neighborhood of that setting, as long as the intrinsic dimensionality of the responses does not change when we perturb the base parameter values (i.e. as long as the QSSA remains valid, see the discussion in Section S2D of the Supplementary Material). For our example, it was sufficient to perform *linear* data processing of the optima
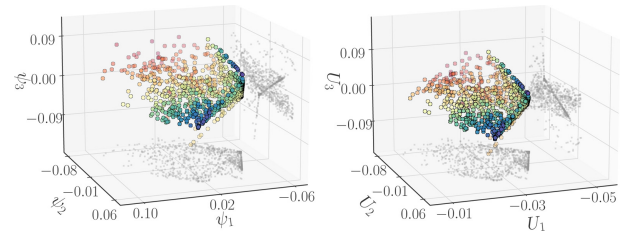
Fig. 3. (Left) Independent eigenvector coordinates, ψ_1, ψ_2, ψ_3, for the optimized data set, colored by ψ_2. (Right) The three dominant singular vectors computed with PCA, colored by the second, U_2.

by PCA. Indeed, the three redundant parameter combinations for the reference trajectory happen to live on a 3D hyperplane in full parameter space; this hyperplane contains $\sim 99\%$ of the total variance of the 6D parameter vectors in the optimization set. In this example, it so happens that linear data analysis (PCA) is sufficient to determine the "minimal response richness": The responses lie on a 3D hyperplane in the 10D output space. In general (and, we expect, most often), PCA will suggest *more* than the truly minimal number of effective parameters to span the data, and nonlinear tools like DMaps would be required to find a minimal set.

We already have our first result: a data-driven corroboration of the *number* of effective parameter combinations. Three of them matter, and three of them do not, adding up to the correct total number of six full inputs. The reader may already have noticed that these structurally unidentifiable combinations *are not global*; they are valid only for the reference trajectory. Beyond finding this number, we will also construct a global parameterization/foliation of the "hypersurfaces that do not matter" in the original input space. Even though only three-dimensional, they are impossible to visualize, leading to our introduction of a visualizable caricature below.

Effective parameter identification

The leading nonharmonic eigenvectors, $\boldsymbol{\phi}$, computed for the transient data $\mathbf{X}$ provide an intrinsic parameterization of this data set, i.e. a set of coordinates parameterizing the model manifold (see the discussion on Section S2A of the Supplementary Material for clarification of the term nonharmonics). However, they are not necessarily physically meaningful. In order to interpret them, the data scientist who knows their dimensionality can now ask a domain scientist to suggest a set of physically meaningful parameter combinations, κ_i, and try to quantitatively establish a one-to-one correspondence between the data-driven ϕ_i and the hypothesized meaningful κ_i. This approach to interpretability has been proposed and used in refs. (31–34) for the case of data-driven
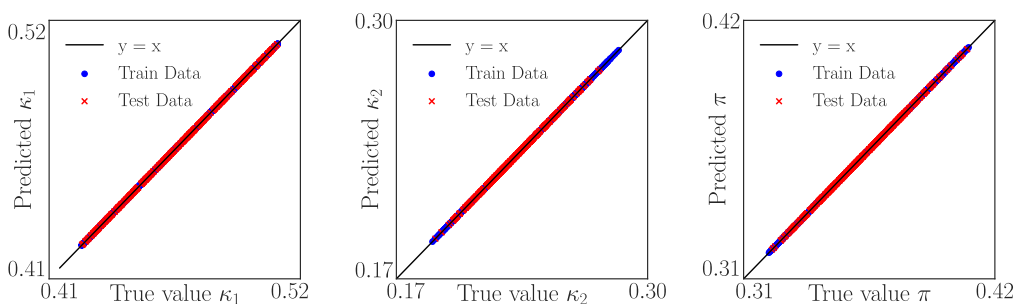
Fig. 4. The three theoretical effective parameters predicted as a functions of the DMaps eigenvectors ϕ with Double DMaps. (Left) $\kappa_1 = f_1(\phi)$, (middle) $\kappa_2 = f_2(\phi)$, and (right) $\pi = f_3(\phi)$. Blue dots denote the training points (7,000 data points) and red crosses the test points (3,000 data points).
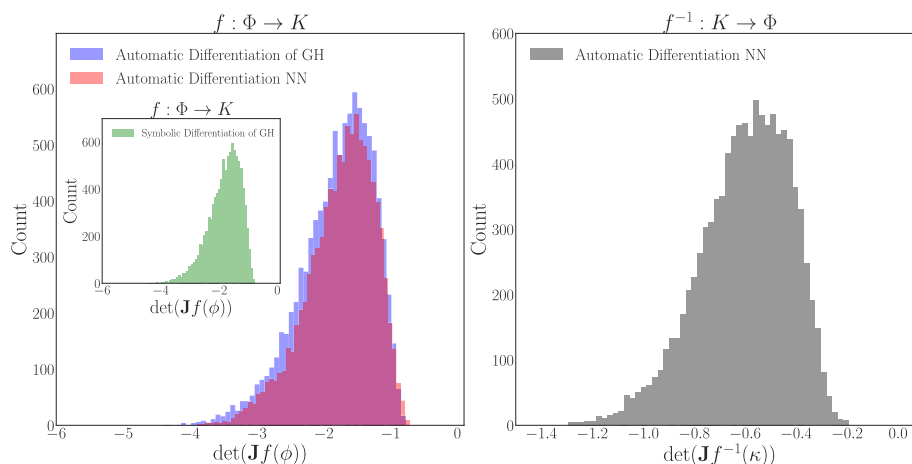
Fig. 5. (Left) Histograms of the determinant of the Jacobian, det $\mathbf{J}f(\phi)$, computed on each observed data point with automatic and symbolic differentiation of GH and with automatic differentiation using a neural network. (Right) Histogram of Jacobian determinants for the inverse function, det $\mathbf{J}f^{-1}(\kappa)$, computed with a neural network.

effective *variables*, and it can be extended, as we propose here, for data-driven effective *parameters*.

In our case, Yeung et al. have already provided us with good candidate analytical effective parameters $\kappa = (\kappa_1, \kappa_2, \pi) \in K \subset \mathbb{R}^3$. We seek a (hopefully smooth) invertible mapping $f: \Phi \to K$ from the DMaps space to the space of analytical effective parameter values and back. This mapping is constructed through a "slight twist" on GH, which we call Double DMaps, explained in Section S2C of the Supplementary Material. From the total 10,000 collected data points, we use 7,000 as training points and 3,000 as test points for our Double DMaps. We use the inverse function theorem (IFT) described in Section S2E of the Supplementary Material to check that our data-driven effective parameters are indeed *locally* one-to-one with the known analytical effective parameters (Eq. 2). We then use our Double DMaps GH to express the three theoretical effective parameters $\kappa = (\kappa_1, \kappa_2, \pi) \in K \subset \mathbb{R}^3$ as (approximate) functions of our coordinates ϕ.

An alternative realization of this map (data-driven effective to analytical effective) *and its inverse* can also be constructed through the "technology" of neural networks: We used the data-driven effective parameters as inputs in a neural network whose outputs are the analytical effective parameters. Specifically, we used a five-layer, fully connected network with 30 nodes per layer and tanh activation functions, which we optimized via **ADAM** to achieve a mean squared error (MSE) on the order of 10^{-6}. Training this network provides an alternative realization of the

mapping between the data-driven ϕ_i and the interpretable (here analytically obtainable) κ_i, the map $f: \Phi \to K$. We also obtained the inverse map, $f^{-1}: K \to \Phi$, by training a second neural network that implemented the same architecture and training scheme but with inputs and outputs switched. Instead of training two separate networks, one could combine the two networks into an autoencoder. Being able to construct the forward and the inverse mapping confirms the global one-to-one correspondence of the two sets on the data: The autoencoder would not be trainable otherwise. Figure 4 plots the ground truth values of the three effective parameters against those interpolated with GH.

To establish that this map $f: \Phi \to K$ is invertible, we first confirm that the determinant of its 3×3 Jacobian matrix is bounded away from zero for all points in our data set. By construction, f is continuously differentiable, so the IFT guarantees local invertibility in a neighborhood of any point $\phi \in \Phi$, where the Jacobian matrix

$$\mathbf{J}f(\phi) = \begin{bmatrix} \partial\kappa_1/\partial\phi_1 & \partial\kappa_1/\partial\phi_3 & \partial\kappa_1/\partial\phi_9 \\ \partial\kappa_2/\partial\phi_1 & \partial\kappa_2/\partial\phi_3 & \partial\kappa_2/\partial\phi_9 \\ \partial\pi/\partial\phi_1 & \partial\pi/\partial\phi_3 & \partial\pi/\partial\phi_9 \end{bmatrix}$$

is nonsingular. In Fig. 5, we illustrate that det $\mathbf{J}f(\phi)$ is bounded away from zero on our complete data set of 10^4 points. Furthermore, our success in training the decoder component indicates that $f: \Phi \to \mathbf{K}$ is globally invertible over our data and that our
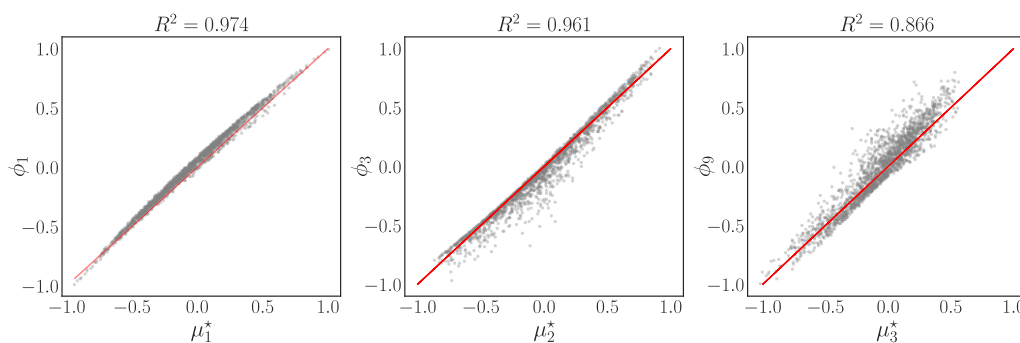
Fig. 6. The three eigenvectors, $\boldsymbol{\phi}$, are fitted as functions of the original parameters, $\boldsymbol{p}$, through a symbolic regression algorithm. Entries of $\boldsymbol{\phi}$ and $\boldsymbol{p}$ were rescaled in the range [− 1, 1]. Expressions for the $\mu^{\star}$ are provided in Eq. (4).

computed data-driven effective parameters are indeed one-to-one with the proposed theoretical ones [Eq. (2)].

The effective parameters proposed in ref. (24) were obtained by applying the QSSA to the full model. Simply by rearranging and simplifying the terms in Eq. (2), we could derive another equally plausible triplet of effective parameters:

$$\mu'_1 = [\text{E}]\frac{k_{\text{f},1}k_{\text{cat},1}}{k_{\text{r},1} + k_{\text{cat},1}} \quad \mu'_2 = \frac{k_4}{k_5 k_6}, \quad \mu'_3 = \frac{k_4}{k_6}. \quad (3)$$

Which of the two triplets would a symbolic regression package [e.g. gplearn (35)] select? We illustrate an answer graphically in Fig. 6 and analytically in Eq. (4). Note that, when performing this regression, we rescaled both the original parameters and the DMaps coordinates to lie in the range [−1, 1], as suggested in the package documentation (35):

$$\mu^{\star}_1 = 0.288(k_{\text{cat},2} - k_{\text{cat},1} + k_{\text{r},2} + k_{\text{f},1}),$$
$$\mu^{\star}_2 = 0.455(k_{\text{f},1} - k_{\text{f},2}),$$
$$\mu^{\star}_3 = 0.218(0.36k^2_{\text{f},1} - 1.38k_{\text{f},1}k_{\text{r},2} - k_{\text{f},2} + k_{\text{cat},2}$$
$$- k_{\text{f},1} - k_{\text{r},2} - 0.436), \quad (4)$$

where $\mu^{\star}_i$ denotes the ith estimated symbolic regression expression/parameter. As illustrated in Fig. 6, these simple linear or quadratic expressions of the original parameters $\boldsymbol{p}$ can fit the coordinates quite accurately. In our opinion, while they can be written down in terms of "simple cognitive basis functions," (i.e. monomials) ultimately these symbolically regressed parameters are almost as mechanistically uninterpretable as our data-driven effective ones.

Behavior estimation

Our computational formulation also allows us to obtain a mapping from new values of the effective parameters to the corresponding system output behavior. Each analytical effective parameter κ_i and each element of every observed behavior vector are functions over the intrinsic model manifold, which is parameterized by the data-driven effective parameters ϕ_i. If we are given a new triplet of ϕ_i, GH on our Double DMaps can recover any element of any observation vector. If, on the other hand, we are given a new triplet of κ_i, we need only locally invert the known $\kappa_i(\phi_j)$ functions to the data-driven effective parameters, and proceed as above to predict the corresponding dynamic behavior through GH. Alternatively, after a round of DMaps on the κ_i, we perform GH on these DMaps to interpolate any desirable element of the expected behavior vector as a function of the κ_i.

To implement this latter procedure, we generated 5,000 triplets of analytical effective parameters by perturbing uniformly within ±20% of the nominal parameter values $(\kappa_1, \kappa_2, \pi) = (0.467, 0.232, 0.362)$, designating 4,000 as training and 1,000 as test points. We used this data set to learn the output concentration profiles for 10 time steps of S_2 with our Double DMaps GH scheme. Figure 7 shows the true values of the concentrations against the predicted values with our scheme for $t = 10$. Across all 1,000 test points for analytical effective parameter values, the relative prediction error does not exceed 0.1%.

Parameter estimation

Even when the kinetic mechanism is known, parameter estimation is often challenging, due to measurement noise and differences in the timescales of individual reactions (24). Estimating the parameters not through optimization but through our data-driven scheme is straightforward from a technical standpoint. For previously unseen behaviors $\boldsymbol{f}(\boldsymbol{p}_{\text{new}}) = [S_2(t_1|\boldsymbol{p}_{\text{new}}), \ldots, S_2(t_f|\boldsymbol{p}_{\text{new}})]$, the Nyström extension (described in Section S2B of the Supplementary Material) directly estimates the corresponding ϕ_i on the model manifold, from which we directly go to the effective parameters $\boldsymbol{\kappa}$ leading to this behavior through our Double DMaps version of GH (see Section S2C). Our approach performs this estimation in the minimal required dimensions—the intrinsic, data-driven ones—that jointly parameterize the observed behavior *and* the meaningful input combinations that produce it. Figure 8 illustrates the projection of 100 previously unseen behaviors to the 3D manifold through the Nyström extension and quantifies how well we can estimate the effective parameters for those unseen behaviors through our scheme.

On the parameter combinations that do not matter

Having identified a data-driven effective parameterization of the model and constructed data-driven maps from behavior to effective parameters and back, we now need to complete the task by mapping behavior to the original, full parameter set. Clearly, this mapping is not one-to-one: For every observed behavior from the model, there exists *an entire level set* of the original parameter space consistent with this behavior—and with a single set of meaningful parameter combination values. For the optimized data in Fig. 3, we showed that an entire 3D level set exists in the original parameter space, for a given output behavior (and so, for a given set of effective parameters/meaningful parameter combinations). But this does not identify the parameter combinations *that do not matter*, that is, those which do not influence the resulting model behavior as one changes their values by moving
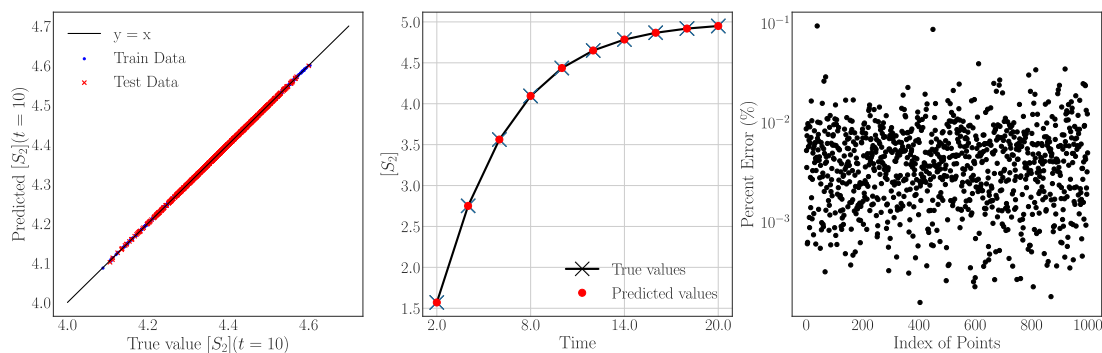
Fig. 7. (Left) Comparison of true and predicted values of the product concentration $[S_2]$ at $t = 10$ with our scheme for 4,000 training and 1,000 test points. (Center) A reconstructed concentration profile of S_2 for a test point. With crosses are illustrated the true values, and with red points the values predicted with Double DMaps. (Right) The relative error for the 1,000 unseen behaviors.
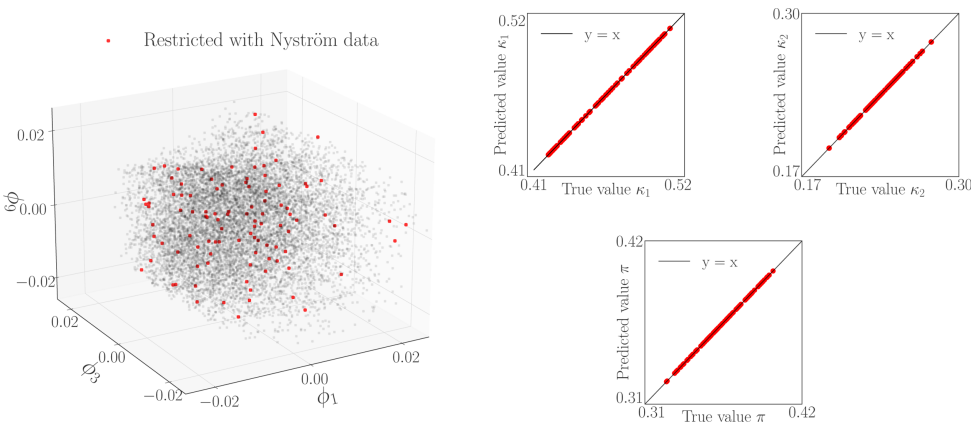

Fig. 8. (Left) The unseen behaviors $f(\boldsymbol{p}_{new})$ projected onto DMaps space via the Nyström extension. (Right) For 100 unseen behaviors, the effective parameters $(\kappa_1, \kappa_2, \pi)$ are predicted with our Double DMaps scheme from previously unseen behaviors $f(\boldsymbol{p}_{new})$.

along a level set of the effective parameters. In order to describe these level sets, we must employ a data-driven approach that allows us to detect the combinations of original parameters that do not affect the model output. This will disentangle the meaningful effective parameter combinations from the redundant ones. In Fig. 1, this disentangled parameterization was given by $\phi \equiv p_1 p_2$ and $\psi \equiv p_1^2 - p_2^2$.

Notice that the level sets of these two types of original parameter combinations are conformal everywhere. Moving p_1 and p_2 along the green level set does not change the model output, whereas moving them on the blue level set suffices to sample all possible output behaviors. In this way, the redundant parameter combinations allow us to construct the set of original, physical parameter values that are consistent with an observed behavior. Alternatively, holding them constant reduces the number of dimensions to be explored when optimizing the model behavior. Finally, after finding a behavior that optimizes a primary objective, the redundant parameter combinations help parameterize the search for an optimal *secondary* objective—not a Pareto multiobjective but rather a *lexicographic* optimization (36). This disentanglement helps outline the nature of this subsequent lexicographic optimization and the dimensions available for it in parameter space. However, since the data are collected locally around the base point, our computation provides only a *springboard* for further systematic exploration. A discussion of the systematic collection of additional data, parsimoniously extending the known "patches" of the level sets is discussed in ref. (37).

A visualizable caricature

The 3D level sets of our working MSP example do not lend themselves to visualization. We therefore turn to a simpler kinetic model to illustrate these ideas and methods:

$$S_0 + E \underset{k_r}{\overset{k_f}{\rightleftarrows}} ES_0 \overset{k_{cat}}{\rightarrow} S_1 + E, \qquad (5)$$
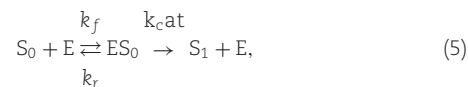
where S_0 and S_1 are two different states of the substrate S; E is the enzyme; and ES_0 and ES_1 are complexes between the enzyme and the substrate. The differential equations can be found in Section S1C of the Supplementary Material. We chose two base values of the original parameters k_f, k_r, k_{cat} to work with. The first base value

$$\boldsymbol{k}_1 = (k_f, k_r, k_{cat}) = (0.71, 19, 6700),$$

gives a single effective parameter $k_{eff} \simeq k_f$; in Section S3 of the Supplementary Material, we describe the discovery, through our manifold learning, of this single effective parameter and also the construction of its level sets. We choose to discuss here our results for the more interesting case of nominal parameters

$$\boldsymbol{k}_2 = (k_f, k_r, k_{cat}) = (0.97, 7000, 10000).$$
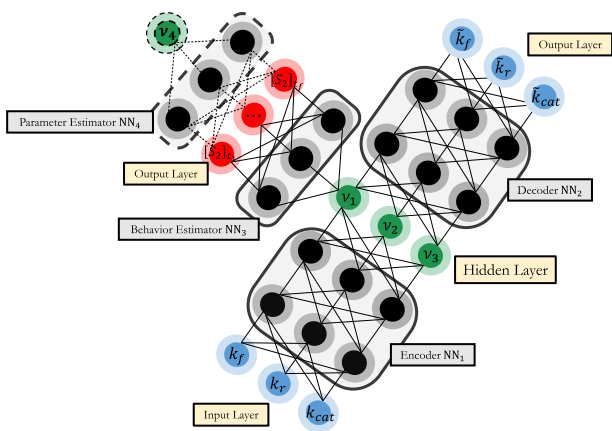
Fig. 9. The proposed Y-shaped Conformal Autoencoder consists of the following subnetworks: an Encoder (NN$_1$), a Decoder (NN$_2$), a Behavior Estimator (NN$_3$), and possibly an additional Parameter Estimator (NN$_4$) [see Eq. (7)].

In this regime, QSSA yields the single effective parameter

$$k_{\text{eff}} = E_{\text{tot}} \frac{k_f k_{\text{cat}}}{k_r + k_{\text{cat}}}, \tag{6}$$

where E_{tot} is the total concentration of the enzyme.

We generated 2,000 parameter vectors by sampling each entry uniformly within ±20% of its nominal value. We collected output system behaviors for each parameter vector by integrating the model mechanism of Eq. (5) from the reference initial condition $([S_0], [E], [S_1], [ES_1]) = (5.0, 0.66, 0, 0)$. The response is recorded every 2 seconds in time for five total points per trajectory. Our data-driven approach again detects that the output behavior of the system is intrinsically 1D, and the new single effective parameter ψ_1 is one-to-one with our data-driven effective parameter k_{eff}, which is a combination of all three original parameters. The level sets of ψ_1 (or k_{eff}) are 2D curved surfaces (manifolds) in the original parameter space. In order to describe this level set, that is, discover the redundant parameter combinations, we introduced a *Conformal Autoencoder* Y-shaped Neural Network architecture (see Fig. 9).

Our Y-shaped Neural Network scheme consists of several connected subnetworks:

$$NN_1 : (k_f, k_r, k_{\text{cat}}) \mapsto (\nu_1, \nu_2, \nu_3),$$
$$NN_2 : (\nu_1, \nu_2, \nu_3) \mapsto (\tilde{k}_f, \tilde{k}_r, \tilde{k}_{\text{cat}}),$$
$$NN_3 : \nu_1 \mapsto ([S_2]|_{t_1}, \ldots, [S_2]|_{t_f}),$$
$$NN_4 : ([S_2]|_{t_1}, \ldots, [S_2]|_{t_f}) \mapsto \nu_4. \tag{7}$$

We used three multilayer perceptrons illustrated in Fig. 9:

(1) "Encoder" (NN$_1$), which transforms the original parameters to a reparameterization, disentangling their meaningful combinations (one in the figure) and the redundant ones (two in the figure);
(2) "Decoder" (NN$_2$), which reconstructs the original parameters; and
(3) "Behavior Estimator" (NN$_3$), which maps the meaningful combination(s) to the observed output data.

An additional "Parameter Estimator" (NN$_4$) could be used to map observed behaviors back to the effective parameter(s) to ensure global invertibility.

The key feature is the loss function, consisting of several parts. The obvious one is the successful reconstruction of the input original parameters (the "Autoencoder" part). Next comes the ability of NN$_3$, whose input is the single effective parameter combination we seek, to reproduce the observed output; this forces ν_1 to be one-to-one with the analytically known parameter k_{eff}. How many output measurements are necessary? Whitney's (and Takens') embedding theorems provide guarantees for $2n + 1$ generic observations, when n is the dimension of the model manifold (26). Clearly, to build the architecture, we need to know in advance the number (here, one) of the required meaningful parameter combinations from the dimensionality of the model manifold. This number is the first quantity we compute with our output-informed DMaps analysis of the transient system observations. The third necessary loss component comes from further imposing an *orthogonality constraint* on the Conformal Autoencoder's latent coordinates ν:

$$\langle d\nu_i, d\nu_j \rangle = 0 \qquad \forall i \neq j,$$

where $d\nu_i$ indicates the vector of partial derivatives of the latent coordinate ν_i in terms of the input parameters (k_f, k_r, k_{cat}) and $\langle \cdot, \cdot \rangle$ indicates the inner product. This constraint is imposed using the automatic differentiation capabilities of the relevant code libraries and aims to disentangle what matters from what does not, making the architecture a "Conformal Autoencoder." We explain the procedure used to train this Neural Network in Section S2G of the Supplementary Material.

We thus discover a parameterization of the two redundant parameter combinations through ν_2 and ν_3. We also discover the Neural Network encoding of the effective parameter, ν_1, which is one-to-one with both k_{eff} and ϕ_1 (see Fig. 10). Our Double DMaps can easily approximate the estimation of ν_1 from new, unobserved behavior. Figure 10 shows representative (orthogonally) intersecting level sets of the three ν_i, and the conformal grid of ν_2, ν_3 on a level set of the effective parameter ν_1.

This network can be used to encode a full set of initial parameter values to the effective parameter values that matter and through them to the predicted behavior. More importantly, the already established path from the new, unobserved behavior to the corresponding value of ν_1, the effective parameter that matters, allows us to fix this value as an input to the Decoder NN$_2$ and reproduce the level set of original parameters consistent with this new observed behavior by varying the values of ν_2, ν_3.

JSF extraction

We conclude this section by discussing how a kernel-based method called JSFs, introduced by Dietrich et al. (23), can be extended and used to disentangle input–output relations. Instead of a Neural Network architecture, the "Jointly Smooth Functions" (23) approach, as its name suggests, could be used to find functions of the original parameters and functions of the output measurements that are *jointly smooth* over the available data. Those JSFs between the original parameters and the output are the effective parameters of the model in our case.

Figure 11 illustrates the results for our second, visualizable example. Two data sets are collected, containing 2,000 samples each. One consists of 20 time-delayed measurements of four output variable observations, (S_0, S_1, ES_0, E), which we express as $\mathbf{x}_1 \in \mathbb{R}^{80}$. The second contains the corresponding parameters $\mathbf{x}_2 \in \mathbb{R}^3$. We use these two data sets as input to the JSF extraction pipeline (Algorithm in the Supplementary Material) and compute 25 such functions. The first JSF is one-to-one with the known effective parameter k_{eff} (bottom left). We additionally plot an output (here,
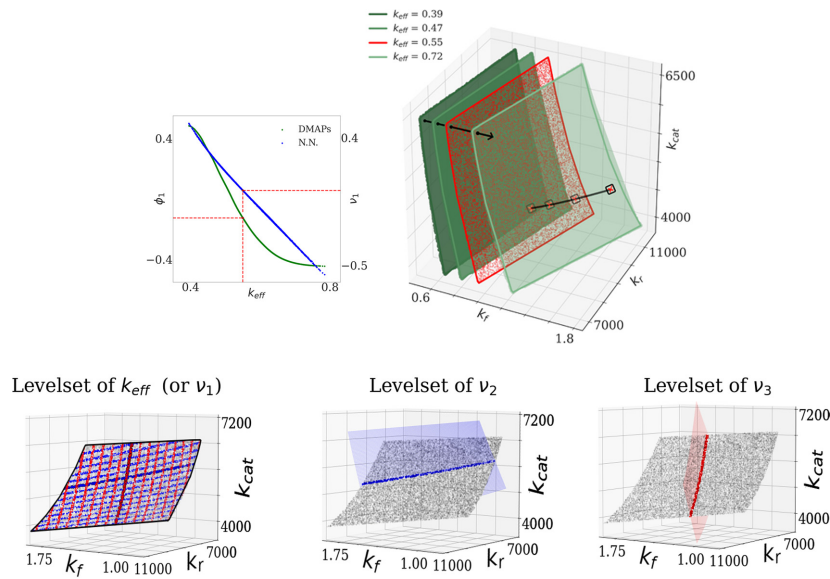
Fig. 10. (Top left) The effective parameter k_{eff} is one-to-one with the data-driven coordinate ϕ_1, and also with the Neural Network effective variable ν_1. (Top right) The level sets of constant behaviors, the level sets here are surfaces of the form $f(k_f, k_r, k_{cat}) = C$. A particular effective parameter (red point) corresponds to a level set (red surface) of the original parameters (k_f, k_r, k_{cat}). (Bottom left) The same level set of k_{eff} (equivalently, of ν_1, since they are one-to-one), on which the conformal directions are colored as a grid of red and blue lines. (Bottom center) The intersection of the level set of k_{eff} with a level set of ν_2. (Bottom right) The intersection of the level set of k_{eff} with a level set of ν_3.
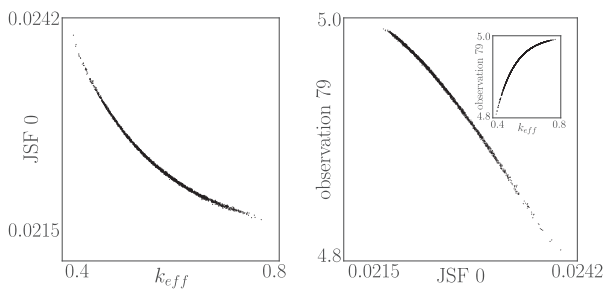
Fig. 11. (Left) The first JSF for the second example, compared to the effective parameter k_{eff}. (Right) The first JSF is one-to-one with the observation $\mathbf{x}_1^{(79)}$.

one of the measurements, the 79th one in time) that is also one-to-one with the first JSF (on the right). Note that, to test the robustness of the approach, the latter half of the output measurements were substituted with random noise uniformly distributed over the measurement range.

In our work, we introduce an additional feature of the JSFs that allows to the computation of *redundant* parameter combinations through the JSF approach; this is illustrated in Section S2H of the Supplementary Material.

Discussion

We have presented a systematic, data-driven approach for obtaining a meaningful reparameterization of parameter-dependent dynamical systems, disentangling the parameter combinations that matter to the output observations (temporal state measurements) from those that do not. The approach is generally applicable to the reparameterization of input–output relations.

We used manifold learning techniques, including DMaps, to jointly parameterize the behaviors observed (the "model manifold") and the parameter combinations leading to them. We found

the minimal number of meaningful parameter combinations (the effective parameters), expressed the outputs as functions of these effective parameters, and showed how to construct data-driven mappings from new effective parameters to the estimated outputs (prediction) and from new output observations back to effective parameters (estimation). It is worth mentioning that, in the case of noisy outputs, the DMaps parametrization will be robust to output noise as long as the scale parameter $\sqrt{\varepsilon}$ remains larger than the amplitude of the noise (19).

Disentangling the parameter combinations that affect the output from those combinations that do not (the redundant parameter combinations) was obtained through a conformal autoencoder neural network. This allows us to now provide, for any observed behavior, not only the effective parameter values for it but also *the level set, in full input parameter space* consistent with this behavior. The capability of disentangling meaningful from redundant by enforcing conformality seems a promising research tool in tasks ranging from data-driven dimensional analysis to the exploration and construction of closures, and to the training of overparameterized neural networks.

We briefly discuss the computational scalability of our approach. Generally, the ambient space dimension of the data influences the computational complexity less than the intrinsic dimension of the model manifold, i.e. the number of effective parameters. The detection of effective parameters in an intrinsically high-dimensional (say, five- or more dimensional) model manifold is less constrained by the scaling of our approach, but hinges on the large amount of data needed to sample the manifold well. Ambient space dimension, i.e. the number of given parameters (including redundant ones) as well as the number of observations, does not matter as much for the computational complexity of our approaches, since DMaps, GH, and JSFs are all based on pair-wise distance matrices that effectively ignore ambient dimension. The computational efficiency of the JSF approach is discussed in ref. (23). In general, kernel-based methods such as DMaps require more careful numerical implementations than Neural Network

approaches, otherwise the number of data points becomes a bottleneck. Efficient algorithms that scale to millions of data points, even in high-dimensions, are available; see refs. (23) and (38) for a discussion. Regarding memory, the Conformal Autoencoder network is less demanding than kernel-based approaches, because we can utilize minibatching for training and highly parallelized software with efficient implementations is readily available. The analysis of the computational complexity of the network approach is much more involved than for kernel-based approaches, however, and out of the scope of this paper. Even convergence of the training is not clear, although some recent work hints on global convergence at least in controlled settings (39, 40).

It is interesting to consider the interplay of this approach with multiobjective optimization: If some input parameter combinations matter to a dominant objective, while others do not, we can, after a first round of optimization, exploit the redundant parameter combinations and optimize a second, "subservient" objective on optimal level sets of the first, dominant one. This is termed *lexicographic* optimization and can also be related to "lifelong learning." A conceptually simple example is the training of an overparameterized neural network to perform some task: The primary objective will be the accuracy of the prediction, while the "subservient," secondary objective can be the pruning of the network for sparsity *while remaining on the level set of successfully optimized predictions*.

Finally, we explored interpretability of our data-driven effective parameters through establishing bijections between them and candidate "tuples" of physical ones, which must come from domain experts. We also explored another simple approach to effective parameter interpretability by symbolically regressing the data-driven effective parameters as functions of the input ones.

This work, creating mappings between parameters (in a sense, inputs to a dynamical system) and observed behavior (outputs), can be extended to create mappings between inputs and states, as well as mappings between states and outputs. We are exploring this direction toward data-driven balanced realizations. We expect that our level set parameterizations of the parameter sets that matter/do not matter (whether through Conformal NN or through JSF computations) may lead to useful extensions of the controllability and observability subspaces of linear theory. In this more general problem formulation, one can go beyond structurally unidentifiable inputs, and uncover spurious observations that are not system outputs (e.g. intrinsic sensor noise in our output observations) (41). We are also exploring JSFs as a promising alternative kernel-based approach. Extracting the components of the inputs and outputs in the jointly smooth directions "that matter" can also help highlight those that do not. A key benefit is that, in addition to removing irrelevant input directions, this computation also removes output directions that are not influenced by the input (parameter) data, and provides a numerically stable and accurate approximation of the function space over the space of the effective parameters.

Conclusion

We conclude by reiterating that, while the paper was focused on parameter nonidentifiabiity, in a context where the original model parameters function as "inputs" to the model, and the observed state time series are the "output," our approach is generally applicable to data-driven (re)parametrization of more general input–output relations, with an eye toward disentangling meaningful inputs from redundant ones. Applicability of our current framework in an experimental setting involves (after selection of a reference set of conditions) the systematic local perturbation of all distinct experimental parameters/inputs; data mining on the response/output then leads to the discovery of the meaningful and redundant parameter combinations.

Supplementary Material

Supplementary material is available at *PNAS Nexus* online.

Authors' Contributions

N.E., N.J.W., G.A.K., F.D., and I.G.K. designed and performed research. M.K. and S.M. provided data and models. N.E., N.J.W., F.D., and I.G.K. wrote the manuscript.

Previous Presentation

Parts of these results were previously presented at 2021 Virtual AIChE Annual Meeting, November 17, 2021.

Preprints

A preprint of this article can be found at arXiv:2110.06717.

Data Availability

The data underlying this article and the codes used to perform the computations are available in a public repository from the authors at https://gitlab.com/nicolasevangelou/on_the_parameters.

References

1. Goussis DA, Maas U., 2011. Model reduction for combustion chemistry. In: Echekki T., Mastorakos E., editors. Turbulent combustion modeling. Dordrecht: Springer, p. 193–220.
2. Kopf A, Claassen M., 2021. Latent representation learning in biology and translational medicine. Patterns. 2(3): 100198.
3. Snowden TJ, van der Graaf PH, Tindall MJ., 2017. Methods of model reduction for large-scale biological systems: a survey of current methods and trends. Bull Math Biol. 79(7): 1449–1486.
4. Benner P, Gugercin S, Willcox K., 2015. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. 57(4): 483–531.
5. Quarteroni A, Rozza G, *et al*. 2014. Reduced order methods for modeling and computational reduction. volume 9. Cham: Springer.
6. Chiavazzo E, Gear CW, Dsilva CJ, Rabin N, Kevrekidis IG., 2014. Reduced models in chemical kinetics via nonlinear data-mining. Processes. 2(1): 112–140.

7. Deane A, Kevrekidis I, Karniadakis GE, Orszag S., 1991. Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders. Phys Fluid A: Fluid Dyn. 3(10): 2337–2354.

8. Foias C, Jolly MS, Kevrekidis IG, Sell GR, Titi ES, 1988. On the computation of inertial manifolds. Phys Lett A. 131(7): 433–436.

9. Nadler B, Lafon S, Coifman RR, Kevrekidis IG., 2006. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. Appl Comput Harmon Anal. 21(1): 113–127.

10. Shvartsman SY., 2000. Order reduction for nonlinear dynamic models of distributed reacting systems. J Process Contr. 10(2-3): 177–184.

11. Barenblatt GI., 1996. Scaling, self-similarity, and intermediate asymptotics. Cambridge: Cambridge University Press.

12. Cole D, 2020. Parameter redundancy and identifiability. New York (NY): CRC Press.

13. Raue A., 2009. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. Bioinformatics. 25(15): 1923–1929.

14. Brouwer AF, Eisenberg MC., 2018. The underlying connections between identifiability, active subspaces, and parameter space dimension reduction. arXiv:1802.05641v1.

15. Gutenkunst RN., 2007. Universally sloppy parameter sensitivities in systems biology models. PLoS Comput Biol. 3(10): e189.

16. Transtrum MK, Qiu P., 2014. Model reduction by manifold boundaries. Phys Rev Lett. 113(9): 098701.

17. Constantine PG., 2015. Active subspaces: emerging ideas for dimension reduction in parameter studies. Philadelphia, PA: SIAM.

18. Riolo R, Vladislavleva E, Moore JH., 2011. Genetic programming theory and practice IX. New York (NY): Springer Science & Business Media.

19. Coifman RR, Lafon S., 2006. Diffusion maps. Appl Comput Harmon Anal. 21(1): 5–30.

20. Holiday A., 2019. Manifold learning for parameter reduction. J Comput Phys. 392: 419–431.

21. Achille A, Soatto S., 2018. Emergence of invariance and disentanglement in deep representations. J Mach Learn Res. 19(1): 1947–1980.

22. Locatello F, *et al.* Challenging common assumptions in the unsupervised learning of disentangled representations. In: Proceedings of the 36th International Conference on Machine Learning PMLR. p. 4114–4124.

23. Dietrich F, Yair O, Mulayoff R, Talmon R, Kevrekidis IG., 2020. Spectral discovery of jointly mmooth features for multimodal data. SIAM Journal on Mathematics of Data Science 4(1): 410–430

24. Yeung E., 2020. Inference of multisite phosphorylation rate constants and their modulation by pathogenic mutations. Curr Biol. 30(5): 877–882.

25. Bodenstein M., 1913. Eine theorie der photochemischen reaktionsgeschwindigkeiten. Zeitschrift für Physikalische Chemie. 85(1): 329–397.

26. Takens F., 1981. Detecting strange attractors in turbulence. In: Rand D., Young LS., editors. Dynamical systems and turbulence. Berlin: Springer, p. 366–381.

27. Sethna J., Sloppy Models. https://sethna.lassp.cornell.edu/research_area/sloppy_models [accessed 2022-09-01].

28. Transtrum MK, Machta BB, Sethna JP, 2010. Why are nonlinear fits to data so challenging? Phys Rev Lett. 104(6):060201.

29. Aoki K, Takahashi K, Kaizu K, Matsuda M., 2013. A quantitative model of ERK MAP kinase phosphorylation in crowded media. Sci Rep. 3(1): 1–8.

30. Dsilva CJ, Talmon R, Coifman RR, Kevrekidis IG., 2018. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study. Appl Comput Harmon Anal. 44(3): 759–773.

31. Frewen TA, *et al.* Coarse collective dynamics of animal groups. In: Gorban AN., Roose D., editors. Coping with complexity: model reduction and data analysis. Heidelberg, Germany: Springer. p. 299–309.

32. Kattis AA, Holiday A, Stoica AA, Kevrekidis IG, 2016. Modeling epidemics on adaptively evolving networks: a data-mining perspective. Virulence. 7(2): 153–162.

33. Meila M, Koelle S, Zhang H., 2018. A regression approach for explaining manifold embedding coordinates. arXiv:181111891.

34. Sonday BE, Haataja M, Kevrekidis IG., 2009. Coarse-graining the dynamics of a driven interface in the presence of mobile impurities: effective description via diffusion maps. Phys Rev E. 80(3): 031102.

35. Stephens T., Genetic programming in Python with scikit-learn inspired API: gp-learn. https://github.com/trevorstephens/gplearn. Updated: 2022-08-04. [Accessed: 2022-09-01].

36. Arora J., 2004. Introduction to optimum design. San Diego, CA, Elsevier.

37. Chiavazzo E., 2017. Intrinsic map dynamics exploration for uncharted effective free-energy landscapes. Proc Natl Acad Sci. 114(28): E5494–E5503.

38. Shen C, Wu HT., 2020. Scalability and robustness of spectral embedding: landmark diffusion is all you need. arXiv:200100801.

39. Jacot A, Gabriel F, Hongler C., 2018. Neural tangent kernel: convergence and generalization in neural networks. Advances in Neural Information Processing Systems 31Montréal, Canada.

40. Rotskoff GM, Vanden-Eijnden E., 2018. Trainability and accuracy of neural networks: an interacting particle system approach. arXiv:180500915.

41. Talmon R, Wu HT., 2019. Latent common manifold learning with alternating diffusion: analysis and applications. Appl Comput Harmon Anal. 47(3): 848–892.