

Scientific Machine Learning of Chaotic Systems Learns Reduced-Order Equations for Neural Populations

Anthony G. Chesebro^{1,2}, David Hofmann^{1,3}, Vaibhav Dixit³, Earl K. Miller⁴, Richard H. Granger⁵, Alan Edelman³, Christopher V. Rackauckas^{3,*}, Lilianne R. Mujica-Parodi^{1,2,3,6,†} and Helmut H. Strey^{1,2,3,‡}

¹*Department of Biomedical Engineering and Laufer Center for Physical and Quantitative Biology, State University of New York at Stony Brook, Stony Brook, New York, USA*

²*Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital and Harvard Medical School, Charlestown, Massachusetts, USA*

³*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*

⁴*Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*

⁵*Department of Psychological and Brain Sciences, Dartmouth College, Hanover, New Hampshire, USA*

⁶*Santa Fe Institute, Santa Fe, New Mexico, USA*



(Received 27 May 2026; accepted 13 August 2026; published 8 October 2026)

Extracting interpretable mathematical models from complex dynamical systems is difficult, especially for chaotic dynamics observed with noisy experimental data. We present PEM-UDE, a method that combines prediction-error methodology with universal differential equations to discover governing equations from limited, noise-corrupted observations. Prediction-error feedback smooths the chaotic optimization problem; for noise-free data generated within the model class, it preserves the data-consistent zero-loss set, whereas noise and model misspecification introduce a gain-dependent stability-bias trade-off. Preservation of the zero-loss set is not a guarantee of unique structural identifiability. We test the method on two benchmark chaotic systems—the Rössler attractor and a real electrical circuit—and recover the correct functional forms even when one observed dimension contains noise of 5 times the signal magnitude. The method also accepts prior knowledge of the system as an initial functional form, which we use to learn neural circuit equations that account for sparse connectivity, a feature missing from conventional neural mass models. Applied to a population of Izhikevich neurons, PEM-UDE yields a multiscale neural mass model that ties single-neuron parameters to macroscopic network dynamics and predicts a relationship between connection density, dominant oscillation frequency, and synchrony. We test these predictions against three intracranial recording datasets from rat and human cortices. For the neuroscience application, the learned equations are a reduced-order closure for a specified simulated Izhikevich network family; the experimental recordings provide an indirect consistency check of predicted frequency and synchrony trends, not a direct fit of the equations to recordings.

DOI: [10.1103/xtb5-mlc8](https://doi.org/10.1103/xtb5-mlc8)

I. INTRODUCTION

Discovering the equations that govern complex systems has been an objective of dynamics for much of the past century, and typically requires both expert knowledge of the application domain and some understanding of statistical mechanics [1–5]. The approach has produced results across many fields, from climate models [6,7] to chemical kinetics [8,9], human social interactions [10], and neural computation [11]. Recent work has produced automated methods that

assist the normally expert-driven process of describing data with systems of differential equations [12–15]. Symbolic regression techniques, including sparse identification methods (SINDy; [12]) and genetic algorithms [16], fit equations directly from data, but typically work less well with noisy data [17]. Universal differential equations (UDEs) combine ideas from machine learning and scientific computing [18], embedding artificial neural networks in differential equation frameworks to learn unknown dynamics directly from data. The training of the network is constrained in two ways: by an imposed known functional form on the dynamics, with additional terms learned around it [18]; and by structural constraints on the learning process itself [19]. The contribution here is therefore specific: We do not introduce observers, UDEs, or symbolic regression as separate ideas, but combine observerlike prediction-error feedback with UDE training so that chaotic trajectories can be fit before the learned residual field is converted to an explicit symbolic expression.

*Contact author: crackauc@mit.edu

†Contact author: mujica@lcneuro.org

‡Contact author: helmut.strey@stonybrook.edu

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

These approaches struggle with chaotic systems, which are common in neuroscience and other biological contexts [20–23]. Even simple physical systems can exhibit complex, chaotic solutions when certain parameters are brought past a bifurcation point [24], and neural dynamics often display characteristics of chaotic oscillators [22,25]. The presence of chaos introduces numerical problems for the fitting procedure: The trajectory of a chaotic system is not well defined beyond a Lyapunov time. Any numerical ordinary differential equation (ODE) solver introduces truncation error at each step, and sensitive dependence on inputs means that solver output will have an $\mathcal{O}(1)$ error in the produced trajectory shortly after a Lyapunov time. The only guarantee that the numerical simulation gives a meaningful trajectory is provided by the shadowing lemma, which states that there exists an ϵ ball around the initial condition in which the numerical solver's trajectory matches the trajectory of one of the other potential initial conditions within the ϵ ball, effectively describing the density of solutions around the attractor [26]. The shadowing lemma has two consequences that make chaotic identification difficult. The divergent nature of chaotic trajectories creates instability in all sensitivity analysis methods used to compute gradients. Standard automatic differentiation through solvers then produces inaccurate derivative estimates unless specialized modifications are applied [27–31]. Consequently, the standard method for calibration of trajectories (fitting a loss function defined by numerical solutions of the ODE solver) is not a convergent algorithm in the context of chaotic systems. Noisy observational data can also produce a loss function that cannot distinguish between nearby trajectories during training, leading to inaccuracies in the fit dynamics [26]. With exponential dependence of the solution on initial parameters, infinitesimal changes to the data may land the data point exponentially further along the time-series trajectory. Because of these properties, the long-term time series of a chaotic system can only be interpreted probabilistically, based on statistical or ergodic properties [32–34]. The traditional technique of fitting a time series via the L_2 norm is therefore ill-defined in chaotic contexts. In real-world models, we additionally only have partial observability, which makes parameter fitting for any dynamical system difficult. This is particularly true in neuroscience, where neural interactions form complex systems operating across multiple scales: from molecular processes such as neurotransmitter activity to brain-wide electric fields that can be observed using electroencephalography (EEG) [21,35,36].

Beyond the complexity inherent in describing these multiscale interactions, brain dynamics are often poised near critical points that allow switching between different dynamic states depending on task requirements, metabolic resources, and the overall health of the neural circuits [37–39]. The operation near critical points gives flexibility in information processing, but it also makes neural dynamics sensitive to small perturbations [22,40]. Traditional approaches to modeling neuron populations, termed neural mass models, focused on capturing specific macroscale properties without a systematic approach to accounting for microscale biophysical parameters such as ion channel conductance [41–43]. Next-generation neural mass models (NGNMMs) have since provided an analytical expression for population-level activ-

ity that preserves the microscale detail present in individual neurons while also capturing emergent macroscale properties such as time-varying synchrony [44–47]. These analytical approaches typically rely on the Ott-Antonsen ansatz [48] to avoid a Fokker-Planck derivation of the mean-field dynamics [49,50]. A caveat is that these derivations capture many desirable multiscale dynamics but rely on nonbiophysical assumptions, in particular, all-to-all connectivity between neurons [44] and current conservation throughout the system [50]. Recent work has extended these approaches to account for fluctuations arising from sparse connectivity and noise in networks of quadratic integrate-and-fire neurons [51]. Such extensions remain limited to simplified neuron models that lack biophysically detailed adaptation and recovery dynamics. Extending analytical mean-field derivations to biophysically detailed neuron models (e.g., Izhikevich neurons with adaptation currents), while at the same time relaxing connectivity assumptions, remains an open problem and motivates data-driven approaches.

The disconnect between the underlying neurobiology and the assumption of fully connected networks that grounds these derivations is consequential. The assumption is approximately true for deep and subcortical structures [45], but far from physiological in the cortex, where only 1%–5% connectivity is typical [52]. Cortical sparsity is not accidental: Several lines of evidence indicate that the sparse, structured connectivity patterns in the cortex are essential for the structured information flow that supports higher cognitive processes [53,54]. Accounting for sparsity in equations that capture population activity is therefore needed for accurate models of the brain.

To address these limitations, we present a method for training UDEs to fit chaotic system dynamics, and we use it to learn reduced-order closure terms for simulated neural-population models across connectivity levels. We describe the procedure, which uses the prediction-error method (PEM) [19,55] to train the UDE on chaotic system data, and then extracts a symbolic form of the learned expression from the UDE to generalize the dynamics. We show that this approach can learn unknown states in a chaotic system, accurately recovering a hidden state in the Rössler system that traditional UDE approaches cannot, and that it does so by removing the sensitive dependence on parameter choices from the system. The approach also learns the dynamics of a chaotic electrical circuit under incomplete observation, where one dimension of the observed data is corrupted by noise of 5 times the magnitude of the original signal. Under these conditions, sequential thresholding least squares (STLSQ) applied directly to the data fails to recover the correct dynamic system, while PEM-UDE succeeds; we therefore argue that the method is more data efficient than pure symbolic regression. Finally, from simulated firing-rate and voltage observations of a spiking network, we learn a reduced-order correction to an existing neural mass model. The symbolic correction generalizes across simulated connection probabilities and reveals how connectivity changes can alter the rhythmic dynamics of this network family. The later comparison with recordings tests whether these qualitative predictions match regional physiology; it does not claim direct equation discovery from the recordings. These controlled comparisons isolate the contribution of PEM relative to an otherwise identical UDE and

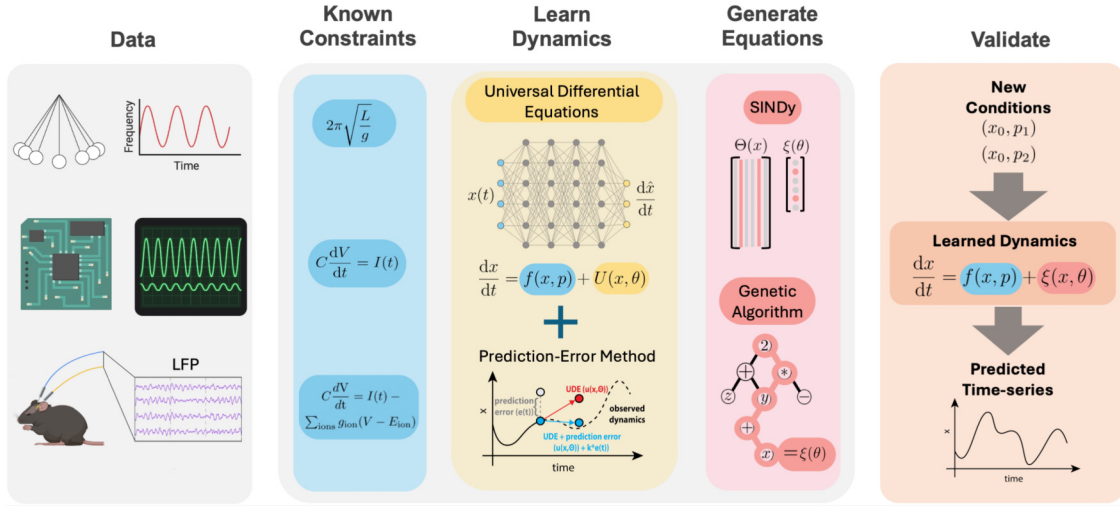


FIG. 1. Schematic of the PEM-UDE approach. We begin with observations of a physical or biological system that have time-varying activity (data). From these, and from prior work, we form an initial description of the system under observation (known constraints). We then fit a UDE to describe the part of the dynamics not covered by this initial estimate, using the prediction-error method to help train the UDE on chaotic systems (learn dynamics). We then extract a symbolic form of the dynamics described by the UDE (generate equations) and check that this symbolic form correctly predicts activity beyond the initial observations (validate).

direct STLSQ under the conditions studied; they are not a comprehensive benchmark of all methods for chaotic system identification. A graphical overview of the full pipeline is provided in Fig. 1.

II. RESULTS

A. Formal description of the PEM-UDE approach

We give a formal definition of the combined PEM-UDE/symbolic regression method here, with implementation details in Sec. IV and the accompanying code. The associated repository includes a demonstration using toy data that runs on standard computers, for readers who want to try the method out.

A graphical overview of the approach is provided in Fig. 1. We begin with observations of a physical or biological system that result in a dataset of related time series. The goal is a system of differential equations that describes the dynamics of the system under observation and accurately predicts future dynamics. From the observations and previous work on similar systems, we make an initial guess for the dynamics, using prior information from physical systems to structure the first approximation of the system's behavior [18]. We then set up a UDE system to learn the unknown dynamics directly from the observed data. Standard methods for training UDEs struggle to fit chaotic systems accurately, as shown in the results below. To get around this, we developed a training approach that integrates PEM dynamics with the UDE during simulations, so that the UDE can learn from the data. Once training is complete, we extract a symbolic representation of the dynamics from the UDE network with either sparse linear solver-based symbolic regression (STLSQ) or evolutionary algorithm-based symbolic regression [16], depending on the specific context and data availability. The symbolic description lets us both predict future time series and make

analytical observations about the underlying dynamics. The prediction-error term is a training device: The symbolic equations are evaluated as autonomous dynamics, after the observer correction has been reduced or removed, so success requires the learned vector field itself to reproduce the observed dynamical structure.

1. UDE formulation

Following prior work [18], a universal differential equation with an additive correction of a general process $\mathbf{x}(t)$ with some observable state $\mathbf{y}(t)$ is given by

$$\dot{\mathbf{x}} = f_p(\mathbf{x}, \mathbf{u}(t)) + NN_\theta(\mathbf{x}), \quad (1a)$$

$$\mathbf{y}_\theta(t) = h_q(\mathbf{x}_{p,\theta}(t; \mathbf{x}_0), t). \quad (1b)$$

Here, we use bold variables to denote vectors. f_p are the known (or assumed) dynamics of the system with parameters p , $\mathbf{x}_{p,\theta}(t; \mathbf{x}_0)$ denotes the solution of those dynamics with initial conditions $\mathbf{x}_0$ at parameter values p and θ , $\mathbf{u}(t)$ is an external stimulus, $h_q(\mathbf{x}_{p,\theta}(t; \mathbf{x}_0), t)$ is the observation function parametrized by q , which can explicitly depend on time (for instance, due to noise processes or the influence of the external stimulus on the observation), and $NN_\theta(\mathbf{x})$ is the universal approximator network trained to learn the unknown dynamics of the system based on its parameters θ . For notational simplicity, we drop the parameter dependence of the observation $\mathbf{y}_\theta(t)$ on parameters p and q and initial conditions $\mathbf{x}_0$, but keep the network parameters θ , since we want to optimize those while keeping the rest fixed. We focus on systems of ordinary differential equations in this work, but the approach extends to stochastic and delayed differential equation systems, as has been done in previous work using UDEs [18]. For an observed dataset $\{\hat{\mathbf{y}}_i\}_{i=1, \dots, N}$, training the UDE NN_θ uses a cost function $C(\theta)$ that minimizes the error at every observed time point t_i

such that

$$\arg \min_{\theta} C(\theta) = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_{\theta}(t_i) - \hat{\mathbf{y}}_i)^2, \quad (2)$$

where $\mathbf{y}_{\theta}(t_i)$ is the observed solution of the ODE with neural network weights θ at the measurement times t_i . In all equations that rely on observational data, we denote any observed data for $\mathbf{x}$ as $\hat{\mathbf{x}}$. We chose $C(\theta)$ to be the mean-squared error in this work, but other cost functions are appropriate for different scenarios, depending on how measurement noise affects the measured signal. We then use the prediction-error method to train UDEs on this loss function for chaotic systems.

2. PEM formulation

The PEM is a standard optimization technique for fitting parameters of dynamical systems to observational data [19,56]. Similarly to Eq. (1), we consider a general process $\mathbf{x}(t)$ that gives rise to some observable state $\mathbf{y}(t)$ with some measurement noise $\mathbf{e}(t)$:

$$\dot{\mathbf{x}} = \mathbf{f}_p(\mathbf{x}, \mathbf{u}(t)), \quad (3a)$$

$$y_{p,i}(t) = m_{q,i}(x_{p,i}(t)) + e_i(t). \quad (3b)$$

In this statement, $\mathbf{x}$ is the state of a time-varying system with parameters p that we wish to fit. The observed state $\mathbf{y}$ is defined by a measurement function m of the process $\mathbf{x}(t)$, parametrized by q , and a measurement error $\mathbf{e}(t)$. Here, we assume that there is a one-to-one mapping of $\mathbf{x}$ to $\mathbf{y}$ (i.e., there are no terms in $\mathbf{y}$ that depend on multiple elements of $\mathbf{x}$). For a more general discussion of the conditions under which PEM can be applied (see Refs. [19,56]). For example, $m_{q,i}$ is not limited to being a linear function of x_i , but can also perform a convolution or incorporate delays [56]. In all of our examples, a few components of $m_{q,i}$ are zero, indicating that these components, $y_{p,i}$, are not observed. The estimation error ϵ is weighted by a function $K(t, \beta)$ and added at each time step to the original dynamics $\mathbf{x}(t)$ to create an adjusted estimate $\tilde{\mathbf{x}}(t)$ and observations $\tilde{\mathbf{y}}(t)$ such that

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{f}_p(\tilde{\mathbf{x}}, \mathbf{u}(t)) + K(t, \beta)\epsilon_p(t), \quad (4a)$$

$$\tilde{y}_{p,i}(t) = m_{q,i}(\tilde{x}_{p,i}(t)), \quad (4b)$$

$$\epsilon_p(t) = \hat{\mathbf{y}}(t) - \tilde{\mathbf{y}}_p(t), \quad (4c)$$

where $\hat{\mathbf{y}}(t)$ is the observed data interpolated to whatever step is needed by the solver. The error gain factor $K(t, \beta)$ can take on different forms. In this work, we set $K(t, \beta) = K$ as a constant, the simplest form of what previous work has called the parametrized observer [56]. This observer form is closely related to synchronization-based stabilization of chaotic trajectories [57]; here it is used inside the UDE training loop rather than as a standalone state-estimation procedure. If there is strong reason to suspect that the quality of the observations varies over time (e.g., much higher noise at certain time intervals), then a time-varying $K(t, \beta)$ —for instance, an extension of Eq. (4) to resemble an extended Kalman filter, where $K(t, \beta)$ is the Kalman gain—might give better performance when fitting. Previous work has nonetheless shown that the parametrized observer outperforms the extended Kalman filter in practice, as tuning the hyperparameter over time is difficult

[56]. For notational simplicity, we will use K in the remaining sections, without loss of generality: An extended Kalman filter can be implemented if there is a data-driven reason to require it.

3. Combining PEM-UDE into a single method

We combine UDE and PEM into a single system by introducing the universal approximator NN_{θ} to learn the unknown dynamics while simultaneously using the prediction-error feedback to stabilize the training trajectory:

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{f}_p(\tilde{\mathbf{x}}, \mathbf{u}(t)) + K\epsilon_{\theta}(t) + NN_{\theta}(\tilde{\mathbf{x}}), \quad (5a)$$

$$\tilde{y}_{\theta,i}(t) = m_{q,i}(\tilde{x}_{\theta,i}(t)), \quad (5b)$$

$$\epsilon_{\theta}(t) = \hat{\mathbf{y}}(t) - \tilde{\mathbf{y}}_{\theta}(t). \quad (5c)$$

The hyperparameter K controls the loss landscape over which optimization is performed, which we will show is needed to fit UDEs to chaotic systems. In this work, we consider only systems where we have both an initial functional form and observational data. Estimation of unobserved latent states for which we have no initial idea of the functional form is a separate problem discussed elsewhere [45].

4. Symbolic recovery of terms from the PEM-trained UDE

After fitting the dynamics of the system with the combined PEM-UDE approach, we extract a symbolic form of the dynamics from the universal approximator NN_{θ} . The symbolic form usually generalizes to future contexts more robustly than a black-box neural network [3,18] and allows for direct analysis of the learned functional form. The terms learned by the UDEs can be recovered either by STLSQ, which learns features from a library via ridge regression [12], or by a genetic algorithm approach to symbolic regression [16]. Each has trade-offs, and the appropriate choice depends on the structure of the data and known dynamics. The genetic algorithm is the better choice when there are complex constraints on the functional form of the system [16], while STLSQ works better on systems that need many data points to achieve an accurate fit. We provide both approaches in the code accompanying this method and discuss our selection for each of the sections below. Since these approaches are not novel to this work, we leave further discussion to Sec. IV. In this pipeline, symbolic regression is not used to claim that a neural network is intrinsically interpretable; it is used as a sparsifying projection of the trained residual field, followed by forward simulation of the recovered equations.

B. Analytical properties of the PEM-UDE objective

Before the numerical demonstrations, it is useful to make explicit what the prediction-error term does to the optimization problem. Let

$$F_{\theta}(\mathbf{x}, t) = \mathbf{f}_p(\mathbf{x}, \mathbf{u}(t)) + NN_{\theta}(\mathbf{x}) \quad (6)$$

denote the modeled vector field, let $\mathbf{x}_{\theta}(t)$ solve the standard UDE, and let $\tilde{\mathbf{x}}_{\theta,K}(t)$ solve the PEM-UDE with the same initial condition. For a linear observation map $m(\mathbf{x}) = M\mathbf{x}$, the PEM term may be written as $G(\hat{\mathbf{y}} - M\tilde{\mathbf{x}}_{\theta,K})$, with $G = KM^T$ for scalar gain and direct componentwise observations. This is

the matrix form of Eq. (5); for a nonlinear observation map, the same local argument holds with M replaced by the observation Jacobian. We compare the ordinary and PEM-assisted trajectory losses in continuous time,

$$C_0(\theta) = \frac{1}{T} \int_0^T \|M\mathbf{x}_\theta(t) - \hat{\mathbf{y}}(t)\|_2^2 dt, \quad (7)$$

$$C_K(\theta) = \frac{1}{T} \int_0^T \|M\tilde{\mathbf{x}}_{\theta,K}(t) - \hat{\mathbf{y}}(t)\|_2^2 dt, \quad (8)$$

of which Eq. (2) is the sampled version.

Why the minimum is not moved. First, consider the idealized case that isolates the optimization geometry: The data are noise-free and generated by a member of the model class. Thus, there exists θ^* such that

$$\hat{\mathbf{y}}(t) = M\mathbf{x}^*(t), \quad \dot{\mathbf{x}}^* = F_{\theta^*}(\mathbf{x}^*, t), \quad (9)$$

with the same initial condition and unique solutions. At θ^* , the prediction error is identically zero,

$$\epsilon_{\theta^*}(t) = \hat{\mathbf{y}}(t) - M\tilde{\mathbf{x}}_{\theta^*,K}(t) = \mathbf{0}. \quad (10)$$

The PEM feedback therefore switches itself off exactly at the correct dynamics:

$$\frac{d\tilde{\mathbf{x}}_{\theta^*,K}}{dt} = F_{\theta^*}(\tilde{\mathbf{x}}_{\theta^*,K}, t). \quad (11)$$

By uniqueness, $\tilde{\mathbf{x}}_{\theta^*,K}(t) = \mathbf{x}^*(t)$, so $C_K(\theta^*) = 0$. Since the loss is a squared norm, $C_K(\theta) \geq 0$ for all θ , and θ^* is a global minimizer of the PEM-UDE objective for any fixed gain K .

The same argument also shows that PEM leaves the zero-loss set of the observed-state objective unchanged:

$$\{\theta : C_K(\theta) = 0\} = \{\theta : C_0(\theta) = 0\}. \quad (12)$$

If $C_K(\theta) = 0$, then the residual vanishes at all times, so the feedback term vanishes along the whole PEM trajectory and that trajectory also solves the unforced UDE. Conversely, any zero-loss standard UDE trajectory has zero residual and is unaffected by the PEM term. The feedback therefore reshapes the landscape away from the data-consistent manifold, but it does not move that manifold. Under structural identifiability and sufficient observability, this common zero set reduces to the true dynamics; under partial observability or model misspecification, different vector fields can still be observationally equivalent, so the statement should be read as preservation of the data-consistent minimum rather than a guarantee of unique global identifiability.

How PEM makes a chaotic objective trainable. Minimum preservation alone does not explain why the optimizer can find the minimum. The gain matters because it changes the stability of the error dynamics. Let $\xi(t) = \tilde{\mathbf{x}}_{\theta,K}(t) - \mathbf{x}^*(t)$, define the model-error field

$$\delta_\theta(\mathbf{x}, t) = F_\theta(\mathbf{x}, t) - F_{\theta^*}(\mathbf{x}, t), \quad (13)$$

and let $J^*(t) = D_{\mathbf{x}}F_{\theta^*}(\mathbf{x}^*(t), t)$. Linearizing the PEM-UDE around the true trajectory gives

$$\dot{\xi} = \underbrace{[J^*(t) - GM]}_{A_K(t)} \xi + \delta_\theta(\mathbf{x}^*(t), t) + \mathcal{O}(\|\xi\|^2). \quad (14)$$

For full observation with scalar gain, $GM = KI$, so the PEM term shifts each exponent of the linearized error dynamics

by $-K$. The unstable direction with exponent $\lambda_{\max}$ becomes contracting when

$$K > \lambda_{\max}. \quad (15)$$

This gives a simple physical interpretation of the loss-landscape smoothing: The observer damps the exponentially growing separation between nearby trajectories before it contaminates the loss and its gradients. In the Rössler example, the Lyapunov time is $\lambda_{\max}^{-1} \approx 10$ or $\lambda_{\max} \approx 0.1$, and the empirically useful gains $K \approx 0.18$ – 0.25 lie just above this threshold, consistent with Fig. S3 of the Supplemental Material [58]. For partial observation, the same damping occurs in the observed unstable subspace; unobserved unstable directions must be coupled to the observations for the feedback to stabilize them.

Let $\Phi_K(t, s)$ be the state-transition matrix generated by $A_K(t)$. If the observed error dynamics are uniformly contracting, so that $\|\Phi_K(t, s)\| \leq C \exp[-\gamma_K(t - s)]$ for $t \geq s$, variation of constants gives

$$\begin{aligned} \xi(t) &= \int_0^t \Phi_K(t, s) \delta_\theta(\mathbf{x}^*(s), s) ds + \mathcal{O}(\|\xi\|^2), \\ \|\xi(t)\| &\lesssim \frac{C}{\gamma_K} \sup_{0 \leq s \leq t} \|\delta_\theta(\mathbf{x}^*(s), s)\|, \end{aligned} \quad (16)$$

where $\gamma_K = K - \lambda_{\max}$ for full observation with scalar gain. The PEM loss near the minimum is therefore controlled by the autonomous vector-field error rather than by exponentially diverging trajectories:

$$C_K(\theta) \lesssim \frac{C^2 \|M\|^2}{(K - \lambda_{\max})^2} \sup_{0 \leq t \leq T} \|\delta_\theta(\mathbf{x}^*(t), t)\|^2. \quad (17)$$

Thus, K converts a chaotic trajectory-matching problem into a stable, observer-regularized comparison of vector fields, while Eq. (12) keeps the exact minimum fixed. The trade-off is visible in the same scaling: Increasing K suppresses sensitive dependence and smooths the landscape, but it also compresses differences between nearby vector fields by the factor $(K - \lambda_{\max})^{-2}$, producing the plateau seen at large gains.

Noisy observations and gain selection. With measurement noise, $\hat{\mathbf{y}}(t) = M\mathbf{x}^*(t) + \mathbf{e}(t)$, the linearized error dynamics become

$$\dot{\xi} = A_K(t)\xi + \delta_\theta(\mathbf{x}^*(t), t) + G\mathbf{e}(t). \quad (18)$$

The observer now damps chaotic divergence but also injects measurement noise into the simulated state. Let $\mathcal{L}_K$ denote convolution with the stable transition operator Φ_K . To leading order, the observed residual is

$$M\tilde{\mathbf{x}}_{\theta,K} - \hat{\mathbf{y}} \approx M\mathcal{L}_K\delta_\theta + (M\mathcal{L}_KG - I)\mathbf{e}, \quad (19)$$

and, for zero-mean noise independent of the model error,

$$\mathbb{E} C_K(\theta) \approx \frac{1}{T} \int_0^T \|M\mathcal{L}_K\delta_\theta\|_2^2 dt + \mathbb{E} \|(M\mathcal{L}_KG - I)\mathbf{e}\|_2^2. \quad (20)$$

The first term is the useful contrast between different autonomous vector fields; the second term is a K -dependent noise floor. Small gains leave the chaotic instability active.

Very large gains make the observer track the measured time series so strongly that model errors are hidden by feedback and the informative contrast in Eq. (20) becomes shallow. This motivates a useful interval,

$$\lambda_{\max} < K \lesssim K_{\text{noise}}, \quad (21)$$

where the lower bound stabilizes the trajectory and the upper scale is set by the point at which the model-error term becomes comparable to the noise floor. The precise upper scale depends on the observation operator and the spectrum of the measurement noise, so it is not universal. In practice, when a Lyapunov scale can be estimated from data, we choose K just above that scale and check the resulting loss landscape; for the neural mass model, we instead use a continuation schedule and anneal K from 0.3 to 0.1 before symbolic extraction, reducing the residual influence of the observer on the learned autonomous dynamics.

What the neural network term learns. The network is not intended to learn the whole vector field. In the missing-physics formulation, it learns only

$$g(\mathbf{x}, t) = F_{\theta^*}(\mathbf{x}, t) - f_p(\mathbf{x}, \mathbf{u}(t)), \quad (22)$$

the part not already represented by the known physics. This smaller target is central to why a small network can work and why a large network is not automatically better. For noisy data, the fitted residual field has the usual bias-variance structure,

$$\begin{aligned} \mathbb{E}[\|NN_{\hat{\theta}}(\mathbf{x}) - g(\mathbf{x})\|_2^2] &= \underbrace{\|\overline{NN}(\mathbf{x}) - g(\mathbf{x})\|_2^2}_{\text{bias}^2} \\ &+ \underbrace{\mathbb{E}[\|NN_{\hat{\theta}}(\mathbf{x}) - \overline{NN}(\mathbf{x})\|_2^2]}_{\text{variance}}, \end{aligned} \quad (23)$$

where $\overline{NN}$ is the noise-averaged learned network. Increasing the network size can reduce the bias, but it increases the variance term, which is the part that follows a particular noise realization.

This capacity dependence is the standard bias-variance trade-off [59]. A local linearization illustrates it explicitly. If $\Theta \in \mathbb{R}^{N \times P}$ is the resulting feature matrix, P is the number of active degrees of freedom, and σ_{eff}^2 is the effective noise variance in the regression target, then with ridge/weight-decay regularization

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \text{Var}[NN_{\hat{\theta}}(\mathbf{x}_i)] &\approx \frac{\sigma_{\text{eff}}^2}{N} \text{tr}[\Theta(\Theta^\top \Theta + \lambda I)^{-1} \Theta^\top] = \frac{d_{\text{eff}} \sigma_{\text{eff}}^2}{N}, \\ d_{\text{eff}} &\leq P. \end{aligned} \quad (24)$$

The noise-fitting term therefore grows with the effective number of parameters and falls with the number of informative samples. The optimal model size is finite: It is the smallest network for which the bias has become small compared with the variance cost. Because f_p already contains the dominant physical structure, the residual g is much lower complexity than the full dynamics, so this optimal size can be small. This is why we use compact $N_1 \times 10 \times 10 \times N_2$ networks rather than large black-box neural ODEs.

These arguments mitigate, but do not eliminate, the risk that the network absorbs dynamics attributed to the known

TABLE I. Direct symbolic regression with STLSQ applied to noisy Petrzela-Polak circuit data fails to recover the correct dynamics under the same data and noise conditions where the PEM-UDE approach succeeds.

λ	Fit expression
0.1	$\phi_1 + \phi_2 x + \phi_3 y + \phi_4 z + \phi_5 xy + \phi_6 y^2 + \phi_7 xz + \phi_8 z^2$
0.2	$\phi_1 + \phi_2 x + \phi_3 y + \phi_4 z + \phi_5 xy + \phi_6 xz + \phi_7 yz + \phi_8 z^2$
0.3	$\phi_1 + \phi_2 y + \phi_3 z + \phi_4 z^2$
0.4	$\phi_1 + \phi_2 y + \phi_3 z + \phi_4 y^2 + \phi_5 yz$
0.5	$\phi_1 + \phi_2 y + \phi_3 z$

physics. In our experiments, f_p and its assumed parameters are held fixed, so the network target is the residual needed to match the observed trajectory. The data nevertheless constrain the total field mainly on the sampled states: A flexible network can compensate for misspecified known terms or adopt different off-trajectory extensions. Keeping P small and using AdamW weight decay restricts those alternatives through an objective of the form

$$C_{K,\lambda}(\theta) = C_K(\theta) + \lambda R(\theta). \quad (25)$$

Finally, the symbolic regression step imposes an additional sparsity constraint and the learned symbolic equations are validated after the PEM term is reduced or removed. These checks reduce, but cannot rule out, residual nonidentifiability away from the sampled attractor. They also help explain why STLSQ succeeds on PEM-UDE outputs but fails on the raw noisy data in Table I: Integration through the trained UDE produces a smooth generative trajectory, whereas numerical differentiation of noisy data amplifies high-frequency noise by a factor proportional to ω^2 in power.

C. PEM-UDE outperforms standard UDE in fitting chaotic systems

We begin by showing how the PEM-UDE approach fits the Rössler system (Fig. 2). This system was chosen because it is a simple set of ODEs with only a single nonlinear term, yet it produces chaotic dynamics for the right parameter set [24]. In this example, we assume that we know the functional form of x and z but have no information about the functional form of y , and so must learn the dynamics from the time-series observations. With only this challenge, a traditional UDE fails to fit well [Fig. 2(a)]: The UDE-learned dynamics typically converge to the mean of the time series, a local minimum in the objective function. The PEM-UDE approach, in contrast, converges successfully after training, with a clear difference in performance even after the first 10% of the total training time, at which point the standard UDE has already neared its local minimum.

The reason for this performance increase lies in the way that the PEM approach alters the landscape over which the UDE is trained [Fig. 2(b)]. The unmodified landscape is extremely rough and contains several local minima, with only a very sharp decrease in loss at the true global minimum ($a = 0.2$ in this case). When PEM is added to the dynamics of the system during training of the UDE, the landscape becomes much more amenable to standard optimization techniques.

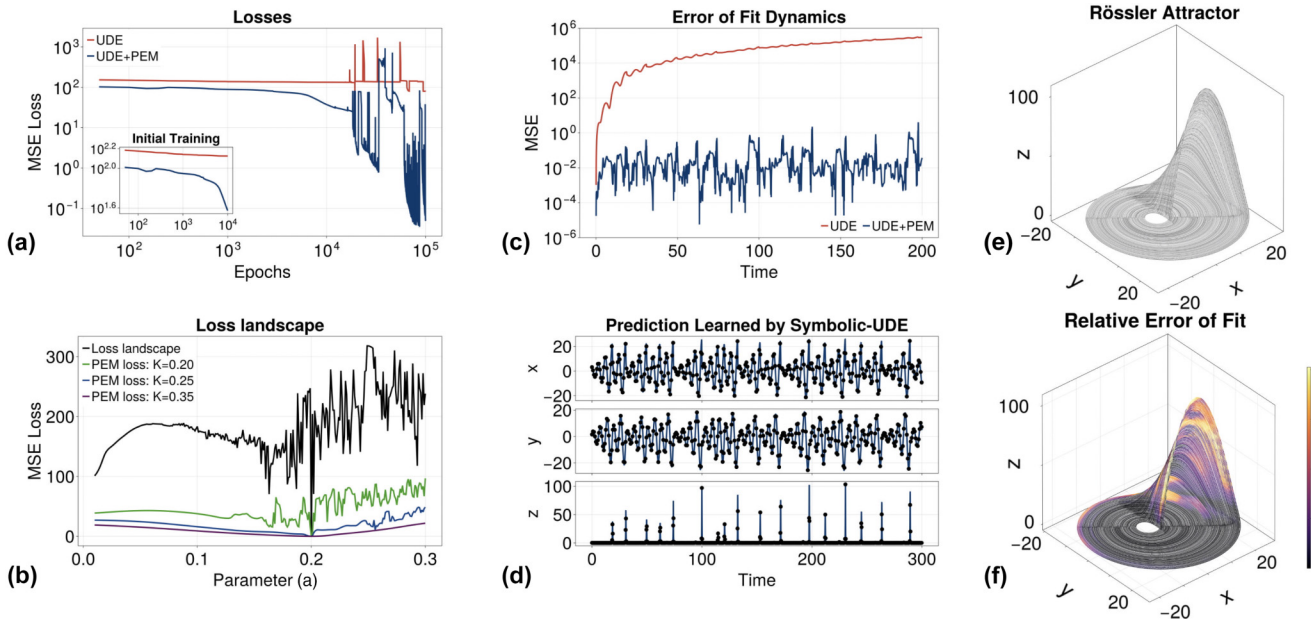


FIG. 2. The combined PEM-UDE approach can learn the dynamics of chaotic systems that traditional UDEs cannot. (a) Comparison of training error for the UDE system (fails to capture dynamics accurately) and the PEM-trained UDE (succeeds). The difference in accuracy is already large 10% into the full training session (inset), at which point the unassisted UDE is no longer progressing. (b) The application of PEM smooths an otherwise difficult parameter loss landscape, with the hyperparameter K tuning the steepness. (c) Mean-squared error of the fit dynamics, UDE vs PEM-trained UDE. (d) Using STLSQ to replace the PEM-trained UDE produces an expression that captures the original dynamics (first 200 time points) and continues to describe them accurately well into the future (last 100 time points). (e) The surface of the Rössler attractor shows chaotic filling of phase space, with a basin near the x - y plane and occasional bursts along the z axis. (f) Relative error of the PEM-trained UDE at each point of the attractor surface.

The choice of K also tunes the steepness of the landscape directly. This should be done in conjunction with considering how the PEM will smooth out the landscapes of other parameters in states where it is not applied. In the noise-free realizable case, PEM preserves the data-consistent zero-loss set, but it can flatten the landscape in directions associated with unobserved states or misspecified parameters. As higher levels of filtering are introduced with larger K , finding the true functional form of the entire system depends more on the accuracy of the estimates of the unfit parameters (see Fig. S2 of the Supplemental Material [58]). With the PEM-UDE combination approach, then, the best choice of K is one that produces a smooth enough landscape for optimization but is still steep enough that the global minimum is clearly distinguishable from the surrounding area, so that noisier estimates of unmeasured parameters are acceptable.

The PEM-UDE approach works because the prediction-error correction removes the sensitive dependence on parameters from the system being fit, neutralizing the defining feature of chaotic systems. As shown in Fig. S3(a) of the Supplemental Material [58], when K is small there is an $\mathcal{O}(1)$ divergence with respect to any deviation in the initial parameter choice [illustrated as a deviation of 10^{-9} in Fig. S3(a) of the Supplemental Material [58]], but when K is sufficiently large ($K \approx 0.18$ in this example) this sensitive dependence is eliminated and the error between the two trajectories shrinks. For sufficiently large K , the residual error depends on the size of the deviation in the parameter, and K can be tuned based on the estimate of this deviation [Fig. S3(b) of the Supplemental Material [58]]. For noise-free observations generated within

the model class, the feedback vanishes on the data-consistent trajectory and preserves the zero-loss set [Eq. (12)]. With noise, partial observability, or misspecification, PEM instead trades stability against a shallower and potentially biased objective [Eq. (20)]. The L_2 -norm trajectory fitting approach, which is unstable on the original chaotic system, is thereby stabilized over the range of gains examined here.

The dynamics learned by the PEM-UDE fit the system well, while the initial estimate of the UDE alone does not offer an accurate prediction even within the Lyapunov time of the system [Fig. 2(c)]. For the parameters selected here, the Lyapunov time of the system is $\lambda^{-1} \approx 10$, with most nearby areas of parameter space having similar characteristic times ($\lambda^{-1} < 25$; see Sec. IV for further details). The mean error of the UDE alone continues to grow throughout the simulation time, whereas the mean error at each time step of the PEM-UDE fit dynamics is 2–3 orders of magnitude smaller than the dynamics themselves.

To generalize these dynamics beyond the original observations, we extracted a symbolic form of the equations using an STLSQ fit to the outputs of the UDE taken at each time step of the dynamics of the PEM-UDE system. The functional form of the missing state is directly recovered by STLSQ ($\dot{y} = p_1 x + p_2 y$), with the initial fits of parameters close to the true values ($p_1 = 1.01$, $p_2 = 0.17$, while the true values are $p_1 = 1$, $p_2 = 0.2$). An additional optimization step that tunes the parameters of this learned symbolic equation after STLSQ converges to the true values (accurate to three decimal places), which then generalize the dynamics forward in time beyond the original simulations [Fig. 2(d)]. The pipeline thus

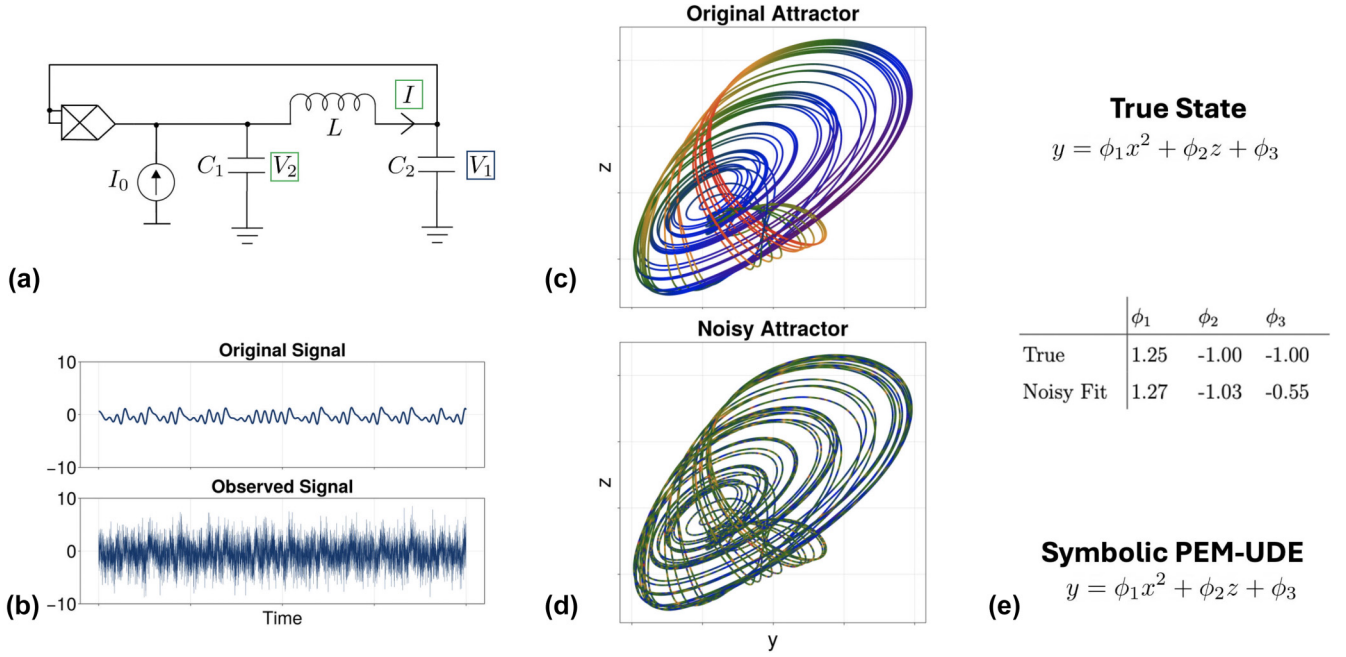


FIG. 3. The PEM-UDE approach is capable of recovering the correct form of a chaotic circuit even in the presence of significant observational noise. (a) The analog Petrzela-Polak circuit that produces the dynamics examined in this section. The observed states are the dimensionless forms of V_1 , V_2 , and I , where V_2 and I (green boxes) are observed with only a small amount of noise, and V_1 (blue box) is observed with large observational noise. (b) In this example system, we assume a faulty observation of one state (V_1), where the true dynamics (top) are masked by observational noise of $5\times$ the true signal's magnitude (bottom). (c) Original attractor surface, with the true form of the observed dimension x in color and the magnitude of error from observational noise shown. (d) Attractor surface fit by the PEM-UDE method with noise in the third dimension (the scattered colors indicate that the noise has significantly corrupted the dynamics of the state x). (e) The true form of the state to be recovered is shown at the top, with the form recovered by STLSQ applied to the PEM-UDE dynamics shown at the bottom. The functional form, although not the exact parameters, is recovered from the PEM-UDE fit (table compares parameters fit by STLSQ to original parameters).

provides an analytical form of the dynamics that generalizes forward in time and can also be examined for insights into the physical system, recovering the original attractor of the system with an error typically under 4% [Figs. 2(e) and 2(f)]. This benchmark therefore evaluates hidden-state recovery, parameter error, out-of-window simulation, and attractor geometry, rather than only short-horizon trajectory matching.

D. PEM-UDE can learn chaotic dynamics under severe observational noise

We next present a second example of the PEM-UDE method being fit to a chaotic system, with two important differences from the Rössler system. First, this example is a physically realized electrical circuit, with the circuit diagram shown in Fig. 3(a) and the parameters for the physical circuit (as described in Ref. [60]) given in Table II. The move from the idealized chaotic dynamics of the previous system to a real-world circuit also has practical relevance, since chaos in electrical circuits has applications of its own [61]. Second, we add a noisy observation to one of the circuit components [V_1 , denoted by the blue box in Fig. 3(a)]. The true equations for this system are given in Eq. (32), with the dimensionless form used in this example given by Eq. (33). In this section, we assume there is an unknown faulty observation of the first state x , such that the noise around the observation is $5\times$ the magnitude of the true signal [Fig. 3(b)]. This creates a

difficulty for the PEM-UDE fit: The true dynamics [Fig. 3(c)] are now masked by enough noise to distort the shape of the attractor being fit [Fig. 3(d)].

We assume that this noise is unknown at the time of fitting and proceed with fitting the PEM-UDE to the corrupted data using the same approach as above [fitting the functional form shown in Eq. (34)]. Even with the substantial observational noise in one of the states, STLSQ applied to the PEM-UDE fit dynamics recovers the exact functional form of the dynamics [Fig. 3(e)]. The approach does not provide the original parameters, although the form is correct. The problem has effectively been shifted from one of unknown dynamics to one of simple parameter optimization, to which other approaches (e.g., simulation-based inference [62]) can be applied. This indicates that even with very noisy observational data, the PEM-UDE approach can recover useful information about the true dynamics of the system.

This example also illustrates why fitting the symbolic form of a trained UDE outperforms an STLSQ fit of the original dataset. As a comparison to the dynamics shown in Fig. 3(e), we fit the same noisy dataset directly using the SINDy approach [12,63]: We compute a dataset of smoothed derivative estimates and perform the STLSQ symbolic regression method directly on the derivative estimates. As shown in Table I, this method cannot recover the true symbolic form across a range of the alternating direction method of multipliers (ADMM) hyperparameter λ , with more sparse

representations incorrectly dropping the noisy state x from the expression entirely. Recovering the symbolic form of the trained UDE is more accurate because the UDE provides a noise-free (or greatly noise-reduced) version of the dynamics that can be sampled with arbitrary density, sidestepping the sensitivity that symbolic regression methods show to noisy data. This comparison is not meant to exhaust all possible smoothing or weak-form variants of SINDy; rather, it isolates a failure mode that is common in derivative-based symbolic discovery, where differentiating noisy observations amplifies high-frequency error while integrating the learned UDE produces a smooth generative trajectory for subsequent sparse regression.

E. Learning novel expressions for sparse networks that violate analytical derivation assumptions

Having shown that this approach can learn the dynamics of chaotic physical systems, we next turn to deriving expressions for the mean-field activity of neuron populations. In this section, the learned equations are reduced-order closure equations for simulated microscopic networks, not equations fit directly to the experimental recordings considered later. The data needed to study the human brain span multiple scales: Individual neuronal activity gives rise to synchronized actions that define interacting functional circuits [Figs. 4(a)–4(c)]. Traditional neural mass models represent the average activity of neuron populations by modeling particular phenomena (e.g., excitatory/inhibitory balance in the Wilson-Cowan model [41], ion-gradient dynamics in excitatory populations in the Larter-Breakspear model [43,64]), but lack a direct tie to many of the biophysical parameters (e.g., neurotransmitters) needed to understand interneuron dynamics. NGNMMs address this by providing an analytical derivation of the mean-field dynamics that retains the biophysical details of individual neurons and also captures the emergent properties at the population level [44,45]. The derivation and its key assumptions are discussed further in Sec. IV.

The analytical form of mean-field dynamics offered by NGNMMs has several advantages over traditional neural mass models. First, there is a direct connection between the single-neuron parameters and the dynamics of the neural population, a feature only available in certain populations of traditional neural masses tuned to be sensitive to these parameters [43,64]. Second, NGNMMs capture emergent properties of neural populations that traditional neural masses do not. For example, β rebound, where suppressed activity in the β band increases in power after being suppressed (a phenomenon often observed in motor cortex studies), is observed as a feature of coupled NGNMMs that are built only to oscillate in the β band, with no functional constraints that force this activity to occur [46].

There are also limitations to the NGNMM approach, mostly involving the assumptions needed for Ott-Antonsen ansatz [48], a key component of the NGNMM derivation. Most consequential from a biophysical standpoint is the assumption of all-to-all connectivity of the neuron population. This is approximately correct in deep regions (e.g., hippocampus, entorhinal cortex) that have significant within-region collaterals [45]. Immediate neighbors of these regions tend

to have fewer collaterals, however, as in the thalamostriatal connection patterns [65]. The assumption of full connectivity becomes obviously false in the cortex [52], which typically has very sparse connectivity [on the order of 1%–5% of all possible connections; Fig. 4(d)]. Accounting for sparsity in NGNMMs is therefore a problem that must be solved for them to be useful in new contexts.

There is disagreement on the correct analytical approach to adjust the derivation of NGNMMs for sparsity [50,51,66,67], so we sidestep the issue of an analytical derivation and learn the form of the sparse networks from simulated microscopic-network data with the PEM-UDE method. Following the analytical approach for Izhikevich neurons [45], we begin with a large population of neurons described by Eq. (35) with synaptic dynamics given by Eq. (36). In the all-to-all case, the analytical NGNMM describing the activity of this population is given by Eq. (37). As we begin to prune connections in the graph to introduce sparsity, this form no longer accurately represents the population activity. We therefore learn correction terms for the mean firing rate r and voltage v given a reduced probability of connectivity p_c , where $p_c \in [0.05, 1]$. Although, there are additional states for the recovery current w and the synaptic dynamics s , we only learn from the firing rate and voltage states and train only on their data, since these are the data typically available from experiments. Restricting training to these two macroscopic states does not make the later experimental recordings the training set; it instead mirrors the observability constraints that population recordings impose.

The PEM-UDE approach captures the mean-field dynamics of the network under different degrees of sparsity with good accuracy [Fig. 4(e)]. This accuracy does not include the initial conditions; this is discussed further in the limitations portion of Sec. III. The method does capture a shift in the dominant frequency of oscillations present in the network as a function of sparsity [Fig. 4(f)]. Densely connected regions typically have a primary frequency peak in the θ band, while the most sparse networks exhibit a primary peak in the α band. This emerges as a natural feature of the learned dynamics and is qualitatively consistent with experimental data. Hippocampal rhythms are typically in θ frequencies and emerge from densely connected areas of the brain [68], while α frequencies typically emerge from sparsely connected cortical regions before propagating through the brain [69]. The PEM-UDE learned dynamics therefore extend the strength of NGNMMs in capturing emergent properties (e.g., synchrony) at the mean-field scale by accurately capturing the ergodic properties of the system. The learned dynamics also generalize beyond the training data: When trained only on $p_c = 0.3$ – 0.9 , the PEM-UDE still recovers dynamics down to 5% connectivity (see Fig. S4 of the Supplemental Material [58]).

As with the previous two systems, we extend the PEM-UDE learned dynamics described in Eq. (39) by fitting a symbolic form of the dynamics to the UDE network with STLSQ, giving an analytical form of the solution [Eq. (40), with fit parameters in Table IV]. Izhikevich and related quadratic-integrate-and-fire NGNMMs can undergo period-doubling and chaotic collective regimes under other parameter choices and coupling structures [25,70,71]. The parameter

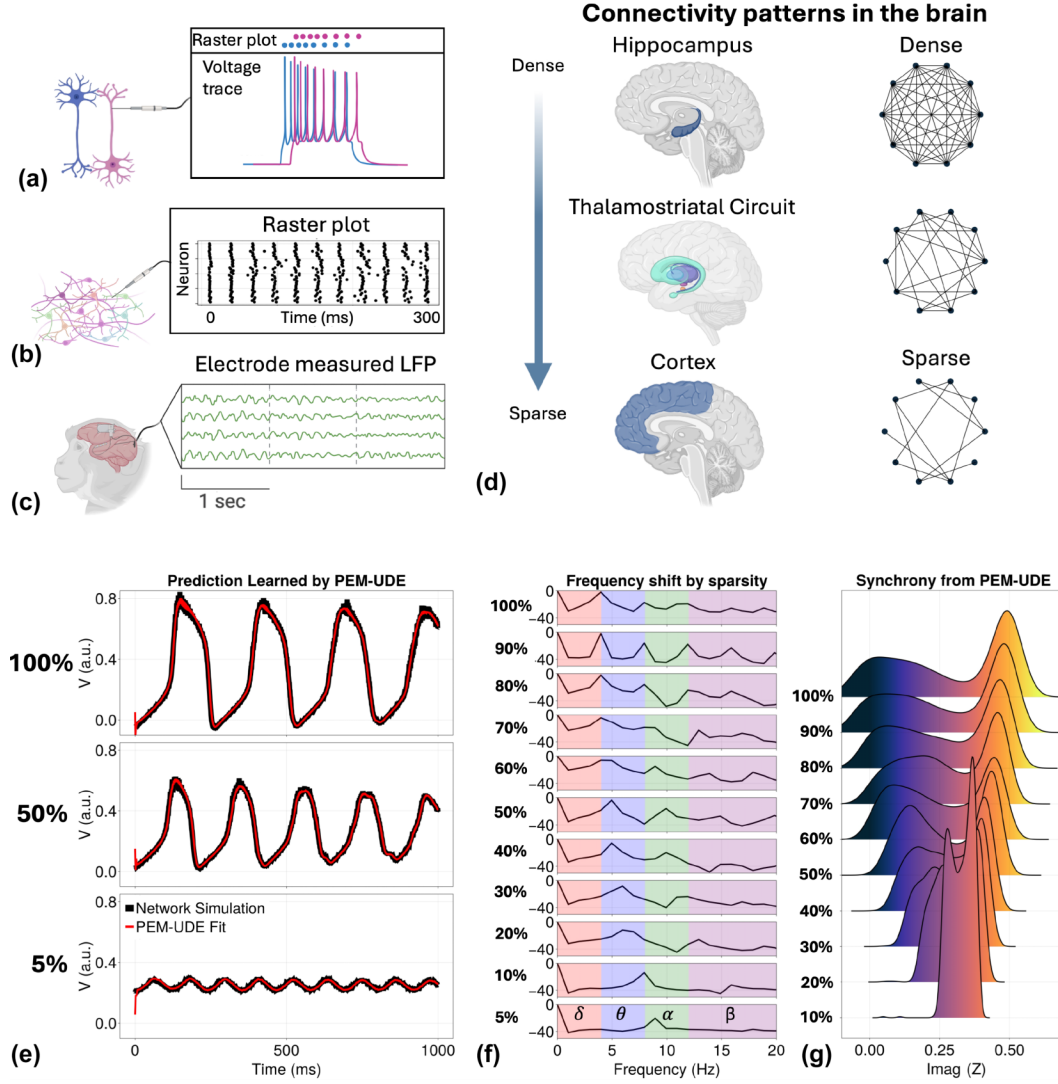


FIG. 4. Learning neural mass models that account for regional heterogeneity in neural connection sparsity helps explain the origin of different frequency bands within the brain. Single neurons produce spike trains [panel (a)], which are encoded as a raster plot of individual dots for each neuron/spike. In a population, this view makes group activity easy to read, and synchronized regimes show clearly visible bursts [panel (b)]. In behavioral experiments, electrode arrays can be placed to record population-level activity from multiple brain regions [panel (c)]. The degree of local connectivity within brain regions varies across the brain [panel (d)], with deeper regions exhibiting greater connectivity and cortical regions having sparser within-region connections. To account for this sparsity, we train a next-generation neural mass model to learn terms that capture the emergent population dynamics as connections become more sparse [panel (e)]. The PEM-UDE learned dynamics exhibit qualitatively similar frequency trends to those seen in real brain regions, with a shift from theta to alpha as the dominant frequency as network connections become increasingly sparse [panel (f)]. Higher sparsity also limits the synchronization that is possible within the network [panel (g)]. Figure created with BioRender.com.

set used here, however, lies in a stable population-oscillation regime. The neural example therefore tests the same training framework in a partially observed oscillatory system and is distinct from the two chaotic benchmarks. Since NGNMMs are derived based on the assumption of a population of coupled oscillators, there is an exact expression for the degree of synchrony within the population given by the Kuramoto order parameter Z :

$$Z = \frac{1 - W^*}{1 + W^*}, \quad (26a)$$

$$W = \pi r + iv. \quad (26b)$$

For the PEM-UDE learned network dynamics, we observe an emergent prediction that more densely connected regions tend to have higher degrees of synchrony as a baseline state, while more sparsely connected regions tend to exhibit lower degrees of synchrony when not acted upon by an external drive [Fig. 4(g)]. This is an unexpected prediction from the learned reduced-order equations for networks of different sparsity. To test it, we examined three datasets of intracranial electrode recordings from regions of different degrees of sparsity (Fig. 5). Because these datasets differ in species, anatomical region, electrode configuration, task context, and behavioral state, this comparison cannot isolate local

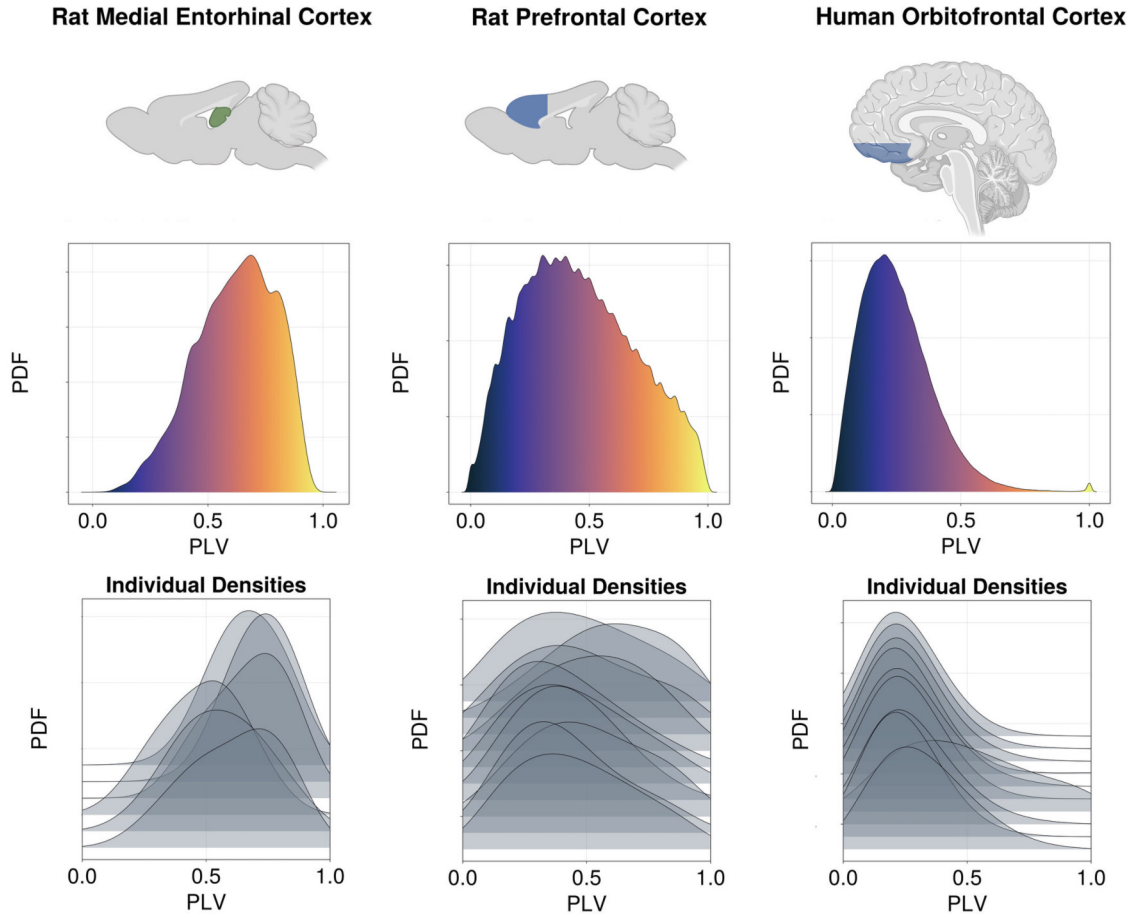


FIG. 5. Intracranial EEG shows synchrony variation across regions consistent with the symbolic PEM-UDE-derived neural mass model. As shown in Fig. 4(g), the typical synchrony in regions with greater connectivity is higher than in regions with lower connectivity. A similar ordering appears in biological brains: We examined three datasets. In the region with the greatest connectivity (rat medial entorhinal cortex; left column), there is a greater degree of synchrony (measured by phase-locking value; PLV) over the duration of the experiment than in the prefrontal cortex (middle column). The difference is larger still compared to the human orbitofrontal cortex at rest (right column), where overall synchronization tends to be very low during spontaneous activity. As all cumulative distributions (upper row) are based on synchrony values from many individuals, synchrony distributions from each subject are also shown (bottom row) to illustrate the variability across individuals. Figure created with BioRender.com.

connection probability as the causal variable. It is instead an external consistency check: regions whose known anatomy implies different typical local connectivity show the same ordering of resting synchrony predicted by the simulated reduced-order model. These datasets included recordings from the rat medial entorhinal cortex (MEC; $N = 6$), a deep region near the hippocampus with high intraregional connectivity [72,73]; the rat prefrontal cortex ($N = 10$), which has significantly sparser connectivity [74,75]; and human orbitofrontal cortex ($N = 10$), which exhibits a similarly high degree of sparsity [76,77]. Consistent with the learned reduced-order equations, the deeper, more densely connected structures exhibit higher synchrony than either of the cortical datasets, and the pattern holds across individuals, although with significant variability between subjects. These results support the plausibility of the model's qualitative prediction outside the simulation set, while leaving causal validation of the learned terms to future experiments with directly measured connectivity.

III. DISCUSSION

Here, we introduce a method to fit universal differential equations in chaotic systems and use it to learn reduced-order equations for simulated neural-population activity. The approach uses the prediction-error method to train the UDE system, and a symbolic form of the learned dynamics is then extracted via symbolic regression [12]. We show that the method can learn the dynamics of a classical chaotic system, the Rössler system [24]. A standard UDE alone cannot learn this system effectively, but the combined PEM-UDE approach captures the dynamics accurately, suggesting that the method has identified the true functional form of the dynamics rather than projecting within conventional statistical boundaries. Prior work in UDEs has focused on stiff systems [8], fitting UDEs in the presence of stochastic noise [78], and extracting explainable functions from UDE networks [18]. Separately, prior work in parameter optimization using the PEM has shown its utility in fitting parameters of

dynamical systems [19,56], and that it can even outperform neural network fits of dynamics [79,80]. Our methodological contribution is the use of prediction-error feedback inside UDE training for chaotic trajectories, followed by recovery and autonomous validation of an explicit residual field; observers, UDEs, and symbolic regression themselves are established methods. The approach succeeds where standard UDE methods fail because it smooths the parameter landscape that the UDE must optimize. Under the noise-free realizability and uniqueness assumptions stated in Sec. II B, this deformation preserves the data-consistent zero-loss set; under noise, misspecification, or incomplete observability, the analysis instead identifies the conditions and trade-offs that limit that statement. We also extend the approach to a second chaotic system from an electrical circuit [60,81] and show that even in the presence of significant observational noise, the PEM-UDE method can recover the true functional form of an unknown state. More accurate estimation of the parameters of these recovered dynamics could be obtained by taking the functional form learned from the PEM-UDE dynamics and applying a more advanced optimization technique that fully exploits the ability to simulate the system in different regimes (e.g., simulation-based inference [62]).

The analytical results of Sec. II B explain these empirical observations while delimiting their scope. Prediction-error feedback leaves the zero-loss set unchanged for noise-free realizable data because the feedback vanishes on the data-consistent manifold [Eq. (12)]. For full observation with scalar gain, it shifts the linearized error exponents by $-K$ [Eq. (14)], so $K > \lambda_{\max}$ is a local sufficient condition for contraction. Partial observation additionally requires coupling of unobserved unstable directions to measured states. With noise or misspecification, the same analysis shows why the gain must remain moderate: Strong feedback can mask autonomous model error. The bias-variance argument of Eqs. (23) and (24) motivates the compact networks used here, but does not by itself guarantee identifiability away from the sampled attractor.

The cost of the observer term is small relative to the UDE solve-and-differentiate loop: It adds interpolation of the measured state and a vector addition to the right-hand side at solver time steps, while the dominant computation remains ODE integration, sensitivity calculation, and network optimization. We therefore view PEM-UDE as a stabilization of standard UDE training rather than a competing state-estimation method. Other approaches developed for chaotic identification, including multiple shooting, weak-form objectives, shadowing-based losses, and filtering-assisted symbolic regression, are natural complementary baselines and can often be combined with the same observer-regularized UDE. Recent SINDy-type work has likewise improved robustness to noisy data by combining linear multistep approximations, ensemble adaptive libraries, and inner-product sparse regression [82]. Our comparisons establish performance relative to an otherwise identical UDE and to direct STLSQ under the tested conditions; they do not establish state-of-the-art performance across all chaotic-identification methods. A systematic cross-method benchmark over higher-dimensional, stiff, and multiscale systems remains future work.

Having shown that the PEM-UDE approach can learn the dynamics of chaotic systems, we then considered how

to learn reduced-order governing equations in systems of neurons under biologically plausible constraints. These constraints are of two kinds: experimental constraints on the data available, which require using only data sources that can be feasibly acquired *in vivo* (firing rate, membrane voltage); and neurobiological constraints, where the sparsity of interneuron connections must be accurately modeled. Experimental constraints are worth keeping in mind because they make the method more immediately applicable, but methodological advances can often relax these over time. Biological constraints, and connection density, in particular, are more important to model accurately, as they shape the kinds of information processing possible within the brain [53,54]. We use a missing-physics approach, where the initial form of the population activity is given by an NGNMM [44,45]. NGNMMs capture emergent properties of neuron populations and retain the microscale parameters that are typically manipulated *in vitro* [46,83], but they rely on the assumption of all-to-all connectivity, which is nonphysiological for most brain regions [44]. There have been varied attempts to sidestep this limitation and include corrections for sparsity by reweighting the degree of synaptic conductance or the distribution of synaptic currents, with varying degrees of success [50,51,66,67,84]. Each of these approaches comes with new assumptions and its own limitations. Here, we instead learn a reduced-order closure for an NGNMM of a simulated population of Izhikevich neurons directly from the observations of the network itself. The closure equations learned this way predict emergent properties that are observed directly in biology, such as the shift from θ to α as the dominant frequency band as networks become more sparse, a feature seen in the brain moving from densely connected regions (e.g., hippocampus) to sparsely connected regions (e.g., prefrontal cortex) [68,69]. The equations also predict that more densely connected regions exhibit greater intraregional synchrony at rest than sparsely connected networks. To test this hypothesis, we computed the intraregional synchrony from three datasets of intracranial electrode recordings (rat MEC, rat prefrontal cortex, and human orbitofrontal cortex) [72,74,76]. The ordering seen in these datasets is consistent with the simulated model's prediction, but because connectivity was not measured directly it should be interpreted as a biological consistency check rather than causal validation.

The neural-population result should be interpreted at the level of a reduced-order closure. The microscopic simulations use Izhikevich neurons and parameter values inherited from the analytical NGNMM construction of Chen and Campbell [45], chosen to represent hippocampal-like dynamics under the all-to-all limit. The learned terms then correct that mean-field system as the connectivity probability is varied. This is different from common statistical fits of neural recordings, such as generalized linear or point-process models of spike trains [85,86]: Those models are often simpler and more identifiable for encoding or decoding experiments, while PEM-UDE aims to preserve a biophysical population-level state space that can retain microscopic parameters and emergent synchrony. The price of that added structure is that the learned sparsity corrections should not be read as unique biological mechanisms until they are tested against recordings with matched structural connectivity or prospectively varied

microscopic parameters. A decisive biological test would pair recordings with measured structural connectivity or controlled perturbations within a common preparation.

The method also has limitations that we will address in future work. Very noisy systems can be difficult to learn, especially since chaotic systems can be hard to average over trials. Care must also be taken: Nonphysical solutions can shadow physical ones [26]. The inclusion of information during training via the PEM, together with the extraction of a symbolic form of the dynamics, helps to mitigate these difficulties. In models that include known physics, PEM can lead to close-but-inaccurate minima if the parameters are not exactly known. This limits the method’s applicability to scenarios where the parameters included in the known physical model are sufficiently accurate. In principle, this can be overcome by fitting to a range of simulations across these parameters, but the training time will increase substantially. The method is also best suited to situations with a known structure of data (i.e., some driving inputs are known). Neural experiments often provide this level of accuracy, but there is also substantial interest in resting-state data [87]. Adapting the approach to fit stationary time series without known perturbations is therefore another extension we see as important for neuroscience applications.

For the neural example specifically, the frequency shift with sparsity may depend on the chosen neuron parameters, synaptic scaling, graph ensemble, finite-size effects, delays, heterogeneity, and external drive. We therefore interpret the theta-to-alpha transition as a prediction of the simulated Izhikevich network family studied here, not as a universal law for all neural populations. The comparison with rat MEC, rat prefrontal cortex, and human orbitofrontal cortex is likewise limited by species, region, electrode, behavioral, and clinical differences; future validation would ideally use recordings paired with measured structural connectivity within a common preparation.

Although the work presented here shows the utility of this approach in neuroscience, there are several areas for further development. Future work can incorporate loss objectives that include optimization of initial conditions, which can help generalization to real-world systems [79,88]. Future work could also focus on learning “black-box” representations of the dynamics, where symbolic regression methods are not applied to learn the dynamics. This can be useful when complexity penalties (i.e., enforced sparsity within the learned expression) bias away from the correct dynamics. For example, when many small inputs lead to complex interactions based on their temporal pattern, the high-dimensional representation present in the neural network may be more desirable. The temporal sequences of visual cortex projections to the prefrontal cortex are one example of this kind of structured high-dimensional input space [52]. In such a case, rather than fitting the dynamics symbolically, a second optimization run of the UDE without PEM after the initial combined run could lead to more generalizable results (since this forces the contribution of PEM to 0 rather than letting it minimize naturally over the optimization problem). Including a method for directly estimating stochastic processes within the dynamics, as has been done in some standard UDE approaches [78], would add another layer of generalization to

the approach. This is especially relevant for finite or partially observed neural populations: The quenched input distribution in the NGNMM appears through $\bar{\eta}$ and Δ , but unobserved neurons can also generate stochastic finite-size fluctuations that are not represented by the deterministic closure used here. A stochastic PEM-UDE would let those fluctuations be learned as process noise rather than forcing the deterministic residual terms to absorb them. Extending the PEM-UDE approach to systems of partial differential equations would also allow for direct fitting of neural field models under different degrees of connection density, which is of increasing interest as neuroscience tries to understand the role of traveling waves in the brain [21,89] and how the relationship between densely and sparsely connected regions shapes the flow of cognitive processes [53].

IV. METHODS

A. Scientific machine learning in the Julia ecosystem

The Julia ecosystem has a solid infrastructure for scientific machine learning [18] that we use to build the PEM-UDE method described here. Since setting up and training UDEs in Julia has been discussed elsewhere (Refs. [8,18] are good examples), we refer interested readers to the accompanying code for implementation details and give only a few brief technical notes here on the choices of solvers, network structure, fitting procedure, and symbolic regression.

In this work, we default to nonstiff solvers (typically Vern7), mostly for speed. As long as the UDE network is large enough to allow for a decent approximation of the unknown dynamics and the system itself is not very stiff, a nonstiff adaptive solver should give sufficiently accurate fits. If the system is known or strongly suspected to be stiff, then switching to a stiff solver (e.g., Rodas4) is recommended and is easy in the provided code, since the Julia ecosystem has a wide range of solvers available. This switch may become necessary either because chaotic systems are often stiff in certain regions, or because even when the system itself is not stiff (as the neural systems discussed here can be in the correct parameter space), the inclusion of a universal approximator will often lead to a stiff problem, at least until the initial parameter weights are adjusted. We recommend increasing the size of the universal approximator network before defaulting to a stiff solver, as this can have a large impact on performance.

We choose the radial basis function (RBF; notation given in Ref. [90]) as the activation function for the universal approximator network, given by

$$\phi(x) = \exp(-x^2). \quad (27)$$

The RBF is chosen for both theoretical and practical reasons. There is a substantial literature showing that the RBF is useful in the context of universal approximation [90–92]. Universal-approximation results motivate this function class but do not make a finite network size arbitrary. For noisy missing-physics problems, larger networks are not automatically preferable: Once the residual field can be represented, additional capacity increases the number of directions in which the fit can follow a particular noise realization. We therefore use a compact architecture and treat capacity as a regularization choice. For the networks shown in this work,

we use a depth of four layers, where for an input dimension N_1 and output dimension N_2 , each network has size $N_1 \times 10 \times 10 \times N_2$.

As the PEM portion of training requires interpolating the original training data to the solver time step, we use `DataInterpolations.jl` to do simple linear interpolation between observations [93]. The additional numerical work introduced by PEM is therefore the interpolation call and the feedback addition to the right-hand side; it does not add a separate filtering pass or a second optimization problem. Although there are many optimizers provided to train the UDEs, we have defaulted to using the AdamW method (decoupled weight decay and gradient updates) [94] in all of the results presented here.

The symbolic recovery of terms uses STLSQ solved using the ADMM [95], as implemented in `DataDrivenDiffEq.jl` [63]. Following Ref. [12], Θ is the feature library with weights ξ such that $y = \Theta\xi$, with sparsity enforced via LASSO [96]:

$$\arg \min_{\xi} \frac{1}{2} \|\Theta\xi - y\|_2 + \lambda \|\xi\|_1. \quad (28)$$

As STLSQ is faster than genetic algorithm approaches for large datasets ($N \gtrsim 10\,000$) [16], we default to this method. In principle, the trained UDE can generate subsamples of arbitrary density, so either approach can be used.

B. Background on the Rössler system

The Rössler system is a set of three nonlinear differential equations originally introduced as a simplified model for studying chaotic dynamics. It contains only one nonlinear term but still produces complex chaotic behavior. The dynamics are described by the three-dimensional ODE system:

$$\dot{x} = -y - z, \quad (29a)$$

$$\dot{y} = x + ay, \quad (29b)$$

$$\dot{z} = b + z(x - c). \quad (29c)$$

For specific parameter values ($a = 0.2$, $b = 0.22$, $c = 14$ in the case presented here), the system produces the funnel-shaped Rössler attractor, which exhibits sensitivity to initial conditions, fractal structure, and period-doubling chaotic activity. We chose this as an initial test of the PEM-UDE combination because, despite its simplicity, the Rössler system has chaotic properties including positive Lyapunov exponents, topological entropy, and various bifurcation sequences that have been used to study transitions between ordered and chaotic regimes in dynamical systems. The Rössler system has also been extended to analytically describe certain physical problems (e.g., mechanical vibrations, atmospheric variability).

In the results presented here, we use a universal approximator U to learn the dynamics of the entire second state, assisted by the observational data through the PEM technique. The system simulated during learning is therefore given by

$$\dot{x} = -y - z, \quad (30a)$$

$$\dot{y} = U(\theta, x, y, z) + K \cdot [\hat{y}(t) - y(t)], \quad (30b)$$

$$\dot{z} = b + z(x - c). \quad (30c)$$

For the results presented in the main text, we use $K = 0.25$. This choice is informed by the steepness of the parameter landscapes discussed in Sec. II; other reasonable values of this hyperparameter will not affect the accuracy of the learned dynamics, provided an appropriate solver is chosen. The results here are based on 100 000 training epochs with the AdamW optimizer [94]. The loss function used to train the UDE network is the mean-squared error between the predicted states at each time step and the true time series (with a small amount of noise added for robustness during training):

$$\ell(\theta) = \|\hat{\mathbf{x}} - \mathbf{x}\|^2. \quad (31)$$

For the parameters selected in our analyses, we characterized the degree of chaos via the Lyapunov time of the system as $\lambda^{-1} \approx 10$. Most parameter combinations in the surrounding space have similar values ($>80\%$ have $\lambda^{-1} \in [0, 25]$; see the code in the Supplemental Material [58] for computation of the empirical cumulative distribution function (CDF) of the maximum Lyapunov exponents). Lyapunov exponents were computed using standard functions offered in `ChaosTools.jl` [97,98].

C. Background on the Petrzela-Polak circuit

Petrzela and Polak [81] introduced a family of analog electrical circuits exhibiting chaotic dynamics under three constraints: a canonical form, structurally stable attractors, and novel dynamics. We work with a particular realization whose chaotic dynamics can be physically realized in a simple circuit, with the attractor observable on an oscilloscope (demonstrated in Ref. [60], from which we take the parameters and nondimensional form discussed here). For the circuit shown in Fig. 3(a), the equations describing the three observed states are

$$\dot{V}_1 = \frac{I}{C_1}, \quad (32a)$$

$$\dot{V}_2 = \frac{I_m - I - I_0}{C_2}, \quad (32b)$$

$$\dot{I} = \frac{V_2 - V_1 - IR_L}{L}. \quad (32c)$$

For convenience in numerical simulations, we work with the nondimensional form of the system (i.e., one in which physical constants are replaced with the minimal number of parameters to represent the same dynamics):

$$\dot{x} = z, \quad (33a)$$

$$\dot{y} = ax^2 - z - c, \quad (33b)$$

$$\dot{z} = y - x - bz. \quad (33c)$$

All parameters for both the dimensional and dimensionless systems are taken from Ref. [60] and are provided in Table II.

The form of the equations learned by the PEM-UDE method is given by

$$\dot{x} = z, \quad (34a)$$

$$\dot{y} = U(\theta, x, y, z) + K \cdot [\hat{y}(t) - y(t)], \quad (34b)$$

$$\dot{z} = y - x - bz. \quad (34c)$$

TABLE II. Physical and nondimensional parameters of the Petrzela-Polak circuit.

Parameter	Value
R_L	0.5 k Ω
R_M	0.08 k Ω
C_1	1.0 μ F
C_2	1.0 μ F
L	1.0 H
I_0	1.0 mA
Dimensionless parameter	
a	1.25
b	0.5
c	1.0

The loss function used to train this PEM-UDE combination is the same as Eq. (31), to test the robustness of the system to a poorly observed state. If it is known *a priori* that the observations of a state are corrupted to that degree, it could be worth modifying the loss in Eq. (31) to weight the error in this state less during training, or to drop the observations altogether. As shown in the previous section, the PEM-UDE training method can recover dynamics even when some states are completely hidden, provided the initial guess of their dynamics is sufficiently accurate.

D. Spiking neural networks and next-generation neural mass models

Next-generation neural mass models rely on the Ott-Antonsen (OA) ansatz to derive the mean-field activity of a population of neurons [44,48]. The OA ansatz was introduced as a way to bypass the need to set up an entire Fokker-Planck equation for the mean-field activity of a population of coupled oscillators, provided certain assumptions are met. The OA ansatz [48] assumes that the population of coupled oscillators exists in the thermodynamic limit (i.e., $N \rightarrow \infty$) and that the probability distribution of the observable (typically voltage) in phase space converges to a Lorentzian. The derivation can approximate other distributions to a reasonable degree of accuracy [50]. The OA ansatz approach also assumes that the noise input into each neuron receives a heterogeneous, quenched noise input η drawn from a general distribution. The currents cannot be stochastic; otherwise, the Lorentzian ansatz is violated. The OA ansatz further assumes that the conditional probabilities of the macroscopic variables are separable. This assumption and its limitations have been examined in detail [84], and a second-order circular cumulant approximation has been derived for neural mass models [66]. The OA ansatz also makes the assumption we address in this paper: The network being approximated has an all-to-all connectivity structure. As discussed above, this is a nonphysiological assumption for most regions of the brain, and removing it is needed to develop more accurate neural mass models.

We do not require the OA ansatz for the work here beyond the initial assumption of the functional form for population activity that will be used to train the UDE, so we omit most of the derivation and give only the main points. The equations shown here are adapted from earlier work that gives the

entire derivation [45]. The parameter set is inherited from that work and is intended to represent hippocampal-like Izhikevich population dynamics in the all-to-all limit; we do not tune these microscopic parameters to the experimental datasets. Following that work, we simulate a network of Izhikevich neurons that take the form

$$\dot{V}_j = V_j(V_j - \alpha) - w_j + \eta_j + I_{\text{ext}} + I_{\text{syn},j}, \quad (35a)$$

$$\dot{w}_j = a(bV_j - w_j), \quad (35b)$$

where V_j and w_j are the voltage and recovery current, respectively, of neuron $j \in [1, N]$ for a network of N neurons. The voltage activity in these neurons depends on the internal neuronal dynamics, the distributed noise source η_j , any external driving current I_{ext} , and the synaptic current from other interneuronal connections $I_{\text{syn},j}$. The synaptic activity is given by

$$I_{\text{syn},j} = g_{\text{syn}}s(e_r - V_j), \quad (36a)$$

$$\dot{s}_j = -\frac{s_j}{\tau_s} + \frac{s_{\text{jump}}}{N} \sum_{k=1}^N \sum_{t_k < t} \delta(t - t_k). \quad (36b)$$

Each neuron therefore receives an impulse of s_{jump}/N every time a presynaptic neuron k fires. If the network is structured to fulfill the assumptions discussed above, the mean-field activity of the entire population can be derived using the OA ansatz and takes the form

$$\dot{r} = \frac{\Delta}{\pi} + 2rv - (\alpha + g_{\text{syn}}s)r, \quad (37a)$$

$$\dot{v} = v(v - \alpha) - w + \bar{\eta} + I_{\text{ext}} + g_{\text{syn}}s(e_r - v) - (\pi r)^2, \quad (37b)$$

$$\dot{w} = a(bv - w) + w_{\text{jump}}r, \quad (37c)$$

$$\dot{s} = -\frac{s}{\tau_s} + s_{\text{jump}}r, \quad (37d)$$

where r is the population firing rate, v the mean voltage, w the mean recovery current, and s the mean synaptic activity. As discussed in earlier work, this is an accurate representation of the mean-field activity for fully connected networks [45]. Although this model class admits chaotic regimes for other parameter choices, the parameters in Table III produce the stable population oscillations studied here.

For all simulations presented here, we simulate a network of 1000 Izhikevich neurons with varying degrees of network sparsity, where the probability of two neurons forming a synapse is given by $p_c \in \{0.05, 0.1, 0.2, \dots, 1.0\}$, as indicated for the sparsities shown in Fig. 4. In a standard network of neurons, as the sparsity of connections is varied, the synaptic currents must be rescaled in proportion to the median in-degree of connectivity K to maintain balanced (synchronous) activity. Previous work has shown that rescaling the synaptic conductance with $g \propto 1/\sqrt{K}$, with similar rescaling of background currents, provides accurate scaling for a broad spectrum of network arrangements (in-degrees with Lorentzian, Gaussian, and Erdős-Rényi distributions) [50]. Following that work, we rescale synaptic conductance with p_c such that

$$s_{\text{jump}} = \frac{s_{\text{jump},0}}{\sqrt{p_c}}, \quad (38)$$

TABLE III. Parameters of the Izhikevich neuron population and NGNMM, from Ref. [45].

Parameter	Value	Interpretation
α	0.6215	Dimensionless Izhikevich voltage parameter
g_{syn}	1.2308	Synaptic conductance
I_{ext}	0	External current
e_r	1	Synaptic reversal potential
a	0.0077	Izhikevich recovery current parameter
b	-0.0062	Izhikevich recovery current parameter
w_{jump}	0.0189	Recovery current update per spike
τ_s	2.6	Synaptic time constant
s_{jump}	1.2308	Synaptic activity update per spike
$v_{\text{peak}}, v_{\text{reset}}$	200	Cutoffs for v_{∞}
$\bar{\eta}$	0.12	Mean of Lorentzian for applied noise
Δ	0.02	Width of Lorentzian for applied noise

with $s_{\text{jump},0} = s_{\text{jump}}\sqrt{1.5} \approx 1.5$. As in the previous sections, we set up the UDE to learn the unknown terms for the states we are interested in, but with two differences. We learn terms for both the first and second states in this derivation and so apply PEM to both of these states. We also modify Eq. (31) to train on only data for the first two states, since these are the only states typically observable in experimental data. The initial parameters for this system are given in Table III, and the formal description of the PEM-UDE system is given by

$$\dot{r} = \frac{\Delta}{\pi} + 2rv - (\alpha + g_{\text{syn}}s)r + U_1(\theta, r, v, w, s) + K_1 \cdot [\hat{r}(t) - r(t)], \quad (39a)$$

$$\dot{v} = v(v - \alpha) - w + \bar{\eta} + I_{\text{ext}} + g_{\text{syn}}s(e_r - v) - (\pi r)^2 + U_2(\theta, r, v, w, s) + K_2 \cdot [\hat{v}(t) - v(t)], \quad (39b)$$

$$\dot{w} = a(bv - w) + w_{\text{jump}}r, \quad (39c)$$

$$\dot{s} = -\frac{s}{\tau_s} + s_{\text{jump}}r. \quad (39d)$$

After learning the dynamics with the PEM-UDE approach and recovering the symbolic form of the UDE expressions via STLSQ, the new NGNMM with correction factors for sparsity is given by the parameter values in Table IV and by the functional form

$$\dot{r} = \frac{\Delta}{\pi} + 2rv - (\alpha + g_{\text{syn}}s)r + f_1(\vec{u}), \quad (40a)$$

$$\dot{v} = v(v - \alpha) - w + \bar{\eta} + I_{\text{ext}} + g_{\text{syn}}s(e_r - v) - (\pi r)^2 + f_2(\vec{u}), \quad (40b)$$

$$\dot{w} = a(bv - w) + w_{\text{jump}}r, \quad (40c)$$

$$\dot{s} = -\frac{s}{\tau_s} + s_{\text{jump}}r, \quad (40d)$$

$$f_1(\vec{u}) = \phi_1 s + \phi_2 rs + \phi_3 vs + c_1, \quad (40e)$$

$$f_2(\vec{u}) = \phi_4 r + \phi_5 rv + \phi_6 rs + \phi_7 s^2 + c_2. \quad (40f)$$

This form is fit only to the sparsity at 50% connectivity (see Fig. S5 of the Supplemental Material [58]) and requires additional tuning to different sparsities. The Sup-

TABLE IV. Parameters of the symbolic PEM-UDE-learned NGNMM at $p_c = 0.5$.

Parameter	Value	Interpretation
ϕ_1	-0.50	Reduction of synaptic contribution to firing rate
ϕ_2	-0.12	Rescaling of g_{syn}
ϕ_3	0.88	Voltage-dependent synaptic activity
ϕ_4	-5.01	Scaling of firing rate contribution to voltage
ϕ_5	9.80	Nonlinear interaction: firing rate $\times$ voltage
ϕ_6	-0.05	Nonlinear interaction: firing rate $\times$ synaptic
ϕ_7	-0.27	Quadratic synaptic contribution to voltage
c_1	0.12	Rescaling of Δ
c_2	0.06	Rescaling of $\bar{\eta}$

plemental Material [58] now gives the complete neural-fit sequence—training data and observables, optimizer and iteration count, gain continuation, symbolic extraction, extrapolation test, and interpretation of the learned terms—together with the limits on mechanistic interpretation.

E. Experimental data acquisition and preprocessing

Acquisition and preprocessing of each of the three experimental datasets are described in detail in their respective publications [72,74,76]; we describe them only briefly here. The rat MEC dataset [73] was acquired from six male Long-Evans rats, with electrode recordings taken during a long period of roaming in a circular track. The rat prefrontal cortex dataset [75] was acquired from four male Long-Evans rats during ten trials in which they successfully learned a behavioral task, including the rest periods before and after the task recordings. The human orbitofrontal cortex dataset [77] was acquired during electrocorticography recordings of ten neurosurgical patients undergoing evaluation for epilepsy. The patients played a decision-making game, and the recordings used for this analysis are the intertrial rest portions of the ECoG data. For all datasets, the synchrony between electrodes within the region was computed as the phase-locking value (PLV) between the angle of Hilbert-transformed signals (ϕ) from each electrode [99,100]. Following prior work [52], spike data from the datasets that recorded spike times rather than LFPs were converted to an LFP proxy by windowed spike counts, with the PLV calculation performed on these instead. PLV is given by

$$\text{PLV} = \left| \frac{1}{N} \sum_{t=1}^N e^{i[\phi_i(t) - \phi_j(t)]} \right| \quad (41)$$

for all discretized $t \in [1, N]$ and all pairs of electrodes (i, j).

ACKNOWLEDGMENTS

We thank Alex Driussi and Botond Antal for their valuable support with graphics. The research presented here was funded by the Baszucki Brain Research Fund. A.G.C.

acknowledges support from the National Institute of General Medical Sciences Medical Scientist Training Program under Award No. T32-GM008444.

Conceptualization: A.G.C., D.H., C.V.R., L.R.M.-P., H.H.S. Methodology: A.G.C., D.H., V.D., C.V.R., H.H.S. Software: A.G.C., D.H., V.D. Formal analysis: A.G.C., D.H., H.H.S. Investigation: A.G.C., D.H. Resources: E.K.M., R.H.G., A.E., L.R.M.-P., H.H.S. Data curation: A.G.C., D.H. Writing—original draft: A.G.C., D.H., H.H.S. Writing—review and editing: all authors. Visualization: A.G.C., D.H. Supervision: C.V.R., L.R.M.-P., H.H.S. Funding acquisition: L.R.M.-P., H.H.S.

C.V.R., E.K.M., R.H.G., A.E., L.R.M.-P., and H.H.S. are cofounders of Neuroblox Inc., a company spun out of Stony Brook University, MIT, and Dartmouth College to develop

a commercial-grade software platform for multiscale computational neuroscience with applications to diagnosis and treatment of brain-based disorders. The other authors declare no competing interests.

DATA AVAILABILITY

The intracranial electrode recording datasets can all be accessed in repositories on CRCNS.org, with the specific conditions of access to each one listed on their individual repository pages [73,75,77]. All code and simulated data needed to reproduce the results and plots in this manuscript, along with an abbreviated demonstration for rapid prototyping, are available in the GitHub repository associated with this publication [101].

- [1] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science* **324**, 81 (2009).
- [2] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Sparkes, K. E. Whelan, and A. Clare, The automation of science, *Science* **324**, 85 (2009).
- [3] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, Discovering symbolic models from deep learning with inductive biases, in *Advances in Neural Information Processing Systems 33*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H.-T. Lin (Curran Associates, Inc., Red Hook, NY, 2020), pp. 17429–17442.
- [4] O. Feinerman, I. Pinkoviezky, A. Gelblum, E. Fonio, and N. S. Gov, The physics of cooperative transport in groups of ants, *Nat. Phys.* **14**, 683 (2018).
- [5] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd ed. (Westview Press, Boulder, CO, 2015).
- [6] D. Kochkov, J. Yuval, I. Langmore, P. Norgaard, J. Smith, G. Mooers, M. Klöwer, J. Lottes, S. Rasp, P. Düben, S. Hatfield, P. Battaglia, A. Sanchez-Gonzalez, M. Willson, M. P. Brenner, and S. Hoyer, Neural general circulation models for weather and climate, *Nature (London)* **632**, 1060 (2024).
- [7] D. Foster, D. Comeau, and N. M. Urban, A Bayesian approach to regional decadal predictability: Sparse parameter estimation in high-dimensional linear inverse models of high-latitude sea surface temperature variability, *J. Clim.* **33**, 6065 (2020).
- [8] S. Kim, W. Ji, S. Deng, Y. Ma, and C. Rackauckas, Stiff neural ordinary differential equations, *Chaos* **31**, 093122 (2021).
- [9] A. Ziepkke, I. Maryshev, I. S. Aranson, and E. Frey, Multi-scale organization in communicating active matter, *Nat. Commun.* **13**, 6727 (2022).
- [10] N. Pagan and F. Dörfler, Game theoretical inference of human behavior in social networks, *Nat. Commun.* **10**, 5507 (2019).
- [11] T. J. Sejnowski, C. Koch, and P. S. Churchland, Computational neuroscience, *Science* **241**, 1299 (1988).
- [12] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **113**, 3932 (2016).
- [13] S. Beregi, D. A. W. Barton, D. Rezgui, and S. Neild, Using scientific machine learning for experimental bifurcation analysis of dynamic systems, *Mech. Syst. Sig. Process.* **184**, 109649 (2023).
- [14] S.-M. Udrescu and M. Tegmark, AI Feynman: A physics-inspired method for symbolic regression, *Sci. Adv.* **6**, eaay2631 (2020).
- [15] K. Ellis, L. Wong, M. Nye, M. Sablé-Meyer, L. Cary, L. Anaya Pozo, L. Hewitt, A. Solar-Lezama, and J. B. Tenenbaum, DreamCoder: Growing generalizable, interpretable knowledge with wake–sleep Bayesian program learning, *Philos. Trans. R. Soc. A* **381**, 20220050 (2023).
- [16] M. Cranmer, Interpretable machine learning for science with PySR and SymbolicRegression.jl, *arXiv:2305.01582*.
- [17] J. Bakarji, K. Champion, J. Nathan Kutz, and S. L. Brunton, Discovering governing equations from partial measurements with deep delay autoencoders, *Proc. R. Soc. A* **479**, 20230422 (2023).
- [18] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, Universal differential equations for scientific machine learning, *arXiv:2001.04385*.
- [19] L. Ljung, Prediction error estimation methods, *Circuits, Syst. Signal Process.* **21**, 11 (2002).
- [20] K. M. Hannay, D. B. Forger, and V. Booth, Macroscopic models for networks of coupled biological oscillators, *Sci. Adv.* **4**, e1701047 (2018).
- [21] M. Breakspear, Dynamic models of large-scale brain activity, *Nat. Neurosci.* **20**, 340 (2017).
- [22] A. G. Chesebro, L. R. Mujica-Parodi, and C. Weistuch, Ion gradient-driven bifurcations of a multi-scale neuronal model, *Chaos Solit. Fractals* **167**, 113120 (2023).
- [23] G. Abrevaya, M. Ramezani-Panahi, J.-C. Gagnon-Audet, P. Polosecki, I. Rish, S. P. Dawson, G. Cecchi, and G. Dumas, Effective latent differential equation models via attention and multiple shooting, *Transactions on Machine Learning Research* (2024), <https://openreview.net/forum?id=uxNfN2PU1W>.

- [24] O.E. Rössler, An equation for continuous chaos, *Phys. Lett. A* **57**, 397 (1976).
- [25] H. Schmidt and D. Avitabile, Bumps and oscillons in networks of spiking neurons, *Chaos* **30**, 033133 (2020).
- [26] N. Chandramoorthy and Q. Wang, On the probability of finding nonphysical solutions through shadowing, *J. Comput. Phys.* **440**, 110389 (2021).
- [27] R. M. Errico, What is an adjoint model?, *Bull. Am. Meteorol. Soc.* **78**, 2577 (1997).
- [28] A. Ni and C. Talnikar, Adjoint sensitivity analysis on chaotic dynamical systems by non-intrusive least squares adjoint shadowing (NILSAS), *J. Comput. Phys.* **395**, 690 (2019).
- [29] A. Ni, Q. Wang, P. Fernández, and C. Talnikar, Sensitivity analysis on chaotic dynamical systems by finite difference non-intrusive least squares shadowing (FD-NILSS), *J. Comput. Phys.* **394**, 615 (2019).
- [30] M. Chater, A. Ni, P. J. Blonigan, and Q. Wang, Least squares shadowing method for sensitivity analysis of differential equations, *SIAM J. Numer. Anal.* **55**, 3030 (2017).
- [31] Q. Wang, R. Hu, and P. Blonigan, Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations, *J. Comput. Phys.* **267**, 210 (2014).
- [32] J. P. Eckmann and D. Ruelle, Ergodic theory of chaos and strange attractors, *Rev. Mod. Phys.* **57**, 617 (1985).
- [33] K. E. Petersen, *Ergodic Theory*, *Cambridge Studies in Advanced Mathematics* (Cambridge University Press, Cambridge, 1983), Vol. 2.
- [34] E. Bradley and H. Kantz, Nonlinear time-series analysis revisited, *Chaos* **25**, 097610 (2015).
- [35] E. Adam, M. Kowalski, O. Akeju, E. K. Miller, E. N. Brown, M. M. McCarthy, and N. Kopell, Ketamine can produce oscillatory dynamics by engaging mechanisms dependent on the kinetics of NMDA receptors, *Proc. Natl. Acad. Sci. USA* **121**, e2402732121 (2024).
- [36] B. B. Antal, A. G. Chesebro, H. H. Strey, L. R. Mujica-Parodi, and C. Weistuch, Achieving Occam’s razor: Deep learning for optimal model reduction, *PLoS Comput. Biol.* **20**, e1012283 (2024).
- [37] C. Weistuch, L. R. Mujica-Parodi, R. M. Razban, B. Antal, H. van Nieuwenhuizen, A. Amgalan, and K. A. Dill, Metabolism modulates network synchrony in the aging brain, *Proc. Natl. Acad. Sci. USA* **118**, e2025727118 (2021).
- [38] F. Habibollahi, B. J. Kagan, A. N. Burkitt, and C. French, Critical dynamics arise during structured information presentation within embodied *in vitro* neuronal networks, *Nat. Commun.* **14**, 5287 (2023).
- [39] Y. Tian, Z. Tan, H. Hou, G. Li, A. Cheng, Y. Qiu, K. Weng, C. Chen, and P. Sun, Theoretical foundations of studying criticality in the brain, *Network Neurosci.* **6**, 1148 (2022).
- [40] I. R. Graf and B. B. Machta, A bifurcation integrates information from many noisy ion channels and allows for milli-Kelvin thermal sensitivity in the snake pit organ, *Proc. Natl. Acad. Sci. USA* **121**, e2308215121 (2024).
- [41] H. R. Wilson and J. D. Cowan, Excitatory and inhibitory interactions in localized populations of model neurons, *Biophys. J.* **12**, 1 (1972).
- [42] B. H. Jansen and V. G. Rit, Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns, *Biol. Cybern.* **73**, 357 (1995).
- [43] R. Larter, B. Speelman, and R. M. Worth, A coupled ordinary differential equation lattice model for the simulation of epileptic seizures, *Chaos* **9**, 795 (1999).
- [44] E. Montbrió, D. Pazó, and A. Roxin, Macroscopic description for networks of spiking neurons, *Phys. Rev. X* **5**, 021028 (2015).
- [45] L. Chen and S. A. Campbell, Exact mean-field models for spiking neural networks with adaptation, *J. Comput. Neurosci.* **50**, 445 (2022).
- [46] S. Coombes and A. Byrne, Next generation neural mass models, in *Nonlinear Dynamics in Computational Neuroscience*, edited by F. Corinto and A. Torcini (Springer International Publishing, Cham, 2019), pp. 1–16.
- [47] S. H. Strogatz, From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators, *Physica D* **143**, 1 (2000).
- [48] E. Ott and T. M. Antonsen, Low dimensional behavior of large systems of globally coupled oscillators, *Chaos* **18**, 037113 (2008).
- [49] D. S. Goldobin, M. di Volo, and A. Torcini, Discrete synaptic events induce global oscillations in balanced neural networks, *Phys. Rev. Lett.* **133**, 238401 (2024).
- [50] M. di Volo and A. Torcini, Transition from asynchronous to oscillatory dynamics in balanced spiking networks with instantaneous synapses, *Phys. Rev. Lett.* **121**, 128301 (2018).
- [51] D. S. Goldobin, M. di Volo, and A. Torcini, Reduction methodology for fluctuation driven population dynamics, *Phys. Rev. Lett.* **127**, 038301 (2021).
- [52] A. Pathak, S. L. Brincat, H. Organtzidis, H. H. Strey, S. Senneff, E. G. Antzoulatos, L. R. Mujica-Parodi, E. K. Miller, and R. Granger, Biomimetic model of corticostriatal micro-assemblies discovers a neural code, *Nat. Commun.* **17**, 390 (2026).
- [53] A. Balcioglu, R. Gillani, M. Doron, K. Burnell, T. Ku, A. Erisir, K. Chung, I. Segev, and E. Nedivi, Mapping thalamic innervation to individual L2/3 pyramidal neurons and modeling their “readout” of visual input, *Nat. Neurosci.* **26**, 470 (2023).
- [54] R. Fruengel and M. Oberlaender, Sparse connectivity enables efficient information processing in cortex-like artificial neural networks, *Front. Neural Circuits* **19**, 1528309 (2025).
- [55] P. Stoica and A. Nehorai, On multistep prediction error methods for time series models, *J. Forecast.* **8**, 357 (1989).
- [56] R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung, Direct prediction-error identification of unstable nonlinear systems applied to flight test data, in *15th IFAC Symposium on System Identification*, IFAC Proceedings Volumes, Vol. 42 (Elsevier, Amsterdam, 2009), pp. 144–149.
- [57] L. M. Pecora and T. L. Carroll, Synchronization in chaotic systems, *Phys. Rev. Lett.* **64**, 821 (1990).
- [58] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/xtb5-mlc8> for additional loss-landscape and parameter-sensitivity analyses, details of neural-population fitting and symbolic recovery, extrapolation tests, and code for computing the Lyapunov-exponent distribution.
- [59] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer Series in Statistics (Springer, New York, NY, 2009).

- [60] J. C. Sprott and W. J.-C. Thio, *Elegant Circuits: Simple Chaotic Oscillators* (World Scientific, Singapore, 2022).
- [61] N. Kuznetsov, T. Mokaev, V. Ponomarenko, E. Seleznev, N. Stankevich, and L. Chua, Hidden attractors in Chua circuit: Mathematical theory meets physical experiments, *Nonlin. Dyn.* **111**, 5859 (2023).
- [62] P. J. Gonçalves, J.-M. Lueckmann, M. Deistler, M. Nonnenmacher, K. Öcal, G. Bassetto, C. Chintaluri, W. F. Podlaski, S. A. Haddad, T. P. Vogels, D. S. Greenberg, and J. H. Macke, Training deep neural density estimators to identify mechanistic models of neural dynamics, *eLife* **9**, e56261 (2020).
- [63] J. Martensen, C. Rackauckas *et al.*, SciML/DataDrivenDiffEq.jl: v0.6.4 (Version v0.6.4) [Computer software], Zenodo, 2021, <https://doi.org/10.5281/zenodo.5083412>.
- [64] M. Breakspear, J. R. Terry, and K. J. Friston, Modulation of excitatory synaptic coupling facilitates synchronization and complex dynamics in a biophysical model of neuronal dynamics, *Network: Comput. Neural Syst.* **14**, 703 (2003).
- [65] J. B. Ding, J. N. Guzman, J. D. Peterson, J. A. Goldberg, and D. J. Surmeier, Thalamic gating of corticostriatal signaling by cholinergic interneurons, *Neuron* **67**, 294 (2010).
- [66] M. di Volo, M. Segneri, D. S. Goldobin, A. Politi, and A. Torcini, Coherent oscillations in balanced neural networks driven by endogenous fluctuations, *Chaos* **32**, 023120 (2022).
- [67] P. Clusella and E. Montbrió, Exact low-dimensional description for fast neural oscillations with low firing rates, *Phys. Rev. E* **109**, 014229 (2024).
- [68] E. V. Lubenov and A. G. Siapas, Hippocampal theta oscillations are travelling waves, *Nature (London)* **459**, 534 (2009).
- [69] M. Halgren, I. Ulbert, H. Bastuji, D. Fabó, L. Erőss, M. Rey, O. Devinsky, W. K. Doyle, R. Mak-McCully, E. Halgren, L. Wittner, P. Chauvel, G. Heit, E. Eskandar, A. Mandell, and S. S. Cash, The generation and propagation of the human alpha rhythm, *Proc. Natl. Acad. Sci. USA* **116**, 23772 (2019).
- [70] W. Nicola and S. A. Campbell, Mean-field models for heterogeneous networks of two-dimensional integrate and fire neurons, *Front. Comput. Neurosci.* **7**, 184 (2013).
- [71] M. K. Nandi, M. Valla, and M. di Volo, Bursting gamma oscillations in neural mass models, *Front. Comput. Neurosci.* **18**, 1422159 (2024).
- [72] J. J. Hernández-Pérez, K. W. Cooper, and E. L. Newman, Medial entorhinal cortex activates in a traveling wave in the rat, *eLife* **9**, e52289 (2020).
- [73] J. J. Hernández-Pérez, K. W. Cooper, and E. L. Newman, Extracellular recordings from across the dorsoventral axis of the medial entorhinal cortex of the rat, CRCNS. 2020, <https://crcns.org/data-sets/hc/hc-27>.
- [74] A. Peyrache, M. Khamassi, K. Benchenane, S. I. Wiener, and F. P. Battaglia, Replay of rule-learning related neural patterns in the prefrontal cortex during sleep, *Nat. Neurosci.* **12**, 919 (2009).
- [75] A. Peyrache, M. Khamassi, K. Benchenane, S. Wiener, and F. Battaglia, Activity of neurons in rat medial prefrontal cortex during learning and sleep, CRCNS. 2018, <https://crcns.org/data-sets/pfc/pfc-6>.
- [76] I. Saez, J. Lin, A. Stolk, E. Chang, J. Parvizi, G. Schalk, R. T. Knight, and M. Hsu, Encoding of multiple reward-related computations in transient and sustained high-frequency activity in human OFC, *Curr. Biol.* **28**, 2889 (2018).
- [77] I. Saez, J. Lin, A. Stolk, E. Chang, J. Parvizi, G. Schalk, R. T. Knight, and M. Hsu, High-frequency activity of human orbitofrontal sites during decision-making play, CRCNS. 2018, <https://crcns.org/data-sets/ofc/ofc-3>.
- [78] J. O'Leary, J. A. Paulson, and A. Mesbah, Stochastic physics-informed neural ordinary differential equations, *J. Comput. Phys.* **468**, 111466 (2022).
- [79] J. Sun, Y. Huang, W. Yu, and A. Garcia-Ortiz, Nonlinear system identification Prediction error method vs neural network, in *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)* (IEEE, Piscataway, NJ, 2021), pp. 1–4.
- [80] G. Pilonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, Deep networks for system identification: A survey, *Automatica* **171**, 111907 (2025).
- [81] J. Petrzela and L. Polak, Minimal realizations of autonomous chaotic oscillators based on trans-immittance filters, *IEEE Access* **7**, 17561 (2019).
- [82] Y. Jiang, X. Niu, H. Chen, J. Zhu, K. Xie, and M. Perc, Ensemble adaptive libraries and inner-product sparse regression for robust nonlinear dynamics discovery, *Appl. Math. Comput.* **522**, 129839 (2026).
- [83] H. Sheheiti and V. Jirsa, Incorporating slow NMDA-type receptors with nonlinear voltage-dependent magnesium block in a next generation neural mass model: Derivation and dynamics, *J. Comput. Neurosci.* **52**, 207 (2024).
- [84] D. S. Goldobin and A. V. Dolmatova, Ott-Antonsen ansatz truncation of a circular cumulant series, *Phys. Rev. Res.* **1**, 033139 (2019).
- [85] L. Paninski, Maximum likelihood estimation of cascade point-process neural encoding models, *Network: Comput. Neural Syst.* **15**, 243 (2004).
- [86] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects, *J. Neurophysiol.* **93**, 1074 (2005).
- [87] E. Nozari, M. A. Bertolero, J. Stiso, L. Caciagli, E. J. Cornblath, X. He, A. S. Mahadevan, G. J. Pappas, and D. S. Bassett, Macroscopic resting-state brain dynamics are best described by linear models, *Nat. Biomed. Eng.* **8**, 68 (2024).
- [88] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, Stiff-PINN: Physics-informed neural network for stiff chemical kinetics, *J. Phys. Chem. A* **125**, 8098 (2021).
- [89] J. A. Roberts, L. L. Gollo, R. G. Abeysuriya, G. Roberts, P. B. Mitchell, M. W. Woolrich, and M. Breakspear, Metastable brain waves, *Nat. Commun.* **10**, 1056 (2019).
- [90] R. J. Craddock and K. Warwick, Multi-layer radial basis function networks. An extension to the radial basis function, in *Proceedings of the International Conference on Neural Networks (ICNN'96)*, Vol. 2 (IEEE, Piscataway, NJ, 1996), pp. 700–705.
- [91] J. Park and I. W. Sandberg, Universal approximation using radial-basis-function networks, *Neural Comput.* **3**, 246 (1991).

- [92] Y. Zhao, J. Pei, and H. Chen, Multi-layer radial basis function neural network based on multi-scale kernel learning, *Appl. Soft Comput.* **82**, 105541 (2019).
- [93] S. Bhagavan, B. de. Koning, S. Maddhashiya, and C. Rackauckas, DataInterpolations.jl: Fast interpolations of 1D data, *J. Open Source Software* **9**, 6917 (2024).
- [94] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *International Conference on Learning Representations* (ICLR, 2019).
- [95] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foun. Trends@ Mach. Learn.* **3**, 1 (2010).
- [96] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. B* **58**, 267 (1996).
- [97] G. Datseris, DynamicalSystems.jl: A Julia software library for chaos and nonlinear dynamics, *J. Open Source Software* **3**, 598 (2018).
- [98] G. Datseris and U. Parlitz, *Nonlinear Dynamics: A Concise Introduction Interlaced with Code* (Springer Nature, Cham, Switzerland, 2022).
- [99] Y. Kuramoto, in *Chemical Oscillations, Waves, and Turbulence*, edited by H. Haken (Springer, Berlin, 1984), Vol. 19.
- [100] F. Mormann, K. Lehnertz, P. David, and C. E. Elger, Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients, *Physica D* **144**, 358 (2000).
- [101] Neuroblox, pem-ude Companion code and simulated data, GitHub, 2026, GitHub repository: <https://github.com/Neuroblox/pem-ude>.