Neuronal temporal filters as normal mode extractors

Siavash Golkar [ORCID],* Jules Berman,* and David Lipshutz [ORCID]
Center for Computational Neuroscience, Flatiron Institute, New York, 10010 New York, USA

Robert Mihai Haret and Tim Gollisch [ORCID]
University Medical Center Göttingen, Department of Ophthalmology
and Bernstein Center for Computational Neuroscience Göttingen, 37075 Göttingen, Germany

Dmitri B. Chklovskii
Center for Computational Neuroscience, Flatiron Institute
and Neuroscience Institute, NYU Medical Center, New York, 10016 New York, USA

This article is part of the Physical Review Research *collection titled* Physics of Neuroscience.

To generate actions in the face of physiological delays, the brain must predict the future. Here we explore how prediction may lie at the core of brain function by considering a neuron predicting the future of a scalar time series input. Assuming that the dynamics of the lag vector (a vector composed of several consecutive elements of the time series) are locally linear, normal mode decomposition decomposes the dynamics into independently evolving (eigen)modes allowing for straightforward prediction. We propose that a neuron learns the top mode and projects its input onto the associated subspace. Under this interpretation, the temporal filter of a neuron corresponds to the left eigenvector of a generalized eigenvalue problem. We mathematically analyze the operation of such an algorithm on noisy observations of synthetic data generated by a linear system. Interestingly, the shape of the temporal filter varies with the signal-to-noise ratio (SNR): a noisy input yields a monophasic filter and a growing SNR leads to multiphasic filters with progressively greater number of phases. Such variation in the temporal filter with input SNR resembles that observed experimentally in biological neurons.

I. INTRODUCTION

The brain must generate behavior in the face of physiological delays on multiple levels: from sensory transduction to axonal conduction to synaptic transmission and muscle activation. To compensate for such delays, it would be useful even for individual neurons to predict future inputs. Not surprisingly, the paradigm of optimal prediction has been used to derive normative models of neurons. Such models can be loosely classified into two categories [1,2]: predictive coding [3–6] and predictive information [7,8].

In predictive coding, a neuron computes an optimal prediction of the signal and subtracts it from the actual value, transmitting prediction error downstream [3–5]. This results in an optimal linear filter whose shape depends on the statistics of the input. Predictive coding approximately accounts for the change in the temporal receptive field with the input SNR [Fig. 1(a)] but fails to reproduce the exact shape of the filter because it has a narrow unitary peak at the exact time of the actual signal (usually present time) [3]. Such peak can be smoothed by adding a denoising objective but that introduces additional parameters to the model.

In predictive information, a neuron filters the past signal to preserve only the part that is most informative about the future [2,7,8]. Such framework expands the assortment of temporal filters depending on the various parameters such as the balance between the past/output and output/future mutual information. Recently, it was shown that multiphasic filter can be derived using this formalism as well [9].

A shortcoming of both approaches is that they require setting the time point relative to the present for which the neuron generates prediction. Such formulation of the prediction problem does not seem suitable when neurons might have different amounts of delay, or the goal is to predict (and act upon) a trend or a mode of the signal that spans multiple time points in the future. Moreover, both approaches view the neuronal filter as a tradeoff between different mathematical terms in the optimization objective and therefore depend on the relative weighting of these terms. Therefore, the shape of the temporal kernel depends on these, as well as other, hyperparameters of the model.

An alternative approach to prediction assumes that the sensory stimuli are generated by a (generally nonlinear) dynamical system. If such dynamics can be learned by the

---

*These authors contributed equally to this work.
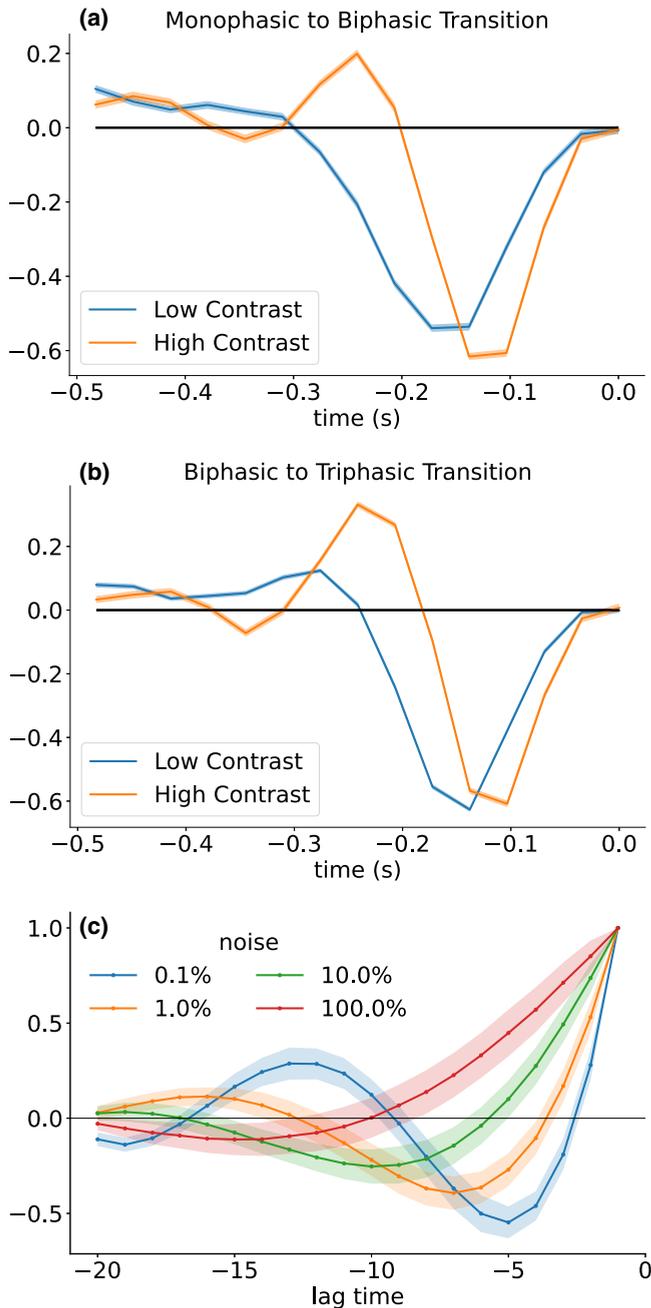
FIG. 1. Neuronal temporal filters. (a), (b) Experimentally obtained by spike-triggered average (STA) from two retinal ganglion cells stimulated by white noise of different contrast. (c) Top left eigenvectors obtained in our framework for different levels of noise in a synthetic example. Shading shows standard deviation. Whereas theoretical filters peak when the neuron spikes, experimental ones peak before that. This artifact is due to signal processing delays from photoabsorption to the ganglion cell firing and the smoothness of experimental filters resulting from the low-pass nature of biophysical processes.

brain from previously seen trajectories, then prediction can be cast as identifying long-living (growing or slow) modes (or manifolds) in the input and extrapolating them into the future. To take the first step in this direction, we take advantage of the fact that in the vicinity of a hyperbolic fixed point the

invariant manifolds can be well approximated by the invariant subspaces of the linearized dynamics (see e.g., Sec. 19.12a of Ref. [10]). Therefore, a neuron that identifies the least decaying linear mode can identify an invariant manifold of a fixed point and generate a nontrivial prediction.

Here, we explore the hypothesis that a neuron extracts the least decaying mode from the incoming scalar time series as in normal mode decomposition [11]. To cast this problem in the language of dynamics, we construct time-lag vectors by windowing the scalar time series. We assume that the system is in the vicinity of a hyperbolic fixed point and identify the linear dynamics of the lag vectors. The eigendecomposition of the linear transition matrix yields the eigenmodes that evolve independently as the right eigenvectors and the corresponding linear filters as the left eigenvectors. Therefore, the top left eigenvector yields a linear filter that projects onto the most dominant mode, representing neuronal output. Thus, unlike the previous predictive coding and predictive information approaches, which optimize prediction for a certain time in the future, in our proposal, a neuron learns a rank-one approximation of the dynamical system generating the input and outputs the top mode, which develops into the future in a predictable way.

Our theoretical proposal makes a prediction that can be compared with experimental data. We find that the shape of the top left eigenvector depends on the input SNR: monophasic filter at low SNR with adding phases for growing SNR [Fig. 1(c)]. This prediction is reminiscent of the SNR-induced variation in the form of the spike-triggered average (STA) of various types of biological neurons [Figs. 1(a) and 1(b)] [3,12–16]. Although this does not prove our proposal it suggests that this approach may be a step in the right direction.

II. NORMAL MODE DECOMPOSITION (NMD)

In this section, we review NMD [11] in the context of a neuron with a single input and a single output (SISO).[1]

A. Problem statement and linearization

Let $\{x(t)\}_{t=1,2,\dots}$ be a scalar time series satisfying the nonlinear relation

$$x(t+1) = f(x(t), \dots, x(t-n)), \quad t \geqslant n+1, \quad (1)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and $n \geqslant 1$ is referred to as the lag. We say $x^* \in \mathbb{R}$ is a fixed point if $x^* = f(x^*, \dots, x^*)$. We embed a scalar time series $x(t)$, $t = 1, \dots, T+n$, into a sequence of lag vectors $\mathbf{x}_t = [x(t-n+1), \dots, x(t)]^\top \in \mathbb{R}^n$. The dynamics of the first $n-1$ components of the lag vector is given by a simple shift in the time step [i.e., $x(t-i) \to x(t-i+1)$] and the dynamics of the final component $x(t)$ is given by Eq. (1) [i.e., $x(t) \to f(x(t), \dots, x(t-n))$]. These can be summarized as $\mathbf{x}_{t+1} =$

$\mathbf{F}(\mathbf{x}_t)$, where the function $\mathbf{F}$ is given by

$$\mathbf{F}(\mathbf{x}_t) = \mathbf{F}\left(\begin{bmatrix} x(t-n+1) \\ \vdots \\ x(t-1) \\ x(t) \end{bmatrix}\right) \equiv \begin{bmatrix} x(t-n+2) \\ \vdots \\ x(t) \\ f(x(t), \dots, x(t-n)) \end{bmatrix}.$$
$$(2)$$

We assume that the dynamics are dominated by the presence of a hyperbolic fixed point at $\mathbf{x}^*$. We approximate dynamics (2) by expanding around this fixed point. Defining $\delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}^*$, we have:

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}^*) + \nabla\mathbf{F}(\mathbf{x}^*)\delta\mathbf{x}_t + o(\|\delta\mathbf{x}_t\|)$$
$$\approx \mathbf{x}^* + \nabla\mathbf{F}(\mathbf{x}^*)\delta\mathbf{x}_t,$$

where we have used the fact that $\mathbf{F}(\mathbf{x}^*) = \mathbf{x}^*$ and $\nabla\mathbf{F}(\mathbf{x}^*)$ is the Jacobian evaluated at $\mathbf{x}^*$. After subtracting $\mathbf{x}^*$ from both sides:

$$\delta\mathbf{x}_{t+1} \approx \nabla\mathbf{F}(\mathbf{x}^*)\delta\mathbf{x}_t.$$

Letting $\mathbf{A} := \nabla\mathbf{F}(\mathbf{x}^*)$ and assuming without loss of generality that $\mathbf{x}^* = \mathbf{0}$, we have the following linear approximation to the nonlinear dynamics in Eq. (2) near the fixed point:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t. \tag{3}$$

From the definition of the function $\mathbf{F}$ in Eq. (2), we see that $\mathbf{A} = \nabla\mathbf{F}(\mathbf{0})$ takes a companion matrix form [17]

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ c_1 & c_2 & \dots & & c_n \end{bmatrix}, \tag{4}$$

where $c_i = \partial f(\mathbf{x}^*)/\partial x^i$.

B. Eigendecomposition of the dynamics

The matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is generically a non-normal matrix possessing an eigendecomposition:

$$\mathbf{A} = \sum_i \lambda_i \mathbf{w}_i \mathbf{v}_i^\top \quad \text{s.t.} \quad \mathbf{v}_i^\top \mathbf{w}_j = \delta_{ij}, \tag{5}$$

where λ_i are the eigenvalues and $\mathbf{w}_i$ ($\mathbf{v}_i^\top$) are the right (left) eigenvectors. Here, we consider matrix $\mathbf{A}$ describing a hyperbolic fixed point ($\lambda_i \neq 1$). Moreover, we assume that all eigenvalues are real and are sorted in descending order $\lambda_i \geqslant \lambda_{i+1}$. We also assume that the top mode is nondegenerate (i.e., $\lambda_1 > \lambda_2$).

The normal modes are found by eigendecomposition of $\mathbf{A}$. As a matrix of companion form, $\mathbf{A}$ is diagonalized using the Vandermonde matrix and its inverse [17]:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1},$$

where $\mathbf{V}$ is the Vandermonde matrix

$$\mathbf{V} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

and $\mathbf{\Lambda} := \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ is the diagonal matrix of eigenvalues λ_i, which are the roots of the characteristic polynomial with coefficients c_i. The left eigenvectors of $\mathbf{A}$, are given by the rows of the inverse of the Vandermonde matrix $\mathbf{V}^{-1}$. It is easily verified that the right eigenvectors of $\mathbf{A}$, corresponding to the columns of the Vandermonde matrix, represent the individual modes of the dynamical system.

C. Projecting onto the dominant mode

We are interested in finding the dominant exponential mode of the dynamics. As stated above, the dominant exponential mode is represented as the top right eigenvector of $\mathbf{A}$. Therefore, by projecting onto the top right eigenvector, we can isolate the dominant mode.

Because of the biorthogonality of the left and the right eigenvectors, the top left eigenvector (henceforth referred to as $\mathbf{v}_1$) is orthogonal to all but the top right eigenvector. This allows us to use $\mathbf{v}_1$ as a projector that zeros out all but the most dominant mode of the dynamics. We therefore propose that it is the neuron's goal to learn the top left eigenvector and project its input onto this vector.

In order to learn the top left eigenvector, one approach would be to first find matrix $\mathbf{A}$ and then perform the eigendecomposition on the inferred $\mathbf{A}$ matrix. This matrix can be found from data $x(t)$ by minimizing the mean-squared error:

$$\min_{\mathbf{A}} \sum_t \|\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t\|^2. \tag{6}$$

Equation (6) can be solved via the following system of equations:

$$\mathbf{X}_+\mathbf{X}^\top = \mathbf{A}\mathbf{X}\mathbf{X}^\top, \tag{7}$$

where we introduce a matrix notation $\mathbf{X} = [\mathbf{x}_n, \dots, \mathbf{x}_{T+n-1}]$ and $\mathbf{X}_+ = [\mathbf{x}_{n+1}, \dots, \mathbf{x}_{T+n}]$.

In this paper, however, instead of directly solving for $\mathbf{A}$ via Eqs. (6) and (7), we substitute the eigendecomposition, Eq. (5) and multiply both sides by $\mathbf{v}_i^\top$ on the left, leading to the equivalent generalized eigenvalue problem:

$$\mathbf{v}_i^\top \mathbf{X}_+\mathbf{X}^\top = \lambda_i \mathbf{v}_i^\top \mathbf{X}\mathbf{X}^\top. \tag{8}$$

This allows us to circumvent the intermediate step of computing the $\mathbf{A}$ matrix explicitly.

III. DETECTING THE DOMINANT MODE

In this section, we apply our formalism to time series synthetically generated by a linear dynamical system of order k.

A. Problem formulation

As stated in Sec. I, we assume that we are in the vicinity of a hyperbolic fixed point of the dynamics of the lag vector. This means that the dynamics of the linearized system can be decomposed as the sum of k exponentials with real exponents determined by the Jacobian of f at the fixed point and coefficients determined by the initial conditions of the system. We
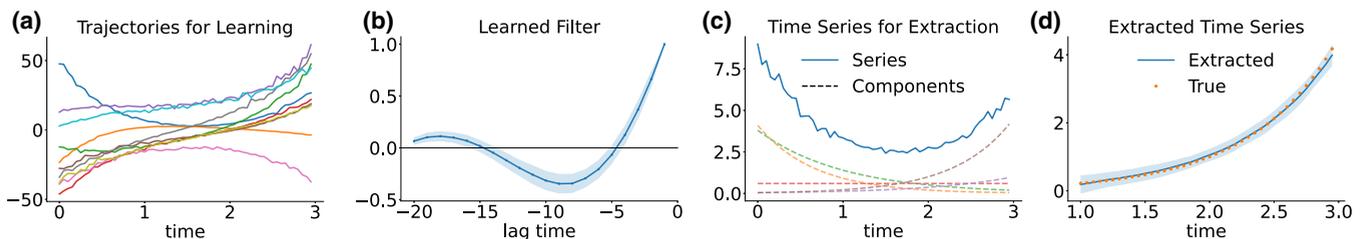
FIG. 2. Problem formulation. (a) We generate a number of trajectories in the vicinity of a hyperbolic fixed point. Each trajectory is comprised of a number of growing and decaying exponentials and white noise [see (c)]. (b) From these trajectories, we learn the top left eigenvector of the inferred **A** matrix. We use this as the time kernel of our proposed neuron. (c) We apply the time kernel to a previously unseen trajectory from the same dynamical system. (d) We extract the most dominant mode by convolving the time series with the computed time kernel. In this example, the presence of the dominant mode is clear from the projection starting at $t = 1$. However, if we only look at the time series in (c), the presence of a growing mode would not be clear until around time $t = 2$. Dynamics details: The time series is comprised of five different exponents with exponents given by $\{-1.5, -1, 0, 1, 1.5\}$. The time series is discretized with time step 0.05.

also assume that an uncorrelated observation noise (see Fig. 2)

$$x(t) = \sum_i^k c_i e^{a_i t} + \eta(t). \tag{9}$$

In general, NMD extracts the exponents via an eigendecomposition, i.e., the eigenvalues of **A**, $\lambda_i = e^{a_i}$. Since a neuron has a single output it can extract only one constituent. Because the greatest contribution to the future is given by the largest exponent (Fig. 2) we propose that a neuron learns the top left eigenvector of **A** and projects its lag vector, $\mathbf{x}_t$, onto this eigenvector, which can then be used for prediction and control.

B. Dependence of the filter on the noise level

In this section, we look at the effect of noise on the shape of the time kernel. We first discuss the effect of additive Gaussian noise analytically, we then verify this numerically on synthetically generated data. Previous work analyzing the application of NMD to data with observation noise have primarily focused on extracting the true eigenvalues and eigenvectors of the system, that is the dynamic modes of the dynamical system in the absence of noise [18,19]. Here, we are interested in extracting the most dominant mode from noisy data, and as we will see (Fig. 5), the eigenvector of the noiseless system is not a good candidate for this task.

a. Analytical calculation. Let us assume that the time series given in Eq. (9) has additive white observation noise, that is, we assume that **X** satisfies Eq. (3) but we observe $\mathbf{X}_\epsilon = \mathbf{X} + \eta$ with $\eta \sim \mathcal{N}(0, \epsilon^2)$. We denote the noiseless data and ground truth dynamics by **X** and **A**, and call the data and the inferred dynamics in the presence of noise $\mathbf{X}_\epsilon$ and $\mathbf{A}_\epsilon$. When averaging over the different draws of the noise, we have $\mathbb{E}_\eta(\mathbf{X}_\epsilon \mathbf{X}_\epsilon^\top) = \mathbf{X}\mathbf{X}^\top + \epsilon^2 \mathbf{I}_k$ and $\mathbb{E}_\eta(\mathbf{X}_{\epsilon+}\mathbf{X}_\epsilon^\top) = \mathbf{X}_+\mathbf{X}^\top + \epsilon^2 \mathbf{S}$, where $\mathbf{I}_k$ is the identity matrix and **S** is the matrix of off-diagonal ones:

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ 0 & 0 & \cdots & & 0 \end{bmatrix}.$$

Note that here $\mathbb{E}_\eta$ denotes averaging only over the noise draws and the averaging over the samples is performed in the product of data matrices $\mathbf{X}_+\mathbf{X}^\top$ and $\mathbf{X}_+\mathbf{X}^\top$. The putative $\mathbf{A}_\epsilon$ matrix (that is the **A** matrix that minimizes the MSE objective with noisy data $\mathbf{X}_\epsilon$) is now given by

$$\begin{aligned} \mathbf{A}_\epsilon &= (\mathbf{X}_{\epsilon+}\mathbf{X}_\epsilon^\top)(\mathbf{X}_\epsilon \mathbf{X}_\epsilon^\top)^{-1} \\ &= (\mathbf{A}\mathbf{X}\mathbf{X}^\top + \epsilon^2 \mathbf{S})(\mathbf{X}\mathbf{X}^\top + \epsilon^2 \mathbf{I}_k)^{-1}. \end{aligned} \tag{10}$$

That is, given the ground truth dynamics and noiseless data covariances (i.e., **A** and $\mathbf{X}\mathbf{X}^\top$), we can determine the inferred dynamics and eigenvectors in the presence of white noise. We provide further details of this computation in Appendix A. While the eigenvalue equations are derived analytically, they are not solvable in closed form and we look at their numerical solutions for varying noise and exponents. Here, we summarize the results and provide some intuition.

When looking at the eigenvalues of $\mathbf{A}_\epsilon$ (Fig. 3), we see that as the noise level ϵ increases, the real parts slowly decrease and eventually cross zero. At the same time, the imaginary parts of the eigenvalues start at zero and grow when the real part becomes negative. This is intuitively understandable: when the increased amount of noise reduces the ability of the system to differentiate different modes, it starts replacing these with complex oscillatory modes.

Furthermore, at the noise value for which a mode's eigenvalue's real part crosses zero, we see that the top left eigenvector's number of phases decreases by one. This is due to the fact that the multiphasic structure of the left eigenvectors are responsible for the biorthogonal structure of the left/right eigenvectors.

b. Numerical verification. Above we looked at the eigenvectors of $\mathbf{A}_\epsilon$, i.e., after averaging over different noise draws. Here, we look at these for specific noise draws, that is, for specific simulations of linear dynamical systems. We looked at the response of the algorithm to synthetic data constructed from dynamics in the vicinity of a hyperbolic fixed point. We initiated a number of trajectories with different initial conditions and noise draws. Our algorithm finds the top eigenvector on the dynamics inferred from these trajectories. We then applied the learned eigenvector to a previously unseen trajectory. Figure 2 (right) shows the projection of this time series onto the largest left eigenvector and we see that we
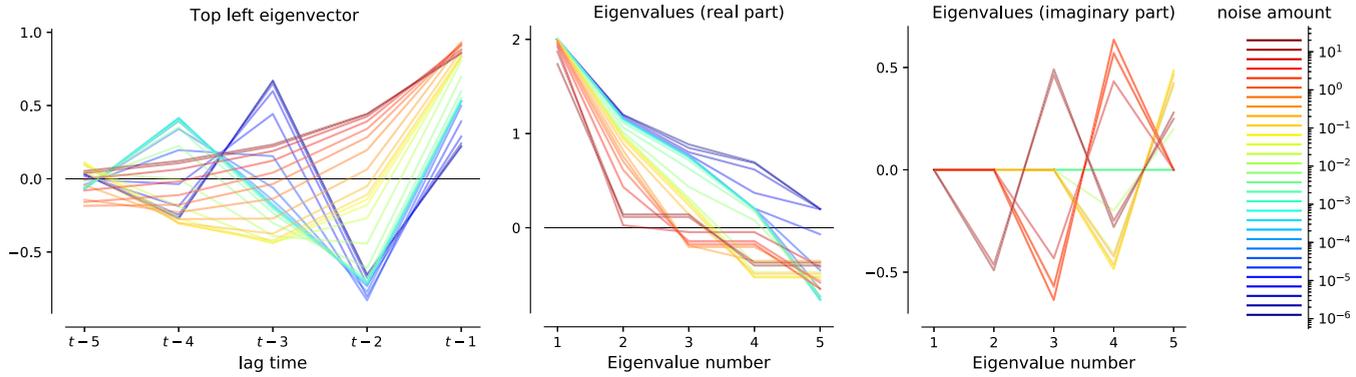
FIG. 3. Analytically derived eigenvectors and eigenvalues for the lag $n = 5$ for different noise levels. The x axis for the eigenvalue plots denotes the eigenvalues sorted from largest to smallest real part. System with exponents $\{0.7, 0.2, -0.1, -0.4, -1.6\}$ and time step equal to 1 for simplicity.

correctly recover the dominant exponential. For details of this experiment and further results and figures see Appendix A.

Figure 4 shows the effect of various parameters such as the noise, the system order, and the lag vector length on the shape of the filter. We see that adding noise decreases the number of the phases as expected [Fig. 4(a)]. Similarly as we decrease the system order k, the number of phases decrease [Fig. 4(b)] as expected from the biorthogonality argument given at the end of Sec. II. Figure 4(c) shows that with a finite amount of noise, as the length of the lag vector is increased, the number of phases can slowly increase. This is due to the fact that with longer lag vectors, there is higher noise tolerance. Indeed as we take the noise values to zero, we see that the lag vector length no longer affects the number of phases in the filter. Increasing the length of the lag vector merely stretches the filter shape (Fig. 6 in Appendix B).

C. Comparison with optimal projection under noisy observations

To verify that in the presence of noise our algorithm provides a sensible approximation of the dominant dynamical mode, in this section we discuss a noise-optimal projection alternative and provide empirical comparisons.

Let $\mathbf{x}_t$ be a time series and suppose our goal is to extract the dominant exponential mode by projecting $\mathbf{x}_t$ onto the filter given by $\mathbf{v}$ [derived as the top eigenvector of (8) when there is no observation noise]. We refer to $\mathbf{v}$ as the noiseless filter.

Suppose, however, we do not have access to $\mathbf{x}_t$, but only a noisy observation $\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t$, where $\mathbf{n}_t$ is isotropic Gaussian noise with variance α^2. What is the optimal projection of $\mathbf{y}_t$? Consider the optimization problem

$$
\min_{\mathbf{w}} E[|\mathbf{w}^\top \mathbf{y}_t - \mathbf{v}^\top \mathbf{x}_t|^2]
$$
$$
= \min_{\mathbf{w}} (\mathbf{w} - \mathbf{v})^\top \mathbf{C}_{xx}(\mathbf{w} - \mathbf{v}) + \alpha^2 \mathbf{w}^\top \mathbf{w},
$$

where we have used the fact that $\mathbf{n}_t$ is mean zero isotropic noise independent of $\mathbf{x}_t$. Solving for the optimal $\mathbf{w}$ we find

$$
\mathbf{w} = (\mathbf{C}_{xx} + \alpha^2 \mathbf{I})^{-1} \mathbf{C}_{xx} \mathbf{v}. \tag{11}
$$

In order to apply this method, we need to know the noise variance α and also the noise-free projector $\mathbf{v}$ and covariance $\mathbf{C}_{xx}$. Computing these noise-free quantities is challenging, especially in a problem where the dynamics might be changing over time. Furthermore, computing the full covariance matrix $\mathbf{C}_{xx}$ requires more samples than just computing the top subspace in our generalized eigenvalue problem [Eq. (8)]. In practice, we find that even if we know α and $\mathbf{v}$, our method gives comparable performance to this noise-optimal solution (Fig. 5). Furthermore, our method provides a better estimate of the most dominant mode than the nonadaptive method which uses the true (i.e., noiseless) eigenvector. In practice, in order to compute the noiseless filter $\mathbf{v}$, we set the noise variance $\alpha = 10^{-6}$.
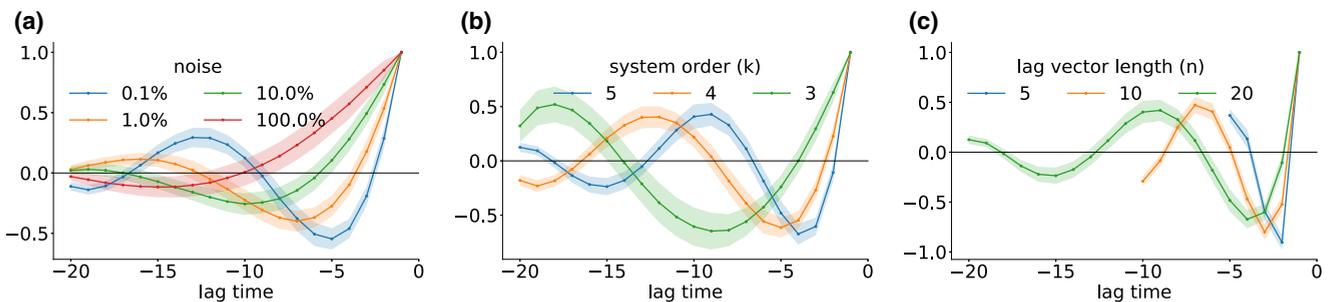
FIG. 4. Dependence of the filter shape for the rank-five system in Fig. 2 on system parameters: (a) noise amplitude, (b) system order (the number of exponential constituents, k), and (c) length of the lag vector, n. In (b) and (c) the noise is fixed at 0.01%. In (b), to achieve a comparable system of lower rank k, we keep the top k exponents of the system.
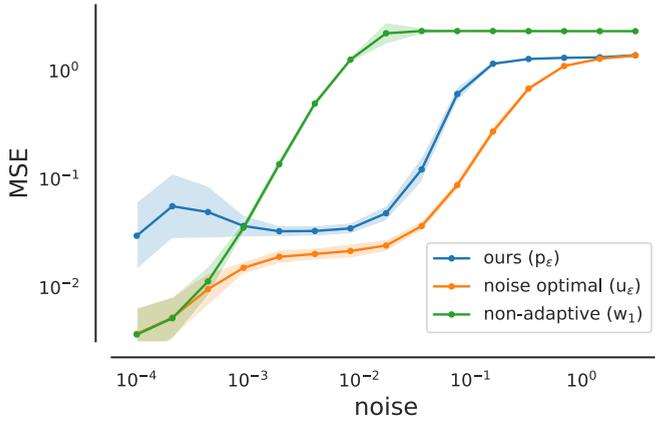
FIG. 5. Comparison of the performance with respect to the mean-square error between the recovered dominant exponential and the ground truth. The noise-optimal solution is given by the filter in Eq. (11) and noiseless filter refers to using the filter computed in the absence of noise but applied to noisy observations. The system is the same as in Fig. 2.

IV. DISCUSSION

We propose to model a neuron on an algorithmic level as performing a NMD on its input. Specifically, the temporal filter of a neuron is given by the top left eigenvector of the generalized eigenproblem formulated in terms of covariances of lag vectors. The neuron outputs the fastest growing mode present in the input thus predicting a future trend.

How could a biological neuron find the top eigenvector of the generative matrix? One possibility is for a neuron to solve the generalized eigenproblem (8) using a power iteration. Specifically, this would require two types of history-dependent active conductances (ion channels), which encode input covariances, $\mathbf{X}_+\mathbf{X}^\top$ and $\mathbf{X}\mathbf{X}^\top$, with opposite signs (depolarizing and hyperpolarizing). Then the solution of (8) would be given by the voltage that balances the two currents. The output of a neuron would reflect such voltage, thus projecting the input on the top left eigenvector of the transition matrix.

Whereas a single neuron computing a scalar signal can represent only one eigenmode, multiple neurons may extract multiple modes (not just the top eigenmode). For each neuron to represent a different mode their output must be decorre-
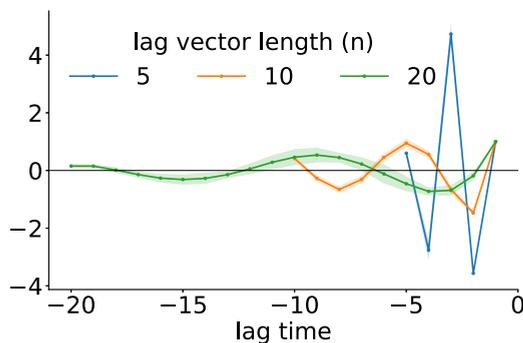
FIG. 6. The dependence of the filter shape vs the length of the lag vector at low noise values (0.00001%).

lated, for example, by the lateral inhibitory connections. The strengths of such connections can be adjusted using biologically plausible local learning rules as has been shown by some of the authors previously in the context of extracting eigenmodes of the multichannel covariance [20–22]. Also, we proposed a framework for deriving multichannel neural networks for solving symmetric generalized eigenvalue problems [23] and the approach can potentially be extended to solve nonsymmetric generalized eigenvalue problems. We anticipate that a similar approach can be applied to extracting eigenmodes of the time-lagged dynamics using local learning rules.

Because the matrix $\mathbf{A}$ in Eq. (3) is generically non-normal, its eigenvectors are not guaranteed to be orthogonal. In practice, this means that they are not robust to perturbations to the elements of the matrix $\mathbf{A}$. This requires particular care to ensure that the eigenvalues are sufficiently distinct as determined by the pseudospectrum of $\mathbf{A}$ [24].

Neuronal projection of its input onto the subspace corresponding to the fastest growing mode has an interpretation in terms of the phase portrait of the generative dynamical system. Close to a hyperbolic fixed point the dynamics are linear and the fastest growing mode would correspond to the unstable invariant subspace, which evolves into an attracting manifold away from the fixed point. Therefore, the sign of the projection onto the unstable subspace predicts along which unstable manifold the future trajectory of the system will develop beyond the linear regime. If the neuronal output is rectified, it effectively assigns the trajectory to a particular future state. Output of a layer of such rectifying neurons becomes a latent vector variable that can in turn be an input to the next layer. This opens a path to stacking the layers of such neurons to discover more and more abstract latent variables.

APPENDIX A: DETAILS OF ANALYTIC ARGUMENTS

Starting with Eq. (10) describing the effect of noise:

$$\mathbf{A} = (\mathbf{X}_+\mathbf{X}^\top)(\mathbf{X}\mathbf{X}^\top)^{-1} = \left(\mathbf{A}_0\mathbf{X}_0\mathbf{X}_0^\top + \epsilon^2\mathbf{S}\right)\left(\mathbf{X}_0\mathbf{X}_0^\top + \epsilon^2\mathbf{I}_k\right)^{-1}.$$
$$(\text{A1})$$

we can derive the noisy version of the dynamics if we know the covariance structure of the non-noisy data $\mathbf{X}_0\mathbf{X}_0^\top$. It is intuitively understandable why this structure is necessary since the effect of noise depends not just on the eigenvalues and eigenvectors of $\mathbf{A}$, it also depends on the magnitudes of each individual mode. Specifically, if we assume the generative model in Eq. (9), it is not just the a_i that determine the noisy dynamics, the c_i are also important and this information is not present in $\mathbf{A}$ but is available in $\mathbf{X}_0\mathbf{X}_0^\top$.

For plots given in Fig. 3, we computed the $\mathbf{X}_0\mathbf{X}_0^\top$ using time series of the form in Eq. (9). We take the length of the time series to be just long enough to compute a full rank covariance matrix.

APPENDIX B: OTHER FIGURES

Figure 6 shows the dependence of the filter shape as we vary the lag vector length for a small amount of noise. We
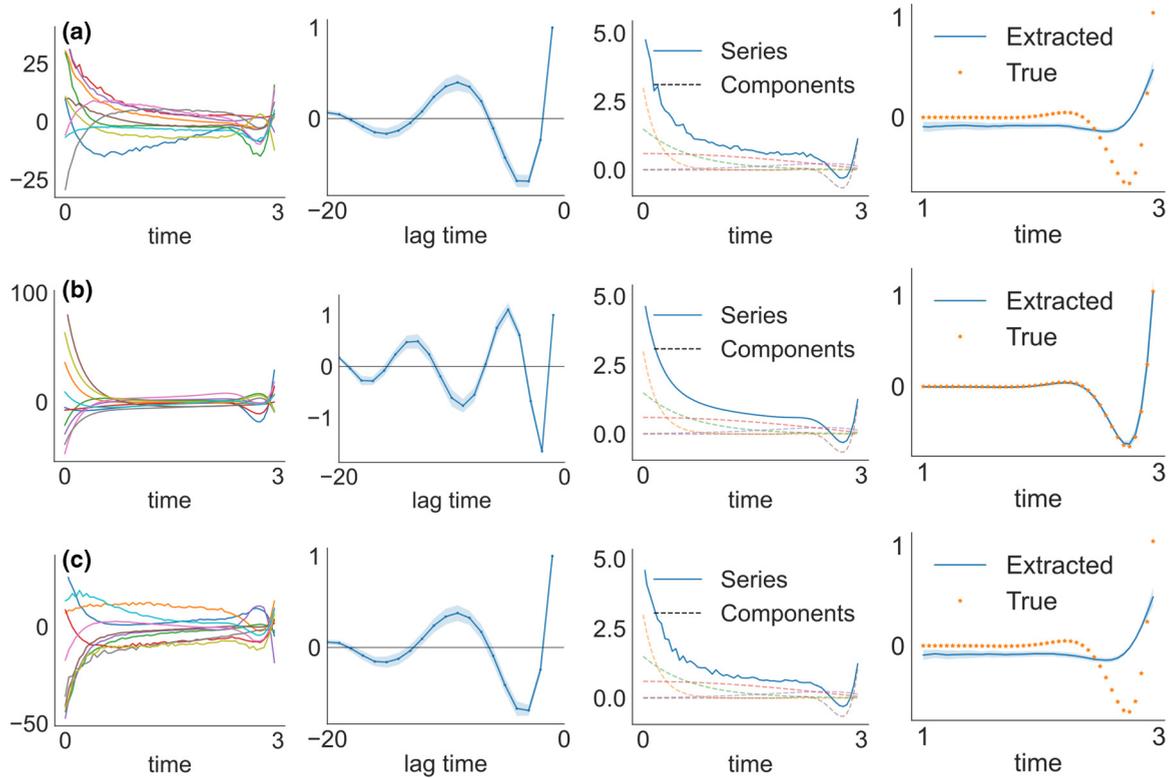
FIG. 7. We reproduce Fig. 2 with imaginary parts in the exponents. (a) Imaginary parts in all trajectories except the one with the largest eigenvalue. (b) Imaginary parts in all trajectories including the one with the largest eigenvalue. (c) Imaginary parts in all trajectories including the one with the largest eigenvalue, but lower noise level.

see that increasing the lag vector length no longer affects the number of phases [cf. Fig. 4(c)].

APPENDIX C: INVERSE OF A VANDERMONDE MATRIX

Recall that in the noiseless case (i.e., $\epsilon = 0$), the matrix $\mathbf{A}$ has eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{V}$ is a Vandermonde matrix. In this section we show that the top left eigenvector of a Vandermonde matrix of rank n changes sign $n - 1$ times. Consider the Vandermonde matrix

$$\mathbf{V} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

where $\lambda_1 > \cdots > \lambda_n > 0$. Rawashdeh [25] showed that the inverse of the Vandermonde matrix V is given by

$$(\mathbf{V}^{-1})_{ij} = (-1)^{i+j} S_{n-j,i} \frac{\prod_{k<l \text{ s.t.} l, k \neq j}^n (\lambda_k - \lambda_l)}{\prod_{k<l}^n (\lambda_k - \lambda_l)},$$

where

$$S_{j,k} := \sum_{1 \leqslant i_1 < \cdots < i_k \leqslant n, i_\ell \neq j} \prod_{m=1}^k \lambda_{i_m} > 0.$$

Importantly, the sign of the $(1, j)$th entry is positive (negative) if j is odd (even). Therefore, the top left eigenvector changes sign $n - 1$ times.

APPENDIX D: IMAGINARY PARTS IN EIGENVALUES

In the main text we only examine a time series composed of real exponents. Here we examine the case where exponents have imaginary parts, which creates oscillatory behavior in the training and test time series. We reproduce Fig. 2 where we infer the $\mathbf{A}$ matrix from training trajectories, extract the top left eigenvector, and use it to extract the dominant mode from an unseen time series. In Fig. 7(a) we add an imaginary part in all trajectories except the one with the largest eigenvalue. We see that we are still able to accurately extract the largest mode of the time series. In Fig. 7(b) we add an imaginary part in all trajectories including the one with the largest eigenvalue. Here we fail to accurately reconstruct the largest mode, although we get the qualitative behavior correct. In Fig. 7(c) we repeat the test in Fig. 7(b) but in addition we lower the noise level. Under these circumstances we see that we are able to accurately reconstruct the largest mode, even when it contains an imaginary part. The specific parameters are given in Table I.

TABLE I. Experimental parameters for Fig. 7.

Row	Exponent Imaginary Part	noise
(a)	$\{-5 + 0.2i, -1.5 + 7i, 1i, 1.5 + 0.1i, 5.001\}$	0.05
(b)	$\{-5 + 0.4i, -1.5 + 0.2i, 0.5i, 1.5 + 0.5i, 5.001 + 6i\}$	0.05
(c)	$\{-5 + 0.4i, -1.5 + 0.2i, 0.5i, 1.5 + 0.5i, 5.001 + 6i\}$	0.0001

[1] Y. Singer, Y. Teramoto, B. D. B. Willmore, J. W. H. Schnupp, A. J. King, and N. S. Harper, Sensory cortex is optimized for prediction of future input, eLife **7**, e31557 (2018).

[2] M. Chalk, O. Marre, and G. Tkavcik, Toward a unified theory of efficient, predictive, and sparse coding, Proc. Natl. Acad. Sci. **115**, 186 (2018).

[3] M. V. Srinivasan, S. B. Laughlin, and A. Dubs, Predictive coding: A fresh view of inhibition in the retina, Proc. R. Soc. London B **216**, 427 (1982).

[4] R. P. N. Rao and D. H. Ballard, Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects, Nature Neurosci. **2**, 79 (1999).

[5] Y. Huang and R. P. N. Rao, Predictive coding, WIRES Cogn. Sci. **2**, 580 (2011).

[6] K. Gregor and D. Chklovskii, A lattice filter model of the visual pathway, Adv. Neural Info. Proc. Syst. **25** (2012).

[7] N. Tishby, F. C. Pereira, and W. Bialek, The information Bottleneck method, in *Proceedings of the 37th Allerton Conference on Communication, Control, & Computing* (University of Illinois, 1999).

[8] W. Bialek, R. R. De Ruyter Van Steveninck, and N. Tishby, Efficient representation as a design principle for neural coding and computation, in *2006 IEEE International Symposium on Information Theory* (IEEE, New York, 2006), pp. 659–663.

[9] N. Y. Jun, G. D. Field, and J. M. Pearson, Efficient coding, channel capacity and the emergence of retinal mosaics, Adv. Neural Info. Proc. Syst. **35**, 33211 (2022).

[10] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Texts in Applied Mathematics (Springer, New York, 2006).

[11] L. D. Landau and E. M. Lifshitz, *Mechanics*, 3rd ed., Course of Theoretical Physics (Butterworth-Heinemann, London, 1976), Vol. 1.

[12] S. A. Baccus and M. Meister, Fast and slow contrast adaptation in retinal circuitry, Neuron **36**, 909 (2002).

[13] J. K. Liu and T. Gollisch, Spike-triggered covariance analysis reveals phenomenological diversity of contrast adaptation in the retina, PLoS Comput. Biol. **11**, e1004425 (2015).

[14] D. Chander and E. J. Chichilnisky, Adaptation to temporal contrast in primate and salamander retina, J. Neurosci. **21**, 9904 (2001).

[15] K. J. Kim and F. Rieke, Temporal contrast adaptation in the input and output signals of salamander retinal ganglion cells, J. Neurosci. **21**, 287 (2001).

[16] Z. F. Mainen and T. J. Sejnowski, Reliability of spike timing in neocortical neurons, Science **268**, 1503 (1995).

[17] R. Bellman, *Introduction to Matrix Analysis* (SIAM, Philadelphia, 1997).

[18] S. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition, Exp. Fluids **57**, 42 (2016).

[19] Y. Ohmichi, Y. Sugioka, and K. Nakakita, Stable dynamic mode decomposition algorithm for noisy pressure-sensitive-paint measurement data, AIAA J. **60**, 1965 (2022).

[20] C. Pehlevan, T. Hu, and D. B. Chklovskii, A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data, Neural Comput. **27**, 1461 (2015).

[21] C. Pehlevan, A. M. Sengupta, and D. B. Chklovskii, Why do similarity matching objectives lead to hebbian/anti-hebbian networks? Neural Comput. **30**, 84 (2018).

[22] C. Pehlevan and D. B. Chklovskii, Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks, IEEE Signal Proc. Mag. **36**, 88 (2019).

[23] D. Lipshutz, Y. Bahroun, S. Golkar, A. M. Sengupta, and D. B. Chklovskii, Normative framework for deriving neural networks with multicompartmental neurons and non-hebbian plasticity, PRX Life **1**, 013008 (2023).

[24] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators* (Princeton University Press, Princeton, 2020).

[25] E. A. Rawashdeh, A simple method for finding the inverse matrix of Vandermonde matrix, Matematicki Vesnik **3**, 207 (2019).