PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements

Daniel Durstewitz*

Dept. of Theoretical Neuroscience, Bernstein Center for Computational Neuroscience Heidelberg-Mannheim, Central Institute of Mental Health, Medical Faculty Mannheim/ Heidelberg University, Mannheim, Germany

* daniel.durstewitz@zi-mannheim.de

OPEN ACCESS

Citation: Durstewitz D (2017) A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. PLoS Comput Biol 13 (6): e1005542. https://doi.org/10.1371/journal.pcbi.1005542

Editor: Matthias Bethge, University of Tübingen and Max Planck Institute for Biologial Cybernetics, GERMANY

Received: November 2, 2016

Accepted: April 26, 2017

Published: June 2, 2017

Data Availability Statement: All Matlab code for PLRNN estimation, running major examples, and the parameter files and experimental data is freely available at https://github.com/durstdby/PLRNNstsp.

Funding: This work was funded through two grants from the German Research Foundation (DFG, Du 354/8-1, and within the Collaborative Research Center 1134, D01) to the author, and by the German Ministry for Education and Research

Abstract

The computational and cognitive properties of neural systems are often thought to be implemented in terms of their (stochastic) network dynamics. Hence, recovering the system dynamics from experimentally observed neuronal time series, like multiple single-unit recordings or neuroimaging data, is an important step toward understanding its computations. Ideally, one would not only seek a (lower-dimensional) state space representation of the dynamics, but would wish to have access to its statistical properties and their generative equations for in-depth analysis. Recurrent neural networks (RNNs) are a computationally powerful and dynamically universal formal framework which has been extensively studied from both the computational and the dynamical systems perspective. Here we develop a semi-analytical maximum-likelihood estimation scheme for piecewise-linear RNNs (PLRNNs) within the statistical framework of state space models, which accounts for noise in both the underlying latent dynamics and the observation process. The Expectation-Maximization algorithm is used to infer the latent state distribution, through a global Laplace approximation, and the PLRNN parameters iteratively. After validating the procedure on toy examples, and using inference through particle filters for comparison, the approach is applied to multiple single-unit recordings from the rodent anterior cingulate cortex (ACC) obtained during performance of a classical working memory task, delayed alternation. Models estimated from kernel-smoothed spike time data were able to capture the essential computational dynamics underlying task performance, including stimulus-selective delay activity. The estimated models were rarely multi-stable, however, but rather were tuned to exhibit slow dynamics in the vicinity of a bifurcation point. In summary, the present work advances a semi-analytical (thus reasonably fast) maximum-likelihood estimation framework for PLRNNs that may enable to recover relevant aspects of the nonlinear dynamics underlying observed neuronal time series, and directly link these to computational properties.

(BMBF, 01ZX1314G, SP10) within the e:Med program. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The author has declared that no competing interests exist.

Author summary

Neuronal dynamics mediate between the physiological and anatomical properties of a neural system and the computations it performs, in fact may be seen as the 'computational language' of the brain. It is therefore of great interest to recover from experimentally recorded time series, like multiple single-unit or neuroimaging data, the underlying stochastic network dynamics and, ideally, even equations governing their statistical evolution. This is not at all a trivial enterprise, however, since neural systems are very high-dimensional, come with considerable levels of intrinsic (process) noise, are usually only partially observable, and these observations may be further corrupted by noise from measurement and preprocessing steps. The present article embeds piecewise-linear recurrent neural networks (PLRNNs) within a state space approach, a statistical estimation framework that deals with both process and observation noise. PLRNNs are computationally and dynamically powerful nonlinear systems. Their statistically principled estimation from multivariate neuronal time series thus may provide access to some essential features of the neuronal dynamics, like attractor states, generative equations, and their computational implications. The approach is exemplified on multiple single-unit recordings from the rat prefrontal cortex during working memory.

Introduction

Stochastic neural dynamics mediate between the underlying biophysical and physiological properties of a neural system and its computational and cognitive properties (e.g. [1–4]). Hence, from a computational perspective, we are often interested in recovering the neural network dynamics of a given brain region or neural system from experimental measurements. Yet, experimentally, we commonly have access only to noisy recordings from a relatively small proportion of neurons (compared to the size of the brain area of interest), or to lumped surface signals like local field potentials or the EEG. Inferring from these the computationally relevant dynamics is therefore not trivial, especially since both the recorded signals (e.g., spike sorting errors; [5]) as well as the neural system dynamics itself (e.g., stochastic synaptic release; [6]) come with a good deal of noise. The stochastic nature of neural dynamics has, in fact, been deemed crucial for perceptual inference and decision making [7–9], and potentially helps to avoid local minima in task learning or problem solving [10].

Speaking in statistical terms, 'model-free' techniques which combine delay embedding methods with nonlinear basis expansions and kernel techniques have been one approach to the problem [11; 12]. These techniques provide informative lower-dimensional visualizations of population trajectories and (local) approximations to the neural flow field, but they may highlight only certain, salient aspects of the dynamics (but see [13]) and, in any case, do not directly return distribution generating equations or underlying computations. Alternatively, state space models, a statistical framework particularly popular in engineering and ecology (e.g. [14]), have been adapted to extract lower-dimensional, probabilistic neural trajectory flows from higher-dimensional recordings [15–25]. State space models link a process model of the unobserved (latent) underlying dynamics to the experimentally observed time series via observation equations, and differentiate between stochasticity in the process and observation noise (e.g. [26]). So far, with few exceptions (e.g. [23; 27]), these models assumed linear latent dynamics, however. Although this may often be sufficient to yield lower-dimensional smoothed trajectories, it implies that the recovered dynamical model may be less apt for capturing highly nonlinear dynamical phenomena in the observations, and will by itself not be

powerful enough to reproduce a range of important dynamical and computational processes in the nervous system, among them multi-stability which has been proposed to underlie neural activity during working memory [28–32], limit cycles (stable oscillations), or chaos (e.g. [33]).

Here we derive a new state space algorithm based on piecewise-linear (PL) recurrent neural networks (RNN). It has been shown that RNNs with nonlinear activation functions can, in principle, approximate any dynamical system's trajectory or, in fact, dynamical system itself (given some general conditions; [34–36]). Thus, in theory, they are powerful enough to recover whatever dynamical system is underlying the experimentally observed time series. Piecewise linear activation functions, in particular, are by now the most popular choice in deep learning algorithms [37–39], and considerably simplify some of the derivations within the state space framework (as shown later). They may also be more apt for producing working memory-type activity with longer delays if for some units the transfer function happens to coincide with the bisectrix (cf. [40]), and ease the analysis of fixed points and stability. We then apply this newly derived algorithm to multiple single-unit recordings from the rat prefrontal cortex obtained during a classical delayed alternation working memory task [41].

Results

State space model

This article considers simple discrete-time piecewise-linear (PL) recurrent neural networks (RNN) of the form

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{W}\max\{\mathbf{0}, \mathbf{z}_{t-1} - \boldsymbol{\theta}\} + \mathbf{C}\mathbf{s}_t + \boldsymbol{\varepsilon}_t \quad , \boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}), \tag{1}$$

where $\mathbf{z}_t = (z_{1t}\ldots z_{Mt})^T$ is the (M×1)-dimensional (latent) neural state vector at time $t = 1\ldots T$, $\mathbf{A} = diag([a_{11}\ldots a_{MM}])$ is an M×M diagonal matrix of auto-regression weights, $\mathbf{W} = (0\ w_{12}\ldots w_{1M},\ w_{21}\ 0\ w_{23}\ldots w_{2M},\ w_{31}\ w_{32}\ 0\ w_{34}\ldots w_{3M},\ldots)$ is an M×M *off*-diagonal matrix of connection weights, $\boldsymbol{\theta} = (\theta_1\ldots\theta_M)^T$ is a set of (constant) activation thresholds, $\mathbf{s}_t$ is a sequence of (known) external K-dimensional inputs, weighted by (M×K) matrix $\mathbf{C}$, and $\boldsymbol{\varepsilon}_t$ denotes a Gaussian white noise process with diagonal covariance matrix $\boldsymbol{\Sigma} = diag([\sigma_{11}^2 \ldots \sigma_{MM}^2])$. The max-operator is assumed to work element-wise.

In physiological terms, latent variables z_{mt} are often interpreted as a membrane potential (or current) which gives rise to spiking activity as soon as firing threshold θ_m is exceeded (e.g. [42,43]). According to this interpretation, the diagonal elements in $\mathbf{A}$ may be seen as the neurons' individual membrane time constants, while the off-diagonal elements in $\mathbf{W}$ represent the between-neuron synaptic connections which multiply with the presynaptic firing rates. In statistical terms, (1) has the form of an auto-regressive model with a nonlinear basis expansion in variables z_{mt} (e.g. [44;45]), which retains linearity in parameters $\mathbf{W}$ for ease of estimation. Restricting model parameters, e.g. $\boldsymbol{\Sigma}$, to be of diagonal form, is common in such models to avoid over-specification and help identifiabiliy (e.g. [26; 46]; see also further below). For instance, including a diagonal in $\mathbf{W}$ would be partly redundant to parameters $\mathbf{A}$ (strictly so in a pure linear model). For similar reasons, and for ease of presentation, in the following we will focus on a model for which $K = M$ and $\mathbf{C} = \mathbf{I}$ (i.e., no separate scaling of the inputs), although the full model as stated above, Eq 1, was implemented as well (and code for it is provided; of course, the case $K>M$ could always be accommodated by pre-multiplying $\mathbf{s}_t$ by some *predefined* matrix $\mathbf{C}$, obtained e.g. by PCA on the input space). While different model formulations are around in the computational neuroscience and machine learning literature, often they may be related by a simple transformation of variables (see [47]) and, as long as the model is powerful enough to express the whole spectrum of basic dynamical phenomena, details of model specification may also not be overly crucial for the present purposes.

A particular advantage of the PLRNN model is that all its fixed points can be obtained easily analytically by solving (in the absence of external input) the 2^M linear equations

$$\mathbf{z}_* = (\mathbf{A} + \mathbf{W}_\Omega - \mathbf{I})^{-1}\mathbf{W}_\Omega\boldsymbol{\theta}, \tag{2}$$

where Ω is to denote the set of indices of units for which we assume $z_m \leq \theta_m$, and $\mathbf{W}_\Omega$ the respective connectivity matrix in which all columns from $\mathbf{W}$ corresponding to units in Ω are set to 0. Obviously, to make $\mathbf{z}_*$ a true fixed point of (1), the solution to (2) has to be consistent with the defined set Ω, that is $z_{*m} \leq \theta_m$ has to hold for all $m \in \Omega$ and $z_{*m} > \theta_m$ for all $m \notin \Omega$. For networks of moderate size (say $M<30$) it is thus computationally feasible to explicitly check for all fixed points and their stability.

For estimation from experimental data, latent state model (1) is then connected to some N-dimensional observed vector time series $\mathbf{X} = \{\mathbf{x}_t\}$ via a simple linear-Gaussian model,

$$\mathbf{x}_t = \mathbf{B}\phi(\mathbf{z}_t) + \boldsymbol{\eta}_t \quad , \boldsymbol{\eta}_t \sim N(\mathbf{0}, \boldsymbol{\Gamma}), \tag{3}$$

where $\phi(\mathbf{z}_t) \coloneqq \max\{\mathbf{0}, \mathbf{z}_t - \boldsymbol{\theta}\}$, $\{\boldsymbol{\eta}_t\}$ is the (white Gaussian) observation noise series with diagonal covariance matrix $\boldsymbol{\Gamma} = diag([\gamma_{11}^2 \ldots \gamma_{NN}^2])$, and $\mathbf{B}$ an $N \times M$ matrix of regression weights. Thus, the idea is that only the PL-transformed activation $\phi(\mathbf{z}_t)$ reaches the 'observation surface' as, e.g., with spiking activity when the underlying membrane dynamics itself is not visible. We further assume for the initial state,

$$\mathbf{z}_1 \sim N(\boldsymbol{\mu}_0 + \mathbf{s}_1, \boldsymbol{\Sigma}), \tag{4}$$

which has, for simplicity, the same covariance matrix as the process noise in general (reducing the number of to be estimated parameters). In the case of multiple, temporally separated trials, we allow each one to have its own individual initial condition $\boldsymbol{\mu}_k$, $k = 1 \ldots K$.

The general goal here is to determine both the model's unknown parameters $\Xi = \{\boldsymbol{\mu}_0, \mathbf{A}, \mathbf{W}, \boldsymbol{\Sigma}, \mathbf{B}, \boldsymbol{\Gamma}\}$ (assuming fixed thresholds $\boldsymbol{\theta}$ for now) as well as the unobserved, latent state path $\mathbf{Z} \coloneqq \{\mathbf{z}_t\}$ (and its second-order moments) from the experimentally observed time series $\{\mathbf{x}_t\}$. These could be, for instance, properly transformed multivariate spike time series or neuroimaging data. This is accomplished here by the Expectation-Maximization (EM) algorithm which iterates state (E) and parameter (M) estimation steps and is developed in detail for model (1) and (3) in the Methods. In the following I will first discuss state and parameter estimation separately for the PLRNN, before describing results from the full EM algorithm in subsequent sections. This will be done along two toy problems, a higher-order nonlinear oscillation (stable limit cycle), and a simple 'working memory' paradigm in which one of two discrete stimuli had to be retained across a temporal interval. Finally, the application of the validated PLRNN EM algorithm will be demonstrated on multiple single-unit recordings obtained from rats on a standard working memory task (delayed alternation; data from [41], kindly provided by Dr. James Hyman, University of Nevada, Las Vegas).

State estimation

The latent state distribution, as explained in Methods, is a high-dimensional (piecewise) Gaussian mixture with the number of components growing as $2^{T \times M}$ with sequence length T and number of latent states M. Here a semi-analytical, approximate approach was developed that treats state estimation as a combinatorial problem by first searching for the mode of the full distribution (cf. [16; 48]; in contrast, e.g., to a recursive filtering-smoothing scheme that makes local (linear-Gaussian) approximations, e.g. [15; 26]). This approach amounts to solving a high ($2^{M \times T}$)-dimensional piecewise linear problem (due to the piecewise quadratic, in the states $\mathbf{Z}$, log-likelihood Eqs 6 and 7). Here this was accomplished by alternating between (1)

solving the linear set of equations implied by a given set of linear constraints $\Omega := \{(m,t)|z_{mt} \leq \theta_m\}$ (cf. Eq 7 in Methods) and (2) flipping the sign of the constraints violated by the current solution $\mathbf{z}_*(\Omega)$ to the linear equations, thus following a path through the (M×T)-dimensional binary space of linear constraints using Newton-type iterations (similar as in [49], see Methods; note that here the 'constraints' are not fixed as in quadratic programming problems). Given the mode and state covariance matrix (evaluated at the mode from the negative inverse Hessian), all other expectations needed for the EM algorithm were then derived analytically, with one exception that was approximated (see Methods for full details).

The toy problems introduced above were used to assess the quality of these approximations. For the first toy problem, an order-15 limit cycle was produced with a PLRNN consisting of three recurrently coupled units, inputs to units #1 and #2, and parameter settings as indicated in Fig 1 and provided Matlab file 'PLRNNoscParam'. The limit cycle was repeated for 50 full cycles (giving 750 data points) and corrupted by process noise (cf. Fig 1). These noisy states (arranged in a (3 x 750) matrix $\mathbf{Z}$) were then transformed into a (3 x 750) output matrix $\mathbf{X}$, to which observation noise was added, through a randomly drawn (3 x 3) regression weight matrix $\mathbf{B}$. State estimation was started from a random initial condition. True (but noise-corrupted) and estimated states for this particular problem are illustrated in Fig 1A, indicating a tight fit (although some fraction of the linear constraints were still violated, ≈0.27% in the present example and <2.3% in the working memory example below; see Methods on this issue).

To examine more systematically the quality of the approximate-analytical estimates of the first and second order moments of the joint distribution across states z and their piecewise linear transformations $\phi(z)$, samples from $p(\mathbf{Z}|\mathbf{X})$ were simulated using bootstrap particle filtering (see Methods). Although these simulated samples are based only on the filtering (not the smoothing) steps (and (re-)sampling schemes may have issues of their own; e.g. [26], analytical and sampling estimates were in tight agreement, correlating almost to 1 for this example, as shown in Fig 2.

Fig 3A illustrates the setup of the 'two-cue working memory task', chosen for later comparability with the experimental setup. A 5-unit PLRNN was first trained by conventional gradient descent ('real-time recurrent learning' (RTRL), see [50; 51]) to produce a series of six 1's on unit #3 and six 0's on unit #4 five time steps after an input (of 1) occurred on unit #1, and the reverse pattern (six 0's on unit #3 and six 1's on unit #4) five time steps after an input occurred on unit #2. A stable PLRNN with a reasonable solution to this problem was then chosen for further testing the present algorithm (cf. Fig 3C). (While the RTRL approach was chosen to derive a working memory circuit with reasonably 'realistic' characteristics like a wider distribution of weights, it is noted that a multi-stable network is relatively straightforward to construct explicitly given the analytical accessibility of fixed points (see Methods); for instance, choosing $\boldsymbol{\theta}$ = (0.5 0.5 0.5 0.5 2), $\mathbf{A}$ = (0.9 0.9 0.9 0.9 0.5), and $\mathbf{W}$ = ($0\ \omega\ -\omega\ -\omega\ -\omega$, $\omega\ 0\ -\omega\ -\omega\ -\omega$, $-\omega\ -\omega\ 0\ \omega\ -\omega$, $-\omega\ -\omega\ \omega\ 0\ -\omega$, 11110) with ω = 0.2, yields a tri-stable system.) Like for the limit cycle problem before, the number of observations was taken to be equal to the number of latent states, and process and observation noise were added (see Fig 4 and Matlab file 'PLRNNwmParam' for specification of parameters). The system was simulated for 20 repetitions of each trial type (i.e., cue-1 or cue-2 presentations) with different noise realizations and each 'trial' started from its own initial condition $\boldsymbol{\mu}_k$ (see Methods), resulting in a total series length of T = 20×2×20 = 800 (although, importantly, in this case the time series consisted of distinct, temporally segregated trials, instead of one continuous series, and was treated as such an ensemble of series by the algorithm). As before, state estimation started from random initial conditions and was provided with the correct parameters, as well as with the observation matrix $\mathbf{X}$. While Fig 3B illustrates the correlation between true (i.e., simulated) and estimated

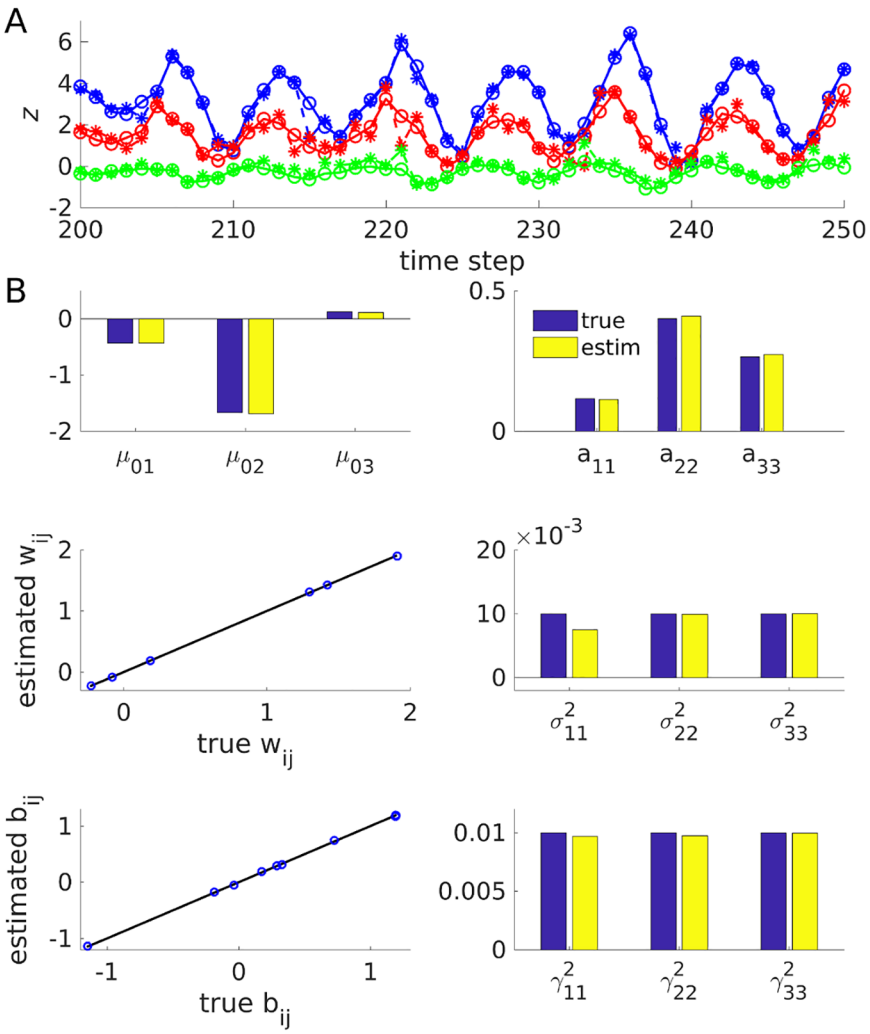

Fig 1. State and parameter estimates for nonlinear cycle example. (A) True (solid/ open-circle lines) and estimated (dashed-star lines) states over some periods of the simulated limit cycle generated by a 3-state PLRNN when true parameters were provided (for this example, $\boldsymbol{\theta} \approx (0.86, 0.09, -0.85)$; all other parameters as in B, see also provided Matlab file 'PLRNNoscParam.mat'). 'True states' refers to the actual states from which the observations $\mathbf{X}$ were generated. Inputs of $s_{it} = 1$ were provided to units $i = 1$ and $i = 2$ on time steps 1 and 10 of each cycle, respectively. Note that true and inferred states are tightly overlapping in this low-noise example (such that the 'stars' appear on top of the 'open circles'). (B) True and estimated model parameters for (from top-left to bottom-right) $\boldsymbol{\mu}_0, \mathbf{A}, \mathbf{W}, \boldsymbol{\Sigma}, \mathbf{B}, \boldsymbol{\Gamma}$, when true states (but not their higher-order moments) were provided. Bisectrix lines (black) indicate identity.

https://doi.org/10.1371/journal.pcbi.1005542.g001

states across all trials and units, Fig 3C shows true and estimated states for a representative cue-1 (left) and cue-2 (right) trial, respectively. Again, our procedure for obtaining (or approximating) the maximum a-posteriori (MAP) estimate of the state distribution appears to work quite well (in general, only locally optimal or approximate solutions may be achieved, however, and the algorithm may have to be repeated with different state initializations; see Methods).

Parameter estimation

Given the true states, how well would the algorithm retrieve the parameters of the PLRNN? To assess this, the actual model states (which generated the observations $\mathbf{X}$) from simulation runs

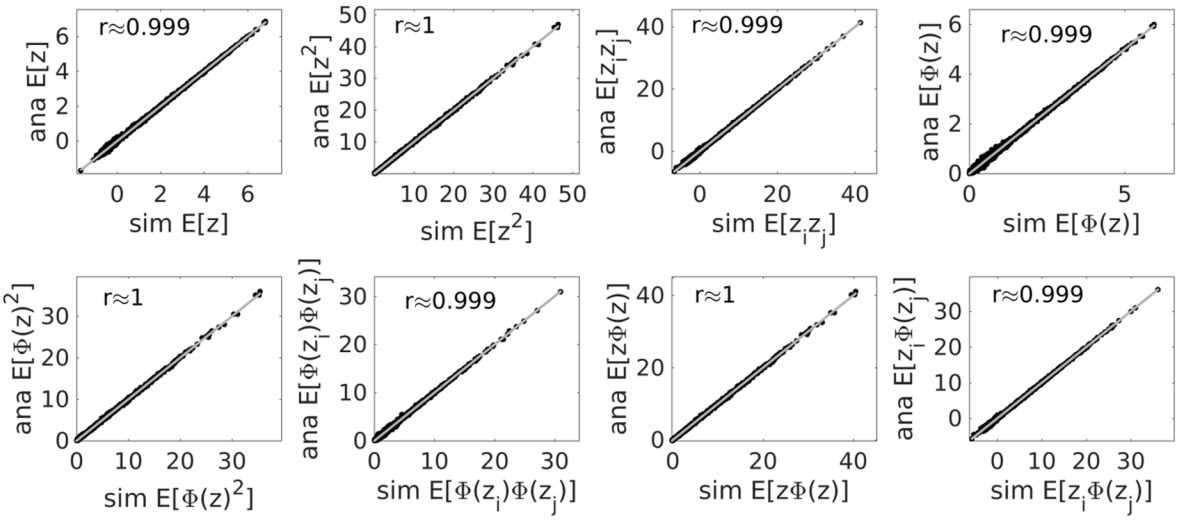

Fig 2. Agreement between simulated (*x*-axes) and semi-analytical (*y*-axes) solutions for state expectancies for the model from Fig 1 across all three state variables and $T = 750$ time steps. Here, $\phi(z_i) \coloneqq \max\{0, z_i - \theta_i\}$ is the PL activation function. Simulated state paths and their moments were generated using a bootstrap particle filter with 10^4 particles. Bisectrix lines in gray indicate identity.

https://doi.org/10.1371/journal.pcbi.1005542.g002

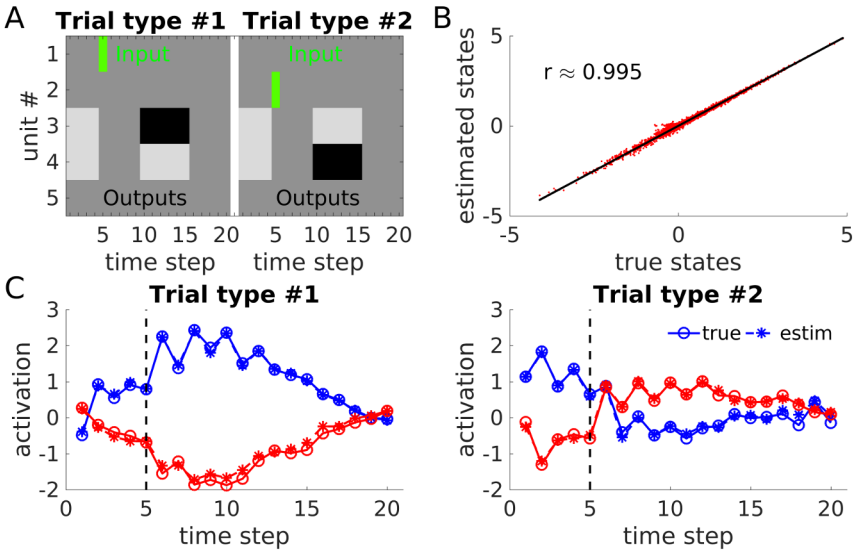

Fig 3. State estimation for 'working memory' example when true parameters were provided. (A) Setup of the simulated working memory task: Stimulus inputs (green bars, $s_{it} = 1$, and 0 otherwise) and requested outputs (black = 1, light-gray = 0, dark-grey = no output required) across the 20 time points of a working memory trial (with two different trial types) for the 5 PLRNN units. (B) Correlation between estimated and true states (i.e., those from which the observations **X** were generated) across all five state variables and $T = 800$ time steps. Bisectrix in black. (C) True (open-circle/ solid lines) and estimated (star-dashed lines) states for output units #3 (blue) and #4 (red) when $s_{15} = 1$ (left) or $s_{25} = 1$ (right) for single example trials. Note that true and inferred states are tightly overlapping in this low-noise example (such that the 'stars' often appear on top of the 'open circles'). Although working memory PLRNNs may, in principle, be explicitly designed (see text), here a 5-state PLRNN was first trained by conventional gradient descent (real-time recurrent-learning [50]) to perform the task in A, to yield more 'natural' and less uniform ground truth states and parameters. Here, all $\theta_i = 0$ (implying that there can only be one stable fixed point). See Matlab file 'PLRNNwmParam.mat' and Fig 4 for details on parameters.

https://doi.org/10.1371/journal.pcbi.1005542.g003

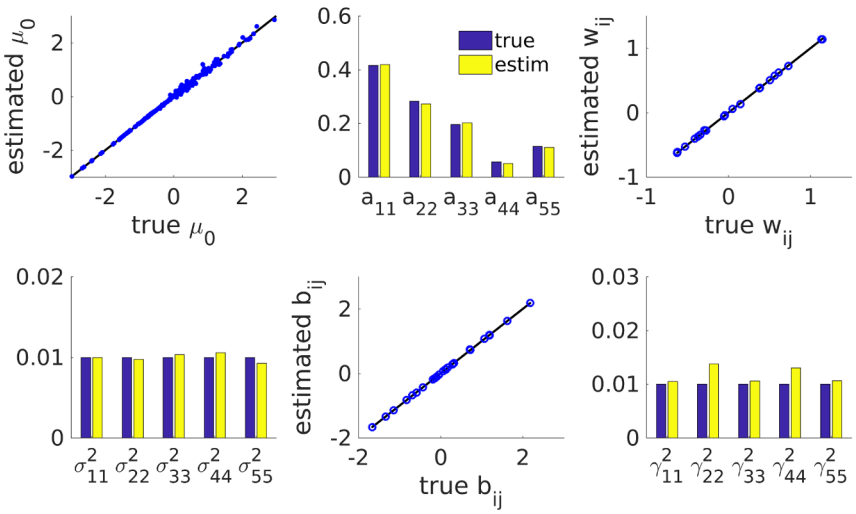

Fig 4. True and estimated parameters for the working memory PLRNN (cf. Fig 3) when true states were provided. From top-left to bottom-right, estimates for: $\boldsymbol{\mu}_0$,**A**,**W**,$\boldsymbol{\Sigma}$,**B**,$\boldsymbol{\Gamma}$. Note that most parameter estimates were highly accurate, although all state covariance matrices still had to be estimated as well (i.e., with the true states provided as initialization for the E-step). Bisectrix lines in black indicate identity.

https://doi.org/10.1371/journal.pcbi.1005542.g004

of the oscillation and the working memory task described above were provided as initialization for the E-step. Based on these, the algorithm first estimated the state covariances for z and $\phi(z)$ (see above), and then the parameters in a second step (i.e., the M-step). Note that the parameters can all be computed analytically given the state distribution (see Methods), and, provided the state covariance matrices (summed across time) as required in Eq 17A, 17D and 17F are non-singular, have a unique solution. Hence, in this case, any misalignment with the true model parameters can only come from one of two sources: i) estimation was based on one finite-length noisy realization of the PLRNN process, ii) all *second order moments* of the state distribution were still *estimated* based on the true state vectors. However, as can be appreciated from Fig 1B (oscillation) and Fig 4 (working memory), for the two (relatively low-noise) example scenarios studied here, all parameter estimates still agreed tightly with those describing the true underlying model.

In the more general case where *both* the states and the parameters are unknown and only the observations are given, note that the model as stated in Eqs 1 & 3 is over-specified as, for instance, at the level of the observations, additional variance placed into $\boldsymbol{\Sigma}$ may be compensated for by adjusting $\boldsymbol{\Gamma}$ accordingly, and by rescaling **W** and, within limits, **A** (cf. [52; 53]). In the following we therefore always arbitrarily fixed $\boldsymbol{\Sigma}$ (to some scalar; see Methods), as common in many latent variable models (like factor analysis), including state space models (e.g. [27; 46]). It may be worth noting here that the relative size of $\boldsymbol{\Sigma}$ vs. $\boldsymbol{\Gamma}$ determines how much weight is put on temporal consistency among states ("$\boldsymbol{\Sigma}<\boldsymbol{\Gamma}$") vs. fitting of the observations ("$\boldsymbol{\Sigma}>\boldsymbol{\Gamma}$") within the likelihood, Eq 5.

Joint estimation of states and parameters by EM

The observations above confirm that our algorithm finds satisfactory approximations to the underlying state path and state covariances when started with the right parameters, and—vice versa—identifies the correct parameters when provided with the true states. Indeed, the M-step, since it is exact, can only increase the expected log-likelihood Eq 5 with the present state expectancies fixed. However, due to the system's piecewise-defined discrete nature, modifying

the parameters may lead to a new set of constraint violations, that is may throw the system into a completely different linear subspace which may imply a decrease in the likelihood in the E-step. It is thus not guaranteed that a straightforward EM algorithm converges (cf. [54; 55]), or that the likelihood would even monotonically increase with each EM iteration.

To examine this issue, full EM estimation of the WM model (as specified in Fig 4, using $N = 20$ outputs in this case) was run 240 times, starting from different random, uniformly distributed initializations for the parameters. Fig 5B ($\Delta t = 0$) gives, for the five highest likelihood solutions across all 240 runs (Fig 5A), the mean squared error (MSE) $\text{avg}[(x_{it} - \hat{x}_{it})^2]$ between actual neural observations x_{it} and model predictions $\hat{x}_{it}$, which is close to 0 (and, correspondingly, correlations between predicted and actual observations were close to 1). With respect to the inferred states, note that estimated and true model states may not be in the same order, as any permutation of the latent state indices together with the respective columns of observation matrix $\mathbf{B}$ will be equally consistent with the data $\mathbf{X}$ (see also [27]). For the WM model examined here, however, partial order information is implicitly provided to the EM algorithm through the definition of unit-specific inputs s_{it}. For the present example, true and estimated states for the highest likelihood solution were nicely linearly correlated for all 5 latent variables (Fig 6), but some of the regression slopes significantly differed from 1, indicating a degree of freedom in the scaling of the states. Note that if the system were strictly linear, the states would be identifiable only up to a linear transformation in general, since any multiplication of the latent states by some matrix $\mathbf{V}$ could essentially be reversed at the level of the outputs by back-multiplying $\mathbf{B}$ with $\mathbf{V}^{-1}$ (cf. [27]). Likewise, in the present *piecewise linear* system, one may expect that there is a class of piecewise-linear transformations of the states which is still compatible with the observed outputs, and hence that the model is only identifiable up to this class of transformations (a general issue with state space models, of course, not particular to the present one; cf. [53]). However, this might not be a too serious issue, if one is primarily interested in the latent dynamics (rather than in the exact parameters).

Fig 7 illustrates the distribution of initial and final parameter estimates around their true values across all 240 runs (before and after reordering the estimated latent states based on the rotation that would be required for achieving the optimal mapping onto the true states, as determined through Procrustes analysis). Fig 7 reveals that a) the EM algorithm does clearly improve the estimates and b) these final estimates seemed to be relatively 'unbiased' (i.e., with deviations centered around 0).

Computational complexity of state inference and EM algorithm

How do the computational costs of the algorithm grow as the number of latent variables in the model is increased? As pointed out in Paninski et al. [16], exploiting the block-tridiagonal nature of the covariance matrices, the numerical operations within one iteration of the state inference algorithm (i.e., solving $\partial Q^*_{\boldsymbol{\Omega}}(\mathbf{Z})/\partial \mathbf{Z} = 0$, Eq 7) can be done in linear, $O(M \times T)$, time, just like with the Kalman filter (due to the model's Markov properties, full inversion of the Hessian is also not necessary to obtain the relevant moments of the posterior state distribution). This leaves open the question of how many more mode search iterations, i.e. linear equation solving (Eq 7) and constraint-flipping (vector $\mathbf{d}_{\boldsymbol{\Omega}}$) steps, are required as the number of latent variables (through either M or T) increases. The answer is provided in Fig 8A which is based on the experimental data set discussed below. Although a full computational complexity analysis is beyond the scope of this paper, at least for these example data (and similar to what has sometimes been reported for the somewhat related Simplex algorithm; [56]), the increase with M appears to be at most linear. Likewise, the *total* number of iterations within the full EM procedure, i.e. the number of mode-search steps summed across *all* EM iterations (thus

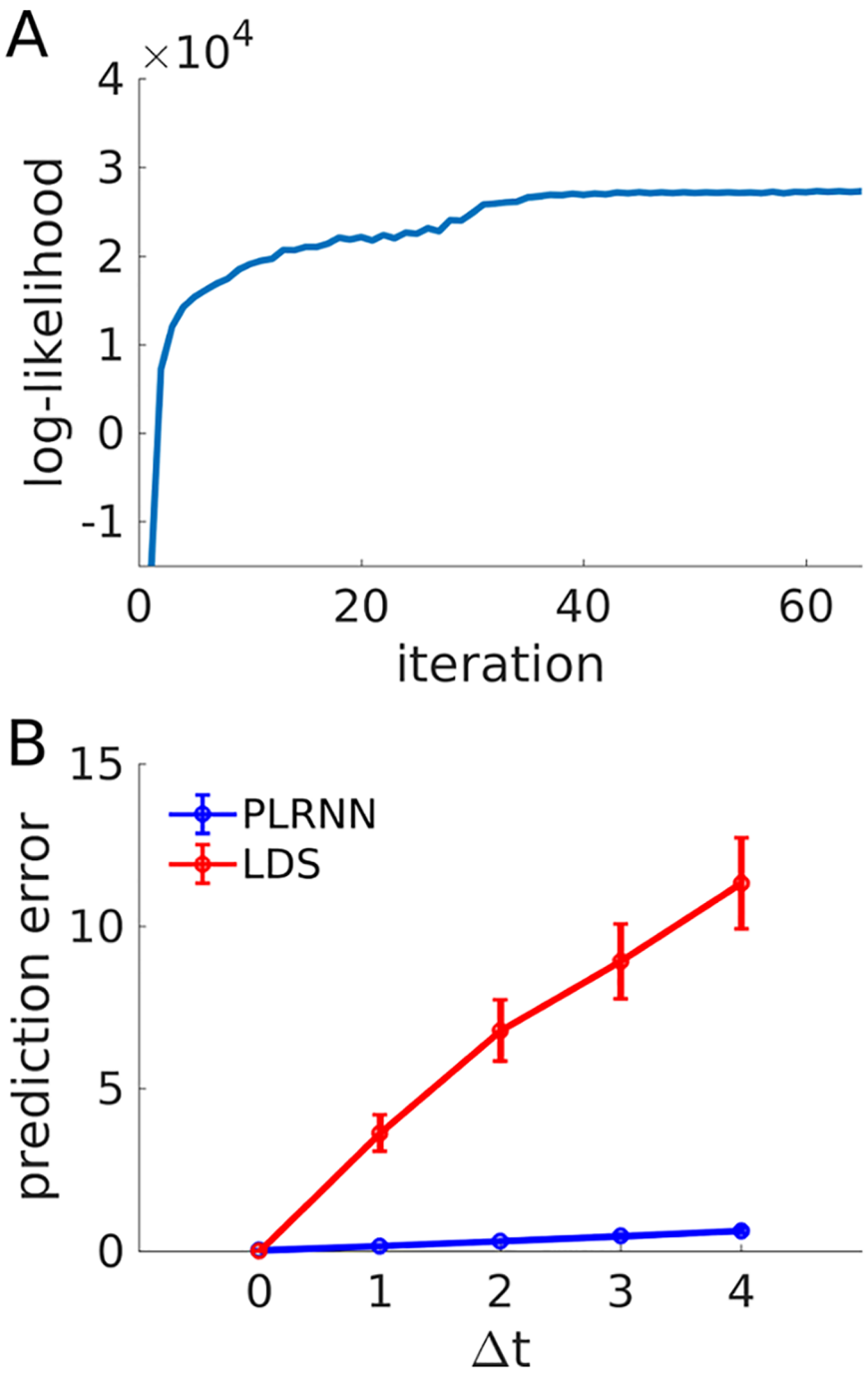

Fig 5. Performance of full EM algorithm on working memory model. (A) Log-likelihood as a function of EM iteration for the highest-likelihood run out of all 240 initializations. As in this example, the log-likelihood, although generally increasing, was not always monotonic (note the little ripples; see discussion in Results). (B) Mean squared prediction error, $\mathrm{avg}[(x_{it} - \hat{x}_{it})^2]$, between true ($\{\mathbf{x}_t\}$) and predicted ($\{\widehat{\mathbf{x}}_t\}$) observations across all 20 output variables and the 5 highest-likelihood solutions, as a function of ahead-prediction time step Δt, for the original PLRNN (blue curve) and for a linear dynamical system (LDS; red curve) estimated via EM from the same, PLRNN-generated data. Note that while the true and estimated observations agree almost perfectly for both the PLRNN and LDS if predicted directly from the inferred states (i.e., $\widehat{\mathbf{x}}_t = \mathbf{B}\phi(\hat{\mathbf{z}}_t)$), prediction quality severely decays for the LDS while remaining high for the PLRNN if $\{\widehat{\mathbf{x}}_t\}$-predictions were made from states forecast Δt time steps into the future (see text for further explanation; note that a slight decay in prediction quality across Δt is inevitable because of the process noise). Error bars = SEM.

https://doi.org/10.1371/journal.pcbi.1005542.g005

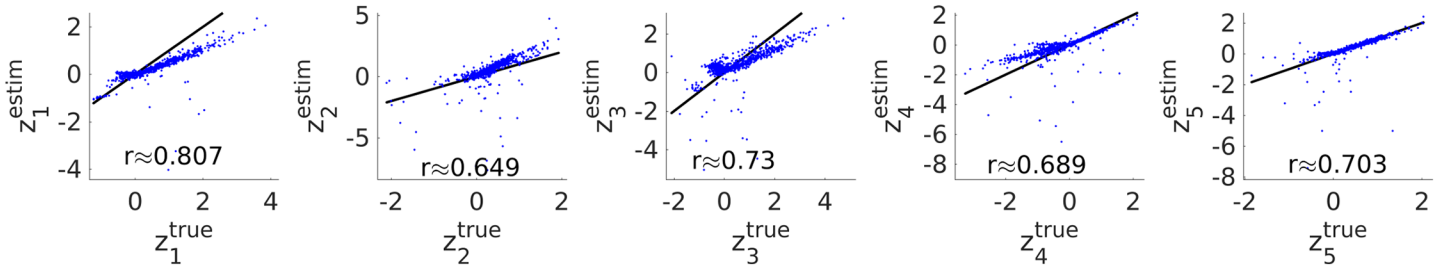

Fig 6. State estimates for ML solution (cf. Fig 5) from the full EM algorithm on the working memory model. In this example, true and estimated states were nicely linearly related, although mostly with regression slopes deviating from 1 (see text for further discussion). State estimation in this case was performed by inverting only the single constraint corresponding to the largest deviation on each iteration (see Methods). Bisectrix lines in black indicate identity.

https://doi.org/10.1371/journal.pcbi.1005542.g006

reflecting the overall scaling of the full algorithm), was about linear (Fig 8B; in this case, single-constraint instead of complete flipping (see Methods) was used which, of course, increases the overall number of iterations but may perform more stably; note that in general the absolute number of iterations will also depend on detailed parameter settings of the algorithm, like the EM convergence criterion and error tolerance). Thus, overall, the present state inference algorithm seems to behave quite favorably, with an at most linear increase in the number of iterations required as the number of latent variables is ramped up.

Application to experimental recordings

I next was interested in what kind of structure the present PLRNN approach would retrieve from experimental multiple ($N = 19$) single-unit recordings obtained while rats were performing

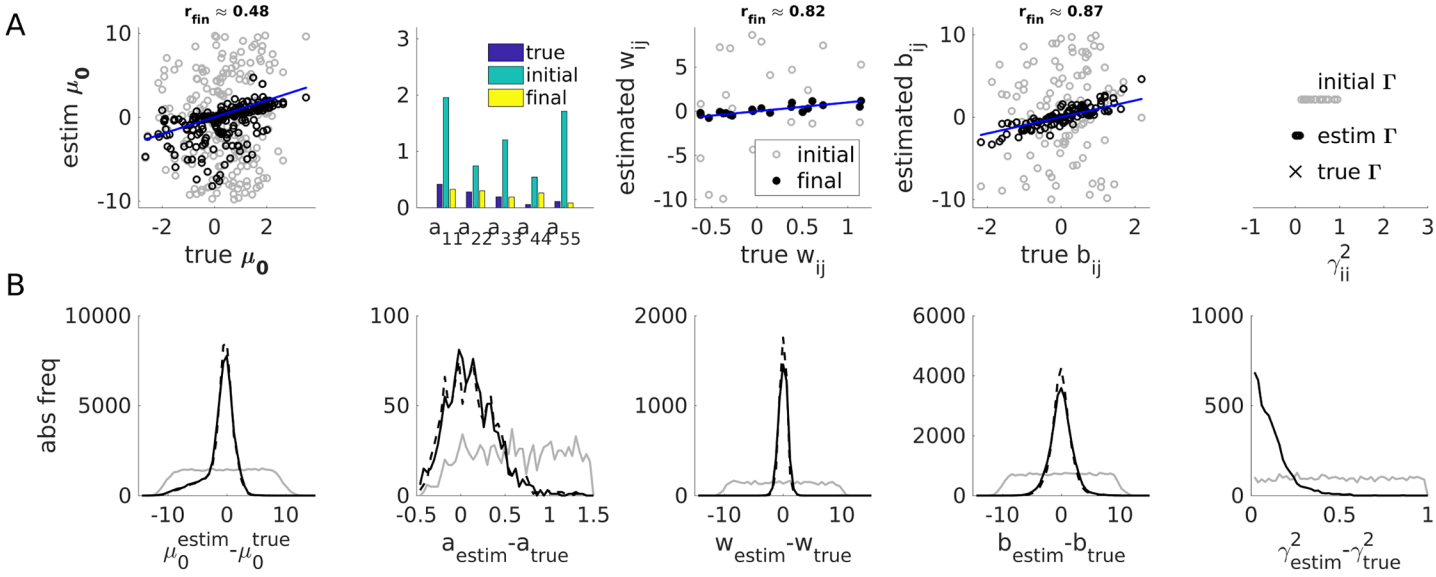

Fig 7. Full EM algorithm on working memory model. (A) Parameter estimates for ML solution from Fig 5. True parameters (on *x*-axes or as blue bars, respectively), initial (gray circles or green bars) and final (black circles or yellow bars) parameter estimates for (from left to right) $\boldsymbol{\mu}_0$,$\mathbf{A}$,$\mathbf{W}$,$\mathbf{B}$,$\boldsymbol{\Gamma}$. Bisectrix lines in blue. Correlations between true and final estimates are indicated on top (note from Eq 17C that the estimates for $\boldsymbol{\mu}_0$ are based on just one state, hence will naturally be less precise). (B) Distributions of initial (gray curves), final (black-solid curves), and final after reordering of states (black-dashed curves), deviations between estimated and true parameters across all 240 EM runs from different initial conditions. All final distributions were approximately centered around 0, indicating that final parameter estimates were relatively unbiased. Note that partial information about state assignments was implicitly provided to the network through the unit-specific inputs (and, more generally, may also come from the unit-specific thresholds θ_i, although these were all set to 0 for the present example), and hence state reordering only produced slight improvements in the parameter estimates.

https://doi.org/10.1371/journal.pcbi.1005542.g007

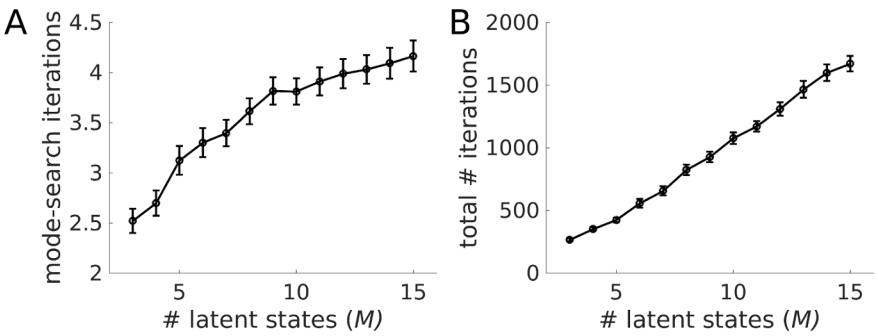

Fig 8. Computational performance of state inference (E-step) and full EM algorithm as the number of latent states is increased. (A) The number of full mode-search iterations, i.e. the number of constraint-sets **Ω** visited as defined through constraint vector **d** (cf. Eq 7) within one E-step, increases (sub-)linearly with the number *M* of latent states included in the model. (B) Likewise, the *total* number of mode-search steps (evaluated with *single-constraint* flipping here) summed across all EM iterations increases about linearly with *M* (single-constraint flipping requires about 10-fold more iterations than full-constraint flipping, but was observed to perform more stably). Note that this measure combines the number of EM iterations with the number of mode-search steps during each EM pass, and in this sense reflects the scaling of the full EM procedure. Performance tests shown were run on the experimental data sets illustrated in Figs 9–12. Means were obtained across 40 different initial conditions (with each, in turn, representing the mean from 3x14 = 42 runs in A, or 14 runs in B, separately for each of 14 trials). Error bars = SEM (across initial conditions).

https://doi.org/10.1371/journal.pcbi.1005542.g008

a simple and well-examined working memory task, namely spatial delayed alternation [41] (see Methods). (Note that in the present context this analysis is mainly meant as an exemplification of the current model approach, not as a detailed examination of the working memory issue itself.) The delay was always initiated by a nose poke of the animal into a port located on the side opposite from the response levers, and had a minimum length of 10 s. Spike trains were first

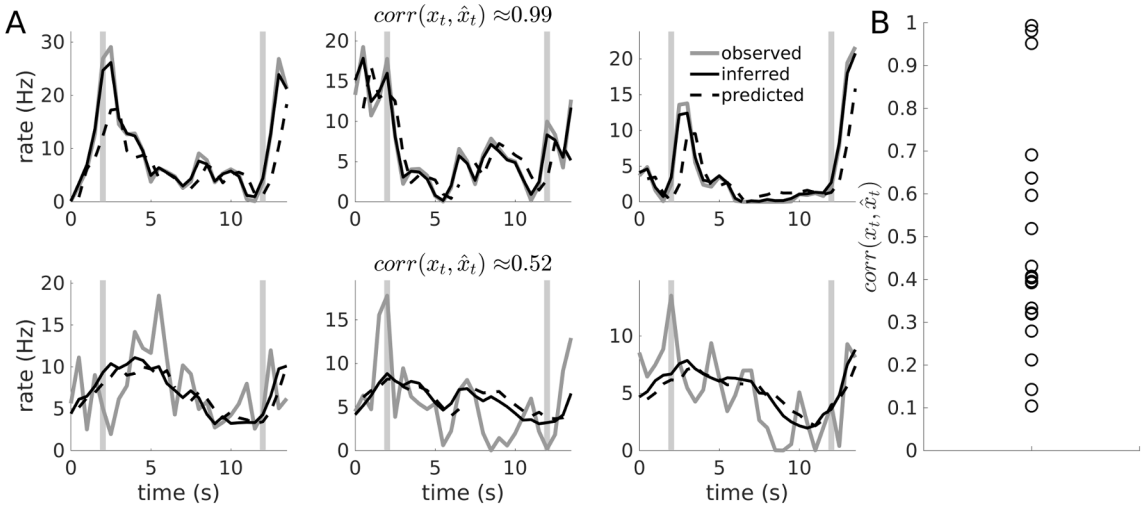

Fig 9. Prediction of single unit responses. (A) Top row: Example of an ACC unit (darker-gray curves) captured very well by the estimated PLRNN despite considerable trial to trial fluctuations (3 consecutive trials shown). Both model estimates from the directly inferred states (black curves) and from 1-step-ahead predictions of states $\mathbf{z}_t$ (dashed curves) are shown. Bottom row: Example of another ACC unit on the same three trials where only the average trend was captured by the PLRNN when firing rates were estimated from either the directly inferred or predicted states. Gray vertical bars in all panels indicate times of cue/ response. State estimation in this case was performed by inverting only the single constraint corresponding to the largest deviation on each iteration (see Methods). (B) Correlations among actual ($\{\mathbf{x}_t\}$) and predicted ($\{\tilde{\mathbf{x}}_t\}$) observations for all 19 neurons within this data set.

https://doi.org/10.1371/journal.pcbi.1005542.g009

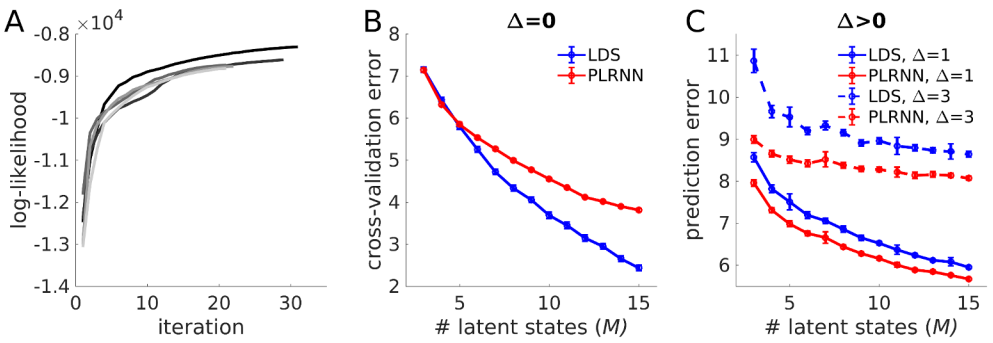

Fig 10. Log-likelihood of PLRNN and cross-validation performance of linear (LDS) and nonlinear (PLRNN) state space models on the ACC data. (A) Examples of log-likelihood curves across EM iterations from the 5/36 highest-likelihood runs for a 5-state PLRNN estimated from 19 simultaneously recorded prefrontal neurons on a working memory task (cf. Fig 9). State estimation here was performed by inverting only the single constraint corresponding to the largest deviation on each iteration (see Methods). (B) Cross-validation error (CVE) for the PLRNN (red curve) and the LDS (blue curve) as a function of the number of latent states *M*. CVE was assessed on each of 14 left-out trials with model parameters estimated from the remaining 14–1 = 13 experimental trials. Shown are squared errors $(x_{it} - \hat{x}_{it})^2$ averaged across all units *i*, time points *t*, and 40 different initial conditions. (C) Same as A, but with outputs $\hat{x}_{it}$ estimated from states predicted $\Delta t = 1$ (solid curves) or $\Delta t = 3$ (dashed curves) time steps ahead. Note that in this case the PLRNN consistently performs better than a LDS for all *M*, with the PLRNN-LDS difference growing as Δt increases. Error bars represent SEMs across those of the 40 initial conditions for which stable models were obtained (same for the means).

https://doi.org/10.1371/journal.pcbi.1005542.g010

transformed into kernel density estimates by convolution with a Gaussian kernel (see Methods), as done previously (e.g. [12; 57; 58]), and binned with 500 ms resolution. This also renders the observed data more suitable to the Gaussian noise assumptions of the present observation model, Eq 3. Models with different numbers of latent states were estimated, with $M = 5$ or $M = 10$ chosen for the examples below. Periods of cue presentation were indicated to the model by setting external inputs $s_{it} = 1$ to units $i = 1$ (left lever) or $i = 2$ (right lever) for three 500 ms time bins surrounding the event (and $s_{it} = 0$ otherwise), and the response period was indicated by setting $s_{3t} = 1$ for 3 consecutive time bins irrespective of the correct response side (i.e., non-discriminatively). The EM algorithm was started from a range of different initializations of the parameters (including thresholds $\mathbf{\theta}$), and the 5 highest likelihood solutions were considered further for the examples below.

Fig 10A gives the log-likelihoods across EM iterations for these 5 highest-likelihood solutions (starting from 36 different initializations) for the $M = 5$ model. Interestingly, there were single neurons whose responses were predicted quite well by the estimated model despite large trial-to-trial fluctuations (Fig 9A, top row), while there were others with similar trial-to-trial fluctuations for which the model only captured the general trend (Fig 9A, bottom row; to put this into context, Fig 9B gives the full distribution of correlations between actual and predicted observations across all 19 neurons). This may potentially indicate that trial-to-trial fluctuations in single neurons could be for very different reasons: For instance, in those cases where strongly varying single unit responses are nevertheless tightly reproduced by the estimated model, a larger proportion of their trial-to-trial fluctuations may have been captured by the latent state dynamics, ultimately rooted in different (trial-unique) initializations of the states (recall that the states are not completely free to vary in accounting for the observations, but are constrained by the model's temporal consistency requirements). In contrast, when only the average trend is captured, the neuron's trial-to-trial fluctuations may be more likely to represent true intrinsic (or measurement) noise sources that the model's deterministic part cannot account for. In practice,

such conclusions would have to be examined more carefully to rule out that no other factors in the estimation procedure, like different local maxima, initializations, or over-fitting issues (see below), could account for these differences. Although this was not further investigated here, this observation nevertheless highlights the potential of (nonlinear) state space models to possibly provide new insights also into other long-standing issues in neurophysiology.

Cross-validation is an established means to address over-fitting [45], although due to the presence of both unknown parameters and unknown states, its application to state space models and its interpretation in this context may be a bit less straightforward. Here the cross-validation error was first assessed by leaving out each of the 14 experimental trials in turn, estimating model parameters Ξ from the remaining 13 trials, inferring states $\mathbf{z}_t$ given these parameters on the left-out trial, and computing the squared prediction errors $(x_{it} - \hat{x}_{it})^2$ between actual neural observations x_{it} and model predictions $\hat{x}_{it}$ on the left-out trial. As shown in Fig 10B, this measure steadily (albeit sub-linearly) decreases as the number M of latent states in the model is increased. At first sight, this seems to suggest that with $M = 5$ or even $M = 10$ the over-fitting regime is not yet reached. On the other hand, the latent states are, of course, not completely fixed by the transition equations, but have some freedom to vary as well (the true effective degrees of freedom for such systems are in fact very hard to determine, cf. [59]). Hence, we also examined the Δt-step-ahead prediction errors, that is, when the transition model were iterated Δt steps forward in time, and $\hat{x}_{i,t+\Delta t} = \mathbf{b}_{i\cdot}\phi(\hat{\mathbf{z}}_{t+\Delta t})$ estimated from the deterministically *predicted* states $\hat{\mathbf{z}}_{t+\Delta t} = \mathrm{H}^{\Delta t}(E[\mathbf{z}_t])$ (with $\mathrm{H}^{\Delta t}$ the Δt-times iterated map $\mathrm{H}(\mathbf{z}_t) = \mathbf{A}\mathbf{z}_t + \mathbf{W}\phi(\mathbf{z}_t) + \mathbf{C}\mathbf{s}_t$), not from the directly inferred states (that is, predictions were made on data points which were neither used to estimate parameters nor to infer the current state). These curves are shown for $\Delta t = 1$ and $\Delta t = 3$ in Fig 10C, and confirm that $M = 5$ might be a reasonable choice at which over-fitting has not yet ensued. (Alternatively, the predictive log-likelihood, $\log p(\mathbf{X}^{test}|\hat{\boldsymbol{\Xi}}^{train}) = \log \int p(\mathbf{X}^{test}|\hat{\mathbf{Z}})p(\hat{\mathbf{Z}}|\hat{\boldsymbol{\Xi}}^{train})d\hat{\mathbf{Z}}$, may be used for model selection (i.e., choice of M), with $p(\hat{\mathbf{Z}}|\hat{\boldsymbol{\Xi}}^{train})$ either approximated through the E-step algorithm (with all $\mathbf{X}$-dependent terms removed), or bootstrapped by generating $\hat{\mathbf{Z}}$-trajectories from the model with parameters $\hat{\boldsymbol{\Xi}}^{train}$ (note that this is different from particle filtering since $p(\hat{\mathbf{Z}}|\hat{\boldsymbol{\Xi}}^{train})$ does not depend on test observations $\mathbf{X}^{test}$). This is of course, however, computationally more costly to evaluate than the Δt-step-ahead prediction error.)

Fig 11 shows trial-averaged latent states for both left- and right-lever trials, illustrated in this case for one of the five highest likelihood solutions (starting from 100 different initializations) for the $M = 10$ model. Recall that the first 3 PLRNN units received external inputs to indicate left cue ($i = 1$), right cue ($i = 2$), or response ($i = 3$) periods, and so, not too surprisingly, reflect these features in their activation. On the other hand, the cue response is not very prominent in unit $i = 1$, indicating that activity in the driven units is not completely dominated by the external regressors either, while unit $i = 10$ (not externally driven) shows a clear left-cue response. Most importantly, many of the remaining state variables clearly distinguish between the left and right lever options throughout the delay period of the task, in this sense carrying a memory of the cue (previous response) within the delay. Some of the activation profiles appear to systematically climb or decay across the delay period, as reported previously (e.g. [1; 60]), but are a bit harder to read (at least in the absence of more detailed behavioral information), such that one may want to stick with the simpler $M = 5$ model discussed above. Either way, for this particular data set, the extracted latent states appear to summarize quite well the most salient computational features of this simple working memory task.

Further insight about the dynamical mechanisms of working memory might be gained by examining the system's fixed points and their eigenvalue spectrum. For this purpose, the EM algorithm was started from 400 different initial conditions (that is, initial parameter estimates

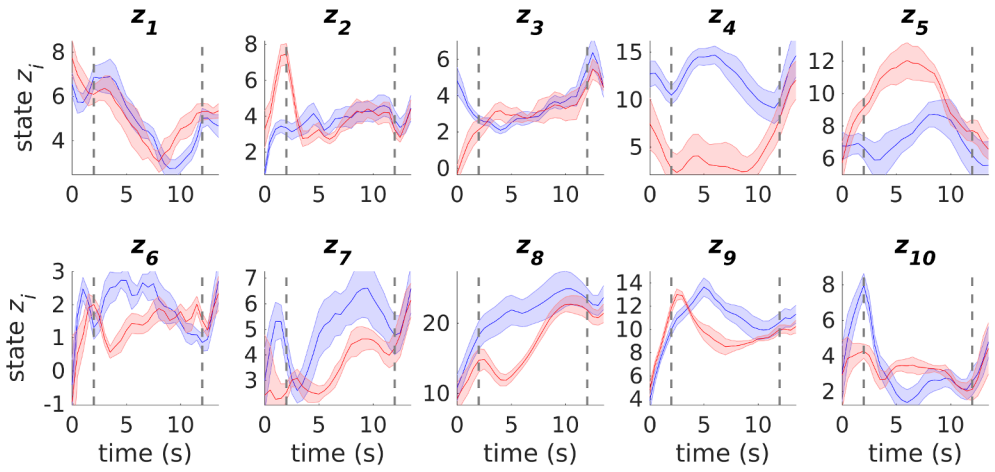

Fig 11. Example (from one of the 5 highest likelihood solutions) for latent states of a PLRNN with $M = 10$ estimated from ACC multiple single-unit recordings during working memory (cf. Figs 9 and 10). Shown are trial averages for left-lever (blue) and right-lever (red) trials with SEM-bands computed across trials. Dashed vertical lines flank the 10 s period of the delay phase used for model estimation. Note that latent variables z_4 and z_5, in particular, differentiate between left and right lever responses throughout most of the delay period.

https://doi.org/10.1371/journal.pcbi.1005542.g011

and threshold settings $\boldsymbol{\theta}$) with maximum absolute eigenvalues (of the corresponding fixed points) drawn from a relatively uniform distribution within the interval [0 3]. Although the estimation process rarely returned truly multi-stable solutions (just 2.5% of all cases), one frequently discussed candidate mechanism for working memory (e.g. [29; 32]), there was a clear trend for the final maximum absolute eigenvalues to aggregate around 1 (Fig 12). For the discrete-time dynamical system (1) this implies it is close to a bifurcation, with fixed points on the brink of becoming unstable, and will tend to produce (very) slow dynamics as the degree of convergence shrinks to zero along the maximum eigenvalue direction (strictly, a single eigenvalue near 1 does not yet guarantee a slow approach, but makes it very likely, especially in a (piecewise) linear system). Indeed, effectively slow dynamics is all that is needed to bridge the delays (see also [1]), while true multi-stability may perhaps even be the physiologically less likely scenario (e.g. [61; 62]). (Reducing the bin width from 500 ms to 100 ms appeared to produce solutions with eigenvalues even closer to 1 while retaining stimulus selectivity across the delay, but this observation was not followed up more systematically here).

Comparison to linear dynamical systems

Linear dynamical systems (LDS) have frequently and successfully been used to infer smooth neural trajectories from spike train recordings [15; 16; 20; 22] or other measurement modalities [63]. However, as noted before, they cannot, on their own, as a matter of principle, produce a variety of dynamical phenomena essential for neural computation and observed experimentally, including multi-stability (e.g. [29; 2]), limit cycles (*stable* oscillations; e.g. [3]), chaos (e.g. [33]), and many types of bifurcations and phase transitions. For instance, the question of whether working memory performance is better explained in terms of multi-stability or effectively slow dynamics (see above, Fig 12) is largely beyond the realm of an LDS, due to its inherent inability to express multi-stability in the first place. An LDS is therefore less suitable for retrieving system dynamics or computations in general.

Nevertheless, it may still be instructive to ask how much of the underlying dynamics could already be explained in linear terms. The most direct comparison of PLRNN to LDS

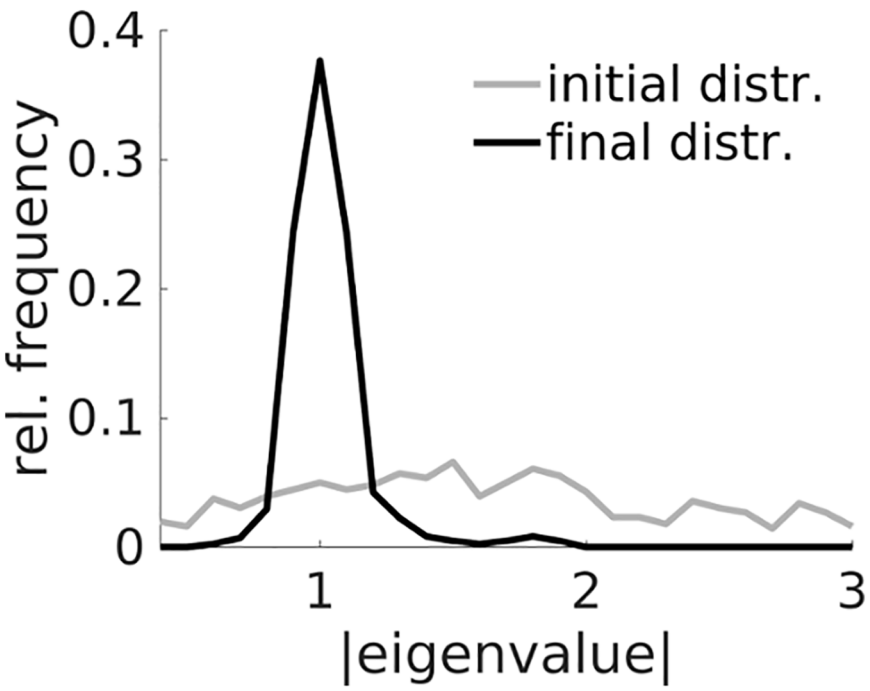

Fig 12. Initial (gray) and final (black) distributions of maximum (absolute) eigenvalues associated with all fixed points of 400 PLRNNs estimated from the experimental data (cf. Figs 9–11) with different initializations of parameters, including the (fixed) threshold parameters θ_i. Initial parameter configurations were deliberately chosen to yield a rather uniform distribution of absolute eigenvalues ≤ 3.

https://doi.org/10.1371/journal.pcbi.1005542.g012

performance is made by replacing the nonlinearity $\phi(\mathbf{z}_t) = \max\{\mathbf{0}, \mathbf{z}_t - \boldsymbol{\theta}\}$ in Eq 1 simply by the linear function $\phi(\mathbf{z}_t) = \mathbf{z}_t - \boldsymbol{\theta}$, yielding an LDS with exactly the same parameters Ξ as the PLRNN which can be subjected to the very same estimation and inference procedures (only that state inference can now be done exactly in just one step). Fig 10B reveals that a LDS fits the observed neural recordings about as well as the PLRNN for $M \leq 5$, and starts to excel PLRNN performance for $M > 5$. Since the major difference in this context is that the PLRNN places a tighter constraint on the temporal consistency of the states through the threshold-nonlinearity, it seems reasonable that this result is due to over-fitting, i.e. the LDS due to its smoothness allows for more freedom for the states to adjust to the actual observations (cf. [64]). It is important to bear in mind that consistency with the actual observations is just one objective of the maximum-likelihood formulation, Eq 5; the other is consistency of states across time according to the model specification.

Either way, the PLRNN starts to significantly outperform the LDS in terms of the Δt-step-ahead prediction errors (see above), with the gap in performance widening as Δt increases (Fig 10C). This strongly suggests that the PLRNN has internalized aspects of the system dynamics which the LDS fails to represent, i.e. supports the presence of nonlinear structure in the transition dynamics. Interestingly, looking back at Fig 5B, it turns out that even for simulated data generated by a PLRNN (at least for this example), for $\Delta t = 0$ an estimated LDS is about as good in reproducing the actual observations as an estimated PLRNN itself (with an MSE close to 0), that is, although, unlike the PLRNN it does not have the correct model structure. However, similar to what has been observed for the experimental data (Fig 10B and 10C), this performance rapidly drops and falls far behind that of the PLRNN (which remains low) as 1 or more time steps into the future are to be predicted (note that for the simulated model, unlike the

experimental example, the true number of states is known of course). This confirms that although the LDS may capture the actual observations quite well, it may not, unlike the PLRNN, be able to properly represent the underlying system within its internal dynamics.

As a note on the side, an LDS could be utilized to find proper, efficient initializations for the corresponding PLRNN, or to first improve initial estimates (although it remains to be examined whether this could potentially also bias the search space in an unfavorable way).

Discussion

Reconstructing computational dynamics from neuronal recordings

In the present work, a semi-analytical, maximum-likelihood (ML) approach for estimating piecewise-linear recurrent neural networks (PLRNN) from brain recordings was developed. The idea is that such models would provide 1) a representation of neural trajectories and computationally relevant dynamical features underlying high-dimensional experimental time series in a much lower-dimensional latent variable space (cf. [20; 25]) and 2) more direct access to the neural system's statistical and computational properties. Specifically, once estimated to reproduce the data (in the ML sense), such models may, in principle, allow for more detailed analysis and in depth insight into the system's probabilistic computational dynamics, e.g. through an analysis of fixed points and their linear stability (e.g. [28; 30; 32; 47; 65–70]), properties which are not directly accessible from the experimental time series.

Model-free (non-parametric) techniques, usually based on Takens' delay embedding theorem [71] and extensions thereof [72; 73], have also frequently been applied to gain insight into neuronal dynamics and its essential features, like attracting states associated with different task phases from in-vivo multiple single-unit recordings [11; 12] or unstable periodic orbits extracted from relatively low-noise slice recordings [74]. In neuroscience, however, one commonly deals with high-dimensional observations, as provided by current multiple single-unit or neuroimaging techniques (which still usually constitute just a minor subset of all the system's dynamical variables). In addition, there is a large variety of both process and measurement noise sources. Measurement noise may come from direct physical sources like, for instance, instabilities and movement in the tissue surrounding the recording electrodes, noise properties of the recording devices themselves, the mere fact that only a fraction of all system variables is experimentally accessed ('sampling noise'), or may result from preprocessing steps like spike sorting (e.g. [75; 76]). Process noise sources include thermal fluctuations and the probabilistic behavior of single ion channel gating [77], probabilistic synaptic release [6], fluctuations in neuromodulatory background and hormone levels, and a large variety of uncontrollable external noise sources via the sensory surfaces, including somatosensory and visceral feedback from within the body. In fact, the stochasticity of the neural dynamics itself has been deemed essential for a number of computational processes like those involved in decision making and inference [7–9]. This is therefore a quite different scenario from the comparatively low-dimensional and low-noise situations in, e.g., laser physics [78], and delay-embedding-based approaches to the reconstruction of neural dynamics may have to be augmented by machine learning techniques to retrieve at least some of its most salient features [11; 12].

Of course, model-based approaches like the one developed here are also plagued by the high dimensionality and high noise levels inherent in neural data, but perhaps to a lesser extent than approaches like delay embeddings that aim to directly construct the state space from the observations (see also [79]). This is because models as pursued in the statistical state space framework explicitly incorporate process and measurement noise assumptions into the system's description, performing smoothing in the latent space. Also, as long as the latent variable space itself is relatively small and related to the observations by simple linear equations, as

here, the high dimensionality of the observations themselves does not constitute a too serious issue for estimation. More importantly, however, it is of clear advantage to have access to process equations generating state distributions consistent with the observations, as this allows for a more in depth analysis of the system's stochastic dynamics and its relation to neural computation (e.g. [2; 28; 30; 47; 68; 70; 33]). There have also been various attempts to account for the observed dynamics directly in terms of nonlinear time series models (e.g. [13, 78, 80]), i.e. without reference to an underlying latent variable model, e.g. through differential equations expressed in terms of nonlinear basis expansions in the observations, estimated through strongly regularized (penalized) regression methods [11; 13; 80]. For neuroscientific data where usually only a small subset of all dimensions is observed, this implies that this approach has to be augmented by delay embedding techniques to replace the unobserved variables. This, in turn, may potentially lead to very high-dimensional systems (cf. [11,13]) that may necessitate further pre-processing steps to reduce the dimensionality again, in a way that preserves the dynamics. Also, there is no distinction between measurement and dynamical noise in these models, and, although functionally generic, the parameters of such models may be harder to interpret in a neuroscientific context. How these different assumptions and methodological steps affect the reconstruction of neural dynamics from high-dimensional, noisy neural time series, as compared to state space models, remains an open and interesting question at this point.

Comparison to other neural state space models

State space models are a popular statistical tool in many fields of science (e.g. [14; 63]), although their applications in neuroscience are of more recent origin [15, 16; 18; 19; 21–24]. The Dynamic Causal Modeling (DCM) framework advanced in the human fMRI literature to infer the functional connectivity of brain networks and their dependence on task conditions [63; 81] may be seen as a state space approach, although these models usually do not contain process noise (except for the more recently proposed 'stochastic DCM' [81]) and are commonly estimated through Bayesian inference, which imposes more constraints (via computational burden) on the complexity of the models that could potentially be dealt with in this framework. In neurophysiology, Smith & Brown [15] were among the first to suggest a state space model for multivariate spike count data by coupling a linear-Gaussian transition model with Poisson observations, with state estimation achieved by making locally Gaussian approximations to Eq 18. Similar models have variously been used subsequently to infer local circuit coding properties [18] or, e.g., biophysical parameters of neurons or synaptic inputs from postsynaptic voltage recordings [82; 17]. Yu et al. [25] proposed Gaussian Process Factor Analysis (GPFA) for retrieving lower-dimensional, smooth latent neural trajectories from multiple spike train recordings. In GPFA, the correlation structure among the latent variables is specified (parameterized) explicitly rather than being given through a transition model. Buesing et al. [20], finally, discuss regularized forms of neural state space models to enforce their stability.

By far most of the models discussed above are linear in their latent dynamics, however (although observations may be non-Gaussian). As demonstrated in the Results, linear state space models may potentially be similarly well fit for reproducing actual observations, at least for the particular model and experimental systems studied here. In fact, this is not at all guaranteed in general, if the underlying processes are highly nonlinear (unlike those in Fig 5 where the nonlinearity was comparatively mild (not depending on multi-stability)). Thus, they may often be sufficient to obtain smoothed neural trajectories or lower-dimensional representations of the observed process [25], to uncover properties of the underlying connectivity [63; 81], or to estimate synaptic/neuronal parameters [16; 82]. However, as linear systems are

strongly limited in the repertoire of dynamics and computations they can produce (e.g. [65; 83]), they cannot serve as a model for the underlying computational processes and dynamics in general, and do not allow for the type of analyses which led into Fig 12. A LDS can, on its own, express at most one *isolated* fixed point (or a neutrally un-/stable continuum), or (neutrally un-/stable) sinusoidal-like cycles, but cannot represent any of the more complex phenomena which characterize physiological activity and are a hallmark of most computation. On the other hand, a direct comparison of LDS vs. PLRNN predictive performance may be highly revealing in itself: While some cognitive processes (like decision making, sequence or syntax generation) would clearly be expected to be highly nonlinear in their underlying dynamics [4; 84; 85], others (early stimulus responses, or value updating, for instance) may follow more of a linear rule (e.g., if stimuli were projected into a high-dimensional space for linear separability; cf. [86]). Directly contrasting LDS with PLRNN predictions on the same data set (as carried out in Fig 10), may uncover such important differences in computational mechanisms, and hence constitute an interesting analysis strategy in its own right.

There are a couple of other exceptions from the linear framework the current work builds on: Yu et al. [23] suggested a RNN with sigmoid-type activation function (using the error function), coupled to Poisson spike count outputs, and used it to reconstruct the latent neural dynamics underlying motor preparation and planning in non-human primates. In their work, they combined the Gaussian approximation suggested by Smith & Brown [15] with the Extended Kalman Filter (EKF) for estimation within the EM framework. These various approximations in conjunction with the iterative EKF estimation scheme may be quite prone to numerical instabilities and accumulating errors, however (cf. [26]). Earlier work by Roweis & Ghahramani [27] used radial basis function (RBF) networks as a partly analytically tracktable approach. Nonlinear extensions to DCM, incorporating quadratic terms, have been proposed as well recently [87]. State and parameter estimation has also been attempted in (noisy) nonlinear biophysical models [88; 89], but these approaches are usually computationally expensive, especially when based on numerical sampling [89], while at the same time pursuing objectives somewhat different from those targeted here (i.e., less focused on computational properties). A very recent article by Whiteway & Butts [90] discusses an approach closely related to the present one in that it also assumed piecewise linear latent states (or, 'rectified linear units (ReLU)'). Unlike here, however, the latent states were not connected through a dynamical systems model with separate process noise (but just constrained through a smoothness prior). Indeed, the objectives of this work were different, as Whiteway & Butts [90] aimed more at capturing unobserved sources of input in accounting for observed neural activity (more in the spirit of factor analysis), rather than attempting to retrieve an underlying stochastic dynamics as in the present work. They found, however, that the inclusion of nonlinearities may help in accounting for observed data and improve interpretability of the latent factors.

In summary, nonlinear neural state space models remain a relatively under-researched topic in theoretical neuroscience. PLRNNs, as chosen here, have the advantage of being mathematically comparatively tracktable, which allowed for the present, reasonably fast, semi-analytical algorithm, yet they are computationally and dynamically still powerful [91–94].

Alternative inference/training schemes, network architectures, and observation distributions

A number of other inference schemes have been suggested for state space models, comprising both analytical approximations [22] and numerical (sampling) techniques (e.g. [26]). Among the former are the Extended Kalman filter (based on local Taylor series approximations), methods based on variational inference as reviewed in Macke et al. [22], or the (global) Laplace

approximation advertized in Paninski et al. ([16]; see also [22]). Durbin & Koopman [26] review different variants of particle filter schemes for sequential numerical sampling. These may often be simpler to use, but are usually computationally much more costly than the semi-analytical methods. The Unscented Kalman Filter may be seen somewhere in between, using a few deliberately chosen sample ('sigma') points for a local parametric assessment [26]. Here we chose a global approach rather than a recursive-sequential scheme, that is by solving the full $M \times T$ system of linear equations within each subspace defined by constraints $\mathbf{\Omega}$ in one go. Apart from its generally nice computational properties as discussed in Paninski et al. [16], it seems particularly well-suited for the present piecewise-linear model Eqs (1) and (3), in dealing with the combinatorial explosion which builds up along the chain from $t = 1 \ldots T$. However, the mathematical properties of the present algorithm, among them issues of convergence/ monotonicity, local maxima/ saddles, and uniqueness and existence of solutions, certainly require further illumination which may lead to algorithmic improvements. In particular, identifiability of dynamics, that is to what degree and under which conditions the true underlying dynamical system could be recovered by the PLRNN-EM approach, remains an open issue (one line of extension toward greater approximation power would be polynomial basis expansions, at the cost, however, of losing the straightforward interpretation in terms of 'neural networks').

Most commonly, different variants of gradient-based techniques are being used to train recurrent neural networks to fit observations [40, 42, 50, 95, 96]. For instance, recurrent network models have been trained to perform behavioral tasks [43] or reproduce behavioral data to infer the dynamical mechanisms potentially underlying working memory [97] or context-dependent decision making [68]. In these settings, however, the observations–that is behavioral data points or requested task outputs–are usually relatively sparse in time compared to the time scale of the underlying dynamics, unlike the neural time series settings studied here where the data can be as dense as the latent state vectors of the model. More importantly, in contrast to these previous gradient-based approaches, the present scheme embeds RNNs into a statistical framework that comes with explicit probability assumptions, thereby puts error bars on state and parameter estimates and returns the posterior probability distribution across latent states, which links in with the observations through a separate measurement function (enabling, for instance, dimensionality reduction), and allows for likelihood-based statistical inference and model comparison. Some preliminary analyses using stochastic Adagrad [98] for training PLRNNs on the time series from the working memory example (cf. Fig 3) seemed, on top, to indicate that the resulting parameter estimates may correlate less well (<0.51 for $\mathbf{A}$ and $\mathbf{W}$, after optimal reordering of states) with the true model parameters than those obtained with the present EM approach (>0.78) for the lowest error/ highest likelihood solutions (this may potentially be improved through teacher forcing, which, however, is not applicable when the observed and latent space differ in dimensionality and are related through an, in general, not strictly invertible transform, as here).

Other observation models, like the Poisson model for spike counts [15; 22], are also relatively straightforward to accommodate within this framework (see [16]). However, there are also other ways to deal with spike count observations, like simple Box-Cox-type transforms to make them more Gaussian, e.g. the sqrt-transform suggested for GPFA [25], or kernel-density smoothing (e.g. [58]) as used here. The latter has the additional advantage of reducing the impact of 'binning noise', due to the somewhat arbitrary mapping of real-valued spike times onto discrete (user-defined) time bins for the purpose of counting. In general, from a practical perspective, it may therefore still be an open question of whether the additional computational burden that comes with non-Gaussian observation models (e.g. the requirement of Newton-Raphson steps for each mode-search iteration) pays off in the end compared to these alternatives. In either case, for the

time being, it seems useful to have a more general approach which can also deal with other measurement modalities, like neuroimaging data, which are not of a count-nature.

The present approach could also be extended by incorporating various additional structural features. For instance, a distinction between units with excitatory vs. inhibitory connections [43] could be accommodated quite easily within the present framework (requiring constrained optimization for weight parameters, however, e.g. through quadratic programming). Or special gated linear units which make LSTM networks so powerful [39,40] may potentially also yield improvements within the present EM/ state-space framework (although, in general, one may want to be cautious about the assumptions that additional structural elements like these may imply about the underlying neural system to be examined).

Mechanisms of working memory

Although the primary focus of this work was to develop and evaluate a state space framework for PLRNNs, some discussion of the applicational example chosen here, working memory, is in order. Working memory is generally defined as the ability to actively hold an item in memory, in the absence of guiding external input, for short-term reference in subsequent choice situations [99]. Various neural mechanisms have been proposed to underlie this cognitive capacity, most prominently multi-stable neural networks which retain short-term memory items by switching into one of several stimulus-selective attractor states [28; 29; 32]. These attractors usually represent fixed points in the firing rates, with assemblies of recurrently coupled stimulus-selective cells exhibiting high rates while those cells not coding for the present stimulus in short-term memory remaining at a spontaneous low-rate base level. These models were inspired by the physiological observation of 'delay-active' cells [100–102], that is cells that switch into a high-rate state during the delay periods of working memory tasks, and back to a low-rate state after completion of a trial, similar to the 'delay-active' latent states observed in Fig 11. Nakahara & Doya [103] were among the first to point out, however, that, for working memory, it may be completely sufficient (or even advantageous) to tune the system close to a bifurcation point where the dynamics becomes very slow (see also [1]), and true multi-stability may not be required. This is supported by the present observation that most of the estimated PLRNN models had fixed points with eigenvalues close to 1 but were not truly bi- or multi-stable (cf. Fig 12), yet this was sufficient to account for maintenance of stimulus-selectivity throughout the 10 s delay of the present task (cf. Fig 11) and for experimental observations (cf. Fig 9). Recently, a number of other mechanisms for supporting working memory, however, including sequential activation of cell populations [104] or synaptic mechanisms [105] have been discussed. Thus, the neural mechanisms of working memory remain an active research area to which statistical model estimation approaches as developed here may contribute, but too broad a topic in its own right to be covered in more depth by this mainly methodological work.

Models and methods

Expectation-maximization algorithm: State estimation

As with most previous work on estimation in (neural) state space models [20; 22; 23; 26], we use the Expectation-Maximization (EM) framework for obtaining estimates of both the model parameters and the underlying latent state path. Due to the piecewise-linear nature of model (1), however, the conditional latent state path density $p(\mathbf{Z}|\mathbf{X})$ is a high-dimensional 'mixture' of partial Gaussians, with the number of integrations to perform to obtain moments of $p(\mathbf{Z}|\mathbf{X})$ scaling as $2^{T\times M}$. Although analytically accessible, this will be computationally prohibitive for almost all cases of interest. Our approach therefore focuses on a computationally reasonably

efficient way of searching for the mode (maximum a-posteriori, MAP estimate) of p($\mathbf{Z}|\mathbf{X}$) which was found to be in good agreement with E($\mathbf{Z}|\mathbf{X}$) in most cases. Covariances were then approximated locally around the MAP estimate.

More specifically, the EM algorithm maximizes the expected log-likelihood of the joint distribution p($\mathbf{X},\mathbf{Z}$) as a lower bound on log $p(\mathbf{X}|\Xi)$ [27], where $\Xi = \{\boldsymbol{\mu}_0,\mathbf{A},\mathbf{W},\Sigma,\mathbf{B},\boldsymbol{\Gamma}\}$ denotes the set of to-be-optimized-for parameters (note that we dropped the thresholds $\boldsymbol{\theta}$ from this for now):

$$\begin{aligned} Q(\boldsymbol{\Xi},\mathbf{Z}) &:= E[\log p(\mathbf{Z},\mathbf{X}|\boldsymbol{\Xi})] \\ &= E\left[-\frac{1}{2}(\mathbf{z}_1-\boldsymbol{\mu}_0-\mathbf{s}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}_1-\boldsymbol{\mu}_0-\mathbf{s}_1)\right] \\ &\quad + E\left[-\frac{1}{2}\sum_{t=2}^{T}(\mathbf{z}_t-\mathbf{A}\mathbf{z}_{t-1}-\mathbf{W}\phi(\mathbf{z}_{t-1})-\mathbf{s}_t)^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}_t-\mathbf{A}\mathbf{z}_{t-1}-\mathbf{W}\phi(\mathbf{z}_{t-1})-\mathbf{s}_t)\right] \\ &\quad + E\left[-\frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t-\mathbf{B}\phi(\mathbf{z}_t))^T\boldsymbol{\Gamma}^{-1}(\mathbf{x}_t-\mathbf{B}\phi(\mathbf{z}_t))\right] - \frac{T}{2}(\log|\boldsymbol{\Sigma}|+\log|\boldsymbol{\Gamma}|). \end{aligned} \quad (5)$$

For state estimation (E-step), if ϕ were a linear function, obtaining $E(\mathbf{Z}|\mathbf{X},\Xi)$ would be equivalent to maximizing the argument of the expectancy in (5) w.r.t. $\mathbf{Z}$, i.e., $\mathrm{E}[\mathbf{Z}|\mathbf{X},\Xi] \equiv \arg\max_Z[\log p(\mathbf{Z},\mathbf{X}|\Xi)]$ (see [16]; see also [106]). This is because for a Gaussian mean and mode coincide. In our case, p($\mathbf{X},\mathbf{Z}$) is piecewise Gaussian, and we still take the approach (suggested in [16]) of maximizing log $p(\mathbf{Z},\mathbf{X}|\Xi)$ directly w.r.t. $\mathbf{Z}$ (essentially a Laplace approximation of $p(\mathbf{X}|\Xi)$ where we neglect the Hessian which is constant around the maximizer; cf. [16, 48]).

Let $\boldsymbol{\Omega}(t) \subseteq \{1\ldots M\}$ be the set of all indices of the units for which we have $z_{mt} \leq \theta_m$ at time t, and $\mathbf{W}_{\Omega(t)}$ and $\mathbf{B}_{\Omega(t)}$ the matrices $\mathbf{W}$ and $\mathbf{B}$, respectively, with all columns with indices $\in \boldsymbol{\Omega}(t)$ set to 0. The state estimation problem can then be formulated as

$$\begin{aligned} &\textit{maximize} \\ Q^*_{\boldsymbol{\Omega}}(\mathbf{Z}) &:= -\frac{1}{2}(\mathbf{z}_1-\boldsymbol{\mu}_0-\mathbf{s}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}_1-\boldsymbol{\mu}_0-\mathbf{s}_1) \\ &\quad -\frac{1}{2}\sum_{t=2}^{T}[\mathbf{z}_t-(\mathbf{A}+\mathbf{W}_{\Omega(t-1)})\mathbf{z}_{t-1}+\mathbf{W}_{\Omega(t-1)}\boldsymbol{\theta}-\mathbf{s}_t]^T\boldsymbol{\Sigma}^{-1}[\mathbf{z}_t-(\mathbf{A}+\mathbf{W}_{\Omega(t-1)})\mathbf{z}_{t-1}+\mathbf{W}_{\Omega(t-1)}\boldsymbol{\theta}-\mathbf{s}_t] \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t-\mathbf{B}_{\Omega(t)}\mathbf{z}_t+\mathbf{B}_{\Omega(t)}\boldsymbol{\theta})^T\boldsymbol{\Gamma}^{-1}(\mathbf{x}_t-\mathbf{B}_{\Omega(t)}\mathbf{z}_t+\mathbf{B}_{\Omega(t)}\boldsymbol{\theta}) \end{aligned} \quad (6)$$

$$w.r.t.\ (\boldsymbol{\Omega},\mathbf{Z}),\ \textit{subject to}\ z_{mt} \leq \theta_m\ \forall t, m \in \boldsymbol{\Omega}(t)\ \ \text{AND}\ \ z_{mt} > \theta_m\ \forall t, m \notin \boldsymbol{\Omega}(t)$$

Let us concatenate all state variables into one long column vector, $\mathbf{z} = (\mathbf{z}_1,\ldots,\mathbf{z}_T) = (z_{11}\ldots z_{mt}\ldots z_{MT})^T$, and unwrap the sums across time into large, block-banded $MT\times MT$ matrices (see [16; 83]) in which we combine all terms quadratic or linear in $\mathbf{z}$, or $\phi(\mathbf{z})$, respectively. Further, define $\mathbf{d}_{\boldsymbol{\Omega}}$ as the binary ($MT\times 1$) indicator vector which has 1s everywhere except for the entries with indices $\in \boldsymbol{\Omega} \subseteq \{1\ldots MT\}$ which are set to 0, and let $\mathbf{D}_{\boldsymbol{\Omega}} := diag(\mathbf{d}_{\boldsymbol{\Omega}})$ the $MT\times MT$ diagonal matrix formed from $\mathbf{d}_{\boldsymbol{\Omega}}$. Let $\boldsymbol{\Theta} := (\boldsymbol{\theta},\boldsymbol{\theta},\ldots,\boldsymbol{\theta})_{(MT\times 1)}$, and $\boldsymbol{\Theta}^{+M}$ the same vector shifted downward

by M positions, with the first M entries set to 0. One may then rewrite $Q^*_{\mathbf{\Omega}}(\mathbf{Z})$ in the form

$$\begin{aligned} Q^*_{\mathbf{\Omega}}(\mathbf{Z}) = -\frac{1}{2}[&\mathbf{z}^T(\mathbf{U}_0 + \mathbf{D}_\Omega\mathbf{U}_1 + \mathbf{U}_1^T\mathbf{D}_\Omega + \mathbf{D}_\Omega\mathbf{U}_2\mathbf{D}_\Omega)\mathbf{z} \\ &- \mathbf{z}^T(\mathbf{v}_0 + \mathbf{D}_\Omega\mathbf{v}_1 + \mathbf{V}_2 diag[\mathbf{d}_\Omega^{+M}]\mathbf{\Theta}^{+M} + \mathbf{V}_3\mathbf{D}_\Omega\mathbf{\Theta} + \mathbf{D}_\Omega\mathbf{V}_4\mathbf{D}_\Omega\mathbf{\Theta}) \\ &- (\mathbf{v}_0 + \mathbf{D}_\Omega\mathbf{v}_1 + \mathbf{V}_2 diag[\mathbf{d}_\Omega^{+M}]\mathbf{\Theta}^{+M} + \mathbf{V}_3\mathbf{D}_\Omega\mathbf{\Theta} + \mathbf{D}_\Omega\mathbf{V}_4\mathbf{D}_\Omega\mathbf{\Theta})^T\mathbf{z}] + const. \end{aligned} \tag{7}$$

The $MT\times MT$ matrices $\mathbf{U}_{\{0\ldots2\}}$ separate product terms that do not involve $\phi(\mathbf{z})$ ($\mathbf{U}_0$), involve multiplication by $\phi(\mathbf{z})$ only from the left-hand or right-hand side ($\mathbf{U}_1$), or from both sides ($\mathbf{U}_2$). Likewise, for the terms linear in $\mathbf{z}$, vector and matrix terms were separated that involved z_{mt} or θ_m conditional on $z_{mt} > \theta_m$ (please see the provided MatLab code for the exact composition of these matrices). For now, the important point is that we have $2^{M\times T}$ different quadratic equations, depending on the bits on and off in the binary vector $\mathbf{d}_\Omega$. Consequently, to obtain the MAP estimator for $\mathbf{z}$, in theory, one may consider all $2^{M\times T}$ different settings for $\mathbf{d}_\Omega$, for each solve the linear equations implied by $\partial Q^*_{\mathbf{\Omega}}(\mathbf{Z})/\partial\mathbf{Z} = 0$, and select among those for which the solution $\mathbf{z}_*$ is consistent with the considered set $\mathbf{\Omega}$ (if one exists; see below) the one which produces the largest value $Q^*_{\mathbf{\Omega}}(\mathbf{z}_*)$.

In practice, this is generally not feasible. Various solution methods for piecewise linear equations have been suggested in the mathematical programming literature in the past [107; 108]. For instance, some piecewise linear problems may be recast as a linear complementarity problem [109], but the pivoting methods often used to solve it work (numerically) well only for smaller scale settings [49]. Here we therefore settled on a similar, simple Newton-type iteration scheme as proposed in [49]. Specifically, if we denote by $\mathbf{z}_*(\mathbf{\Omega})$ the solution to Eq 7 obtained with the set of constraints $\mathbf{\Omega}$ active, the present scheme initializes with a random drawing of the $\{z_{mt}\}$, sets the components of $\mathbf{d}_\Omega$ for which $z_{mt} > \theta_m$ to 1 and all others to 0, and then keeps on alternating between (1) solving $\partial Q^*_{\mathbf{\Omega}}(\mathbf{Z})/\partial\mathbf{Z} = 0$ for $\mathbf{z}_*(\mathbf{\Omega})$ and (2) flipping the bits in $\mathbf{d}_\Omega$ for which $\mathrm{sgn}[2d_\Omega^{(k)} - 1] \neq \mathrm{sgn}[z_{*k}(\mathbf{\Omega}) - \theta_k]$, that is, for which the components of the vector

$$\mathbf{c} := (2\mathbf{d}_\Omega - \mathbf{1})^T \circ (\mathbf{\theta} - \mathbf{z}_*(\mathbf{\Omega})) \tag{8}$$

are positive, until the solution to $\partial Q^*_{\mathbf{\Omega}}(\mathbf{Z})/\partial\mathbf{Z} = 0$ is consistent with set $\mathbf{\Omega}$ (i.e., $\mathbf{c} \leq \mathbf{0}$).

For the problem as formulated in Brugnano & Casulli [49], these authors proved that such a solution always exists, and that the algorithm will always terminate after a finite (usually low) number of steps, given certain assumptions and provided the matrix that multiplies with the states $\mathbf{z}$ in $\partial Q^*_{\mathbf{\Omega}}(\mathbf{Z})/\partial\mathbf{Z} = 0$ (i.e., the Hessian of $Q^*_{\mathbf{\Omega}}(\mathbf{z}_*)$), fulfills certain conditions (Stieltjes-type; see [49] for details). This will usually *not* be the case for the present system; although the Hessian of $Q^*_{\mathbf{\Omega}}(\mathbf{z}_*)$ will be symmetric and positive-definite (with proper parameter settings), its off-diagonal elements may be either larger or smaller than 0. Moreover, for the problem considered here, all elements of the Hessian in (7) depend on $\mathbf{\Omega}$, while in [49] this is only the case for the on-diagonal elements (i.e., in [49] $\mathbf{D}_\Omega$ enters the Hessian only in additive, not multiplicative form as here). For these reasons, the Newton-type algorithm outlined above may not always converge to an exact solution (if one exists in this case) but may eventually cycle among non-solution configurations, or may not even always increase $Q(\mathbf{Z})$ (i.e., Eq 5!). To bypass this, the algorithm was always terminated if one of the following three conditions was met: (i) A solution to $\partial Q^*_{\mathbf{\Omega}}(\mathbf{Z})/\partial\mathbf{Z} = 0$ consistent with $\mathbf{\Omega}$ was encountered; (ii) a previously probed set $\mathbf{\Omega}$ was revisited; (iii) the constraint violation error defined by $\|\mathbf{c}_+\|_1$, the l_1 norm of the positive part of $\mathbf{c}$ defined in Eq 8, went up beyond a pre-specified tolerance level (this is essentially a fast proxy for assessing the likelihood, intended to speed up iterations by using quantities

already computed). With these modifications, we found that the algorithm would usually terminate after only a few iterations (<10 for the examined toy examples) and yield approximate solutions with only a few constraints still violated (<3% for the toy examples). As a caveat, unless condition (i) is met, this procedure implies that the returned solution may not even be locally optimal (in the strict mathematical sense–it would still be expected to live within an 'elevated' region of the optimization landscape defined by $Q(\mathbf{Z})$). On the other hand, since $Q(\mathbf{Z})$ cannot keep on increasing along a closed cycle (it must 'come back'), cycling implies there must be local maxima (or potentially saddles) located on the rims that separate different linear subspaces defined by $\mathbf{d}_\Omega$. Hence, for the elements k of $\mathbf{z}$ for which the constraints are still violated in the end, that is for which $c_k > 0$ in Eq 8, one may explicitly enforce the constraints by setting the violating states $\mathbf{z}_{\{k\}} = \boldsymbol{\theta}_{\{k\}}$, then solve again for the remaining states $\mathbf{z}_{\{l \neq k\}}$ (placing the solution on a ridge; or a quadratic program may be solved for the last step). Either way, it was found that even these approximate (and potentially not even locally optimal) solutions were generally (for the problems studied) in sufficiently good agreement with $\mathrm{E}(\mathbf{Z}|\mathbf{X})$.

In the case of full EM iterations (with the parameters unknown as well), it appeared that flipping violated constraints in $\mathbf{d}_\Omega$ one by one may often (for the scenarios studied here) improve overall performance, in the sense of yielding higher-likelihood solutions and less numerical problems (although it may leave more constraints violated in the end). Hence, this scheme was adopted here for the full EM, that is only the single bit k^* corresponding to the maximum element of vector $\mathbf{c}$ in Eq 8 was inverted on each iteration (the one with the largest wrong-side deviation from $\boldsymbol{\theta}$). In general, however, the resultant slow-down in the algorithm may not always be worth the performance gains; or a mixture of methods, with $d_{k*}^{l+1} = 1 - d_{k*}^{l}$ with $k^* := \arg\max_k \{c_k > 0\}$ early on, and $\mathbf{d}_{\{k\}}^{l+1} = \mathbf{1} - \mathbf{d}_{\{k\}}^{l} \forall k : c_k > 0$ during later iterations, may be considered.

Once a (local) maximum $\mathbf{z}^{\max}$ (or approximation thereof) has been obtained, the covariances may be read off from the inverse negative Hessian at $\mathbf{z}^{\max}$, i.e. the elements of

$$\mathbf{V} := (\mathbf{U}_0 + \mathbf{D}_\Omega \mathbf{U}_1 + \mathbf{U}_1^T \mathbf{D}_\Omega + \mathbf{D}_\Omega \mathbf{U}_2 \mathbf{D}_\Omega)^{-1}. \tag{9}$$

Note that this is a *local* estimate around the current maximizer $\mathbf{z}^{\max}$ (i.e., oblivious to the discontinuities at the borders of the linear subspaces defined by $\mathbf{d}_\Omega$). We then use these covariance estimates to obtain (estimates of) $\mathrm{E}[\boldsymbol{\phi}(\mathbf{z})]$, $\mathrm{E}[\mathbf{z}\boldsymbol{\phi}(\mathbf{z})^T]$, and $\mathrm{E}[\boldsymbol{\phi}(\mathbf{z})\boldsymbol{\phi}(\mathbf{z})^T]$, as required for the maximization step. Denoting by $F(\lambda; \mu, \sigma^2) := \int_\lambda^\infty N(x; \mu, \sigma^2) dx$ the complementary cumulative Gaussian, to ease subsequent derivations, let us introduce the following notation:

$$N_k := N(\theta_k; z_k^{\max}, \sigma_k^2), \quad F_k := F(\theta_k; z_k^{\max}, \sigma_k^2), \quad \sigma_{kl}^2 := \mathrm{cov}(z_k^{\max}, z_l^{\max}) \approx v_{kl}. \tag{10}$$

The elements of the expectancy vectors and matrices above are computed as

$$\begin{aligned} \mathrm{E}[\phi(z_k)] &= \sigma_k^2 N_k + (z_k^{\max} - \theta_k) F_k, \\ \mathrm{E}[\phi(z_k)^2] &= ([z_k^{\max}]^2 + \sigma_k^2 + \theta_k^2 - 2\theta_k z_k^{\max}) F_k + (z_k^{\max} - \theta_k) \sigma_k^2 N_k, \\ \mathrm{E}[z_k \phi(z_l)] &= (\sigma_{kl}^2 - \theta_l z_k^{\max} + z_k^{\max} z_l^{\max}) F_l + z_k^{\max} \sigma_l^2 N_l. \end{aligned} \tag{11}$$

The terms $\mathrm{E}[\phi(z_k)\phi(z_l)]$, for $k \neq l$, are more tedious, and cannot be (to my knowledge and insight) computed exactly (analytically), so we develop them in a bit more detail here:

$$\begin{aligned}\mathrm{E}[\phi(z_k)\phi(z_l)] &= \int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)(z_k-\theta_k)(z_l-\theta_l)dz_k dz_l \\ &= \int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_k z_l dz_k dz_l - \theta_k\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_l dz_k dz_l - \theta_l\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_k dz_k dz_l \\ &\quad + \theta_k\theta_l\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)dz_k dz_l\end{aligned} \tag{12}$$

The last term is just a (complementary) cumulative bivariate Gaussian evaluated with parameters specified through the approximate MAP solution ($\mathbf{z}^{\max}$, $\mathbf{V}$) (and multiplied by the thresholds). The first term we may rewrite as follows:

$$\begin{aligned}\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_k z_l dz_k dz_l &= \int_{\theta_k}^{\infty} p(z_k)z_k\int_{\theta_l}^{\infty} p(z_l|z_k)z_l dz_k dz_l \\ &= \int_{\theta_k}^{\infty} p(z_k)z_k\left[N(\theta_l;\mu_{l|k},\lambda_l^{-1}) + \mu_{l|k}\left(1-\int_{-\infty}^{\theta_l} N(z_l;\mu_{l|k},\lambda_{lk}^{-1})dz_l\right]dz_k\right.\end{aligned} \tag{13}$$

where $\quad \mu_{l|k} := z_l^{\max} - \lambda_l^{-1}\lambda_{lk}(z_k - z_k^{\max})$

$\lambda_l := \sigma_l^2/(\sigma_k^2\sigma_l^2 - \sigma_{kl}^4)$

$\lambda_{lk} := -\sigma_{kl}^2/(\sigma_k^2\sigma_l^2 - \sigma_{kl}^4)$

These are just standard results one can derive by the reverse chain rule for integration, with the λ's the elements of the inverse bivariate (k,l)-covariance matrix. Note that if the variable z_k were removed from the first integrand in Eq 13, i.e. as in the second term in Eq 12, all terms in Eq 13 would just come down to uni- or bivariate Gaussians (times some factor) or a univariate Gaussian expectancy value, respectively. Noting this, one obtains for the second (and correspondingly for the third) term in Eq 12:

$$\begin{aligned}\theta_k\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_l dz_k dz_l &= \theta_k\lambda_k N_l F(\theta_k;\mu_{lk},\lambda_l^{-1}) + \theta_k(z_l^{\max}F_k + \sigma_{kl}^2 N_k)F(\theta_l;z_l^{\max},\lambda_k^{-1}) \\ &\text{with } \mu_{kl} := z_l^{\max} + \sigma_{kl}^2/\sigma_k^2(\theta_k - z_k^{\max})\end{aligned} \tag{14}$$

The problematic bit is the product term $\int_{\theta_k}^{\infty} p(z_k)z_k\mu_{l|k}\int_{-\infty}^{\theta_l} N(z_l;\mu_{l|k},\lambda_{lk}^{-1})dz_l dz_k$ in Eq 13, which we resolve by making the approximation $\mu_{l|k} \approx \mu_l = z_l^{\max}$. This way we have for the first term in Eq 12:

$$\begin{aligned}\int_{\theta_k}^{\infty}\int_{\theta_l}^{\infty} p(z_k,z_l)z_k z_l dz_k dz_l \approx{}& \lambda_k N_l[\lambda_l^{-1}N(\theta_k;\mu_{lk},\lambda_l^{-1}) + \mu_{lk}F(\theta_k;\mu_{lk},\lambda_l^{-1})] \\ &+ [(\sigma_k^2 z_l^{\max} - \sigma_{kl}^2 z_k^{\max})N_k + (z_k^{\max}z_l^{\max} + \sigma_{kl}^2)F_k]F(\theta_l;\mu_{lk},\lambda_k^{-1})\end{aligned} \tag{15}$$

Putting (13)–(15) together with the bivariate cumulative Gaussian yields an analytical approximation to Eq 12 that can be computed based on the quantities obtained from the approximate MAP estimate ($\mathbf{z}^{\max}$, $\mathbf{V}$).

Expectation-maximization algorithm: Parameter estimation

Once we have estimates for $E[\mathbf{z}]$, $E[\mathbf{z}\mathbf{z}^T]$, $E[\phi(\mathbf{z})]$, $E[\mathbf{z}\phi(\mathbf{z})^T]$, and $E[\phi(\mathbf{z})\phi(\mathbf{z})^T]$, the maximization step is standard and straightforward, so for convenience we just state the results here, using the notation

$$\begin{aligned}
&\mathbf{E}_{1,\Delta} \coloneqq \sum_{t=1}^{T-\Delta} E[\phi(\mathbf{z}_t)\phi(\mathbf{z}_t)^T] \quad , \quad \mathbf{E}_2 \coloneqq \sum_{t=2}^{T} E[\mathbf{z}_t \mathbf{z}_{t-1}^T], \quad \mathbf{E}_{3,\Delta} \coloneqq \sum_{t=1+\Delta}^{T-1+\Delta} E[\mathbf{z}_t \mathbf{z}_t^T], \\
&\mathbf{E}_4 \coloneqq \sum_{t=1}^{T-1} E[\phi(\mathbf{z}_t)\mathbf{z}_t^T] \quad , \quad \mathbf{E}_5 \coloneqq \sum_{t=2}^{T} E[\mathbf{z}_t \phi(\mathbf{z}_{t-1})^T], \\
&\mathbf{F}_1 \coloneqq \sum_{t=1}^{T} \mathbf{x}_t E[\phi(\mathbf{z}_t)^T] \quad , \quad \mathbf{F}_2 \coloneqq \sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^T \quad , \quad \mathbf{F}_3 \coloneqq \sum_{t=2}^{T} \mathbf{s}_t E[\mathbf{z}_{t-1}^T], \\
&\mathbf{F}_4 \coloneqq \sum_{t=2}^{T} \mathbf{s}_t E[\phi(\mathbf{z}_{t-1})^T] \quad , \quad \mathbf{F}_5 \coloneqq \sum_{t=1}^{T} E[\mathbf{z}_t]\mathbf{s}_t^T \quad , \quad \mathbf{F}_6 \coloneqq \sum_{t=1}^{T} \mathbf{s}_t \mathbf{s}_t^T
\end{aligned} \tag{16}$$

With these expectancy sums defined, one has

$$\mathbf{B} = \mathbf{F}_1 \mathbf{E}_{1,0}^{-1} \tag{17A}$$

$$\mathbf{\Gamma} = \frac{1}{T}(\mathbf{F}_2 - \mathbf{F}_1\mathbf{B}^T - \mathbf{B}\mathbf{F}_1^T + \mathbf{B}\mathbf{E}_{1,0}^T\mathbf{B}^T) \circ \mathbf{I} \tag{17B}$$

$$\boldsymbol{\mu}_0 = E[\mathbf{z}_1] - \mathbf{s}_1 \tag{17C}$$

$$\mathbf{A} = [(\mathbf{E}_2 - \mathbf{W}\mathbf{E}_4 - \mathbf{F}_3) \circ \mathbf{I}][\mathbf{E}_{3,0} \circ \mathbf{I}]^{-1} \tag{17D}$$

$$\begin{aligned}
\mathbf{\Sigma} = \frac{1}{T}[&\mathrm{var}(\mathbf{z}_1) + \boldsymbol{\mu}_0\mathbf{s}_1^T + \mathbf{s}_1\boldsymbol{\mu}_0^T + \mathbf{E}_{3,1}^T - \mathbf{F}_5 - \mathbf{F}_5^T + \mathbf{F}_6 + (\mathbf{F}_3 - \mathbf{E}_2)\mathbf{A}^T + \mathbf{A}(\mathbf{F}_3^T - \mathbf{E}_2^T) + \mathbf{A}\mathbf{E}_{3,0}^T\mathbf{A}^T \\
&+ (\mathbf{F}_4 - \mathbf{E}_5)\mathbf{W}^T + \mathbf{W}(\mathbf{F}_4^T - \mathbf{E}_5^T) + \mathbf{W}\mathbf{E}_{1,1}^T\mathbf{W}^T + \mathbf{A}\mathbf{E}_4^T\mathbf{W}^T + \mathbf{W}\mathbf{E}_4\mathbf{A}^T] \circ \mathbf{I}
\end{aligned} \tag{17E}$$

Note that to avoid redundancy in the parameters, here we usually fixed $\mathbf{\Sigma} = \mathbf{I} \cdot 10^{-2}$ (for the toy models) or $\mathbf{\Sigma} = \mathbf{I}$ (for the experimental data).

For $\mathbf{W}$, since we assumed this matrix to have an off-diagonal structure (i.e., with zeros on the diagonal), we solve for each row of $\mathbf{W}$ separately:

$$\begin{aligned}
&\mathbf{P}^{(0)} \coloneqq (\mathbf{E}_{3,0} \circ \mathbf{I})^{-1}\mathbf{E}_4^T \\
&\mathbf{P}^{(1)} \coloneqq \mathbf{E}_5 - [(\mathbf{E}_2 - \mathbf{F}_3) \circ \mathbf{I}]\mathbf{P}^{(0)} - \mathbf{F}_4 \\
&\forall m \in \{1 \ldots M\} : \mathbf{W}_{m,\{1:M\}\backslash \mathrm{m}} = \mathbf{P}^{(1)}_{m,\{1:M\}\backslash \mathrm{m}}([\mathbf{E}_{1,1} - \mathbf{E}_{4,\bullet m}\mathbf{P}^{(0)}_{m\bullet}]_{\{1:M\}\backslash \mathrm{m},\{1:M\}\backslash \mathrm{m}})^{-1}
\end{aligned} \tag{17F}$$

where the subscripts indicate the matrix elements to be pulled out, with the subscript dot denoting all elements of the corresponding column or row (e.g., '•m' takes the m^{th} column of that matrix). Should matrices $\mathbf{\Gamma}$, Σ, $\mathbf{W}$ of full form be desired, the derivations simplify a bit–in essence, the diagonal operator '$\circ\mathbf{I}$' in the equations above (except Eq 17D) would have to be omitted, and Eq 17F could be solved in full matrix form (instead of row-wise). An expression for input scaling matrix $\mathbf{C}$ (cf. Eq 1) is given by $\mathbf{C} = (\mathbf{F}_5 - \mathbf{z}_1\mathbf{s}_1^T - \mathbf{W}\mathbf{F}_4^T - \mathbf{A}\mathbf{F}_3^T)(\mathbf{F}_6 - \mathbf{s}_1\mathbf{s}_1^T)^{-1}$, but note that $\mathbf{C}$ would also show up in (17C)–(17F) (multiplying with $\mathbf{s}_t$ everywhere), as well as in the state inference equations; matrices $\mathbf{A}$, $\mathbf{W}$, and $\mathbf{C}$ would therefore need to be solved for

simultaneously in this case (complicating the above expressions a bit; see provided MatLab code for full details).

Starting from a number of different random parameter initializations, the E- and M-steps are alternated until the log-likelihood ratio falls below a predefined tolerance level (while still increasing) or a preset maximum number of allowed iterations are exceeded. For reasons mentioned in the Results, sometimes it can actually happen that the log-likelihood ratio temporarily decreases, in which case the iterations are continued. If $(N–M)^2 \geq N + M$, factor analysis may be used to derive initial estimates for the latent states and observation parameters in (3) [27], although this was not attempted here. Another possibility is to improve initial estimates first through the much faster, corresponding LDS, before submitting them to full PLRNN estimation. For further implementational details see the MatLab code provided on GitHub (repository 'PLRNNstsp').

Particle filter

To validate the approximations from our semi-analytical procedure developed above, a bootstrap particle filter as given in [26] was implemented. In bootstrap particle filtering, the state posterior distribution at time t,

$$\begin{aligned} p_{\Xi}(\mathbf{z}_t|\mathbf{x}_1,\ldots,\mathbf{x}_t) &= \frac{p_{\Xi}(\mathbf{x}_t|\mathbf{z}_t)p_{\Xi}(\mathbf{z}_t|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1})}{p_{\Xi}(\mathbf{x}_t|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1})} \\ &= \frac{p_{\Xi}(\mathbf{x}_t|\mathbf{z}_t)\int_{\mathbf{z}_{t-1}} p_{\Xi}(\mathbf{z}_t|\mathbf{z}_{t-1})p_{\Xi}(\mathbf{z}_{t-1}|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1})d\mathbf{z}_{t-1}}{p_{\Xi}(\mathbf{x}_t|\mathbf{x}_1,\ldots,\mathbf{x}_{t-1})} \end{aligned} \tag{18}$$

is numerically approximated through a set of 'particles' (samples) $\{\mathbf{z}_t^{(1)},\ldots,\mathbf{z}_t^{(K)}\}$, drawn from $p_{\Xi}(\mathbf{z}_t \mid \mathbf{x}_1,\ldots,\mathbf{x}_{t-1})$, together with a set of normalized weights $\{w_t^{(1)},\ldots,w_t^{(K)}\}$, $w_t^{(r)} \coloneqq p_{\Xi}(\mathbf{x}_t|\mathbf{z}_t^{(r)})\left(\sum_{k=1}^{K} p_{\Xi}(\mathbf{x}_t|\mathbf{z}_t^{(k)})\right)^{-1}$. Based on this representation, moments of $p_{\Xi}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$ and $p_{\Xi}(\phi(\mathbf{z}_t) \mid \mathbf{x}_{1:t})$ can be easily obtained by evaluating ϕ (or any other function of $\mathbf{z}$) on the set of samples $\{\mathbf{z}_t^{(r)}\}$ and summing the outcomes weighted with their respective normalized observation likelihoods $\{w_t^{(r)}\}$. A new set of samples $\{\mathbf{z}_{t+1}^{(r)}\}$ for t+1 is then generated by first drawing K times from $\{\mathbf{z}_t^{(k)}\}$ with replacement according to the weights $\{w_t^{(k)}\}$, and then drawing K new samples according to the transition probabilities $p_{\Xi}(\mathbf{z}_{t+1}^{(k)}|\mathbf{z}_t^{(k)})$ (thus approximating the integral in Eq 18). Here we used $K = 10^4$ samples. Note that this numerical sampling scheme, like a Kalman filter, but unlike the procedure outlined above, only implements the filtering step (i.e., yields $p_{\Xi}(\mathbf{z}_t \mid \mathbf{x}_{1:t})$, not $p_{\Xi}(\mathbf{z}_t \mid \mathbf{x}_{1:T})$). On the other hand, it gives (weakly) consistent (asymptotically unbiased; [110; 111]) estimates of all expectancies across this distribution, that is, it does not rely on the type of approximations and locally optimal solutions of our semi-analytical approach that almost inevitably will come with some bias (since, among other factors, the local or approximate mode would usually deviate from the mean by some amount for the present model).

Experimental data sets

Details of the experimental task and electrophysiological data sets used here could be found in [41, 112]. Briefly, rats had to alternate between left and right lever presses in a Skinner box to obtain a food reward dispensed on correct choices, with a $\geq$ 10 s delay enforced between consecutive lever presses. While the levers were located on one side of the Skinner box, animals had to perform a nosepoke on the opposite side of the box in between lever presses for

initiating the delay period, to discourage them from developing an external coding strategy (e.g., through maintenance of body posture during the delay). While animals were performing the task, multiple single units were recorded with a set of 16 tetrodes implanted bilaterally into the anterior cingulate cortex (ACC, a subdivision of rat prefrontal cortex). For the present analyses, a data set from only one of the four rats recorded on this task was selected for the present exemplary purposes, namely the one where the clearest single unit traces of delay activity were observed in the first place. This data set consisted of 30 simultaneously recorded units, of which the 19 units with spiking rates >1 Hz were retained, on 14 correct trials (only correct response trials were analyzed). The trials had variable length, but were all cut down to the same length of 14 s, including 2 s of pre-nosepoke, 5 s extending into the delay from the nosepoke, 5 s preceding the next lever press, and 2 s of post-response phase (note that this may imply temporal gaps in the middle of the delay on some trials, which were ignored here for convenience). All spike trains were convolved with Gaussian kernels (see, e.g., [12; 57; 112]), with the kernel standard deviation set individually for each unit to one half of its mean inter-spike-interval. Note that this also brings the observed series into tighter agreement with the Gaussian assumptions of the observation model, Eq 3. Finally, the spike time series were binned into 500 ms bins (corresponding roughly to the inverse of the overall (across all 30 recorded cells) average neural firing rates of ≈2.2 Hz), which resulted in 14 trials of 28 time bins each submitted to the estimation process. As indicated in the section '*State space model*', a trial-unique initial state mean $\boldsymbol{\mu}_k$, $k = 1...14$, was assumed for each of the 14 temporally segregated trials.

Acknowledgments

I thank Dr. Georgia Koppe for her feedback on this manuscript, Dr. Hazem Toutounji for providing the Matlab routine for efficient inversion of block-tridiagonal matrices, and for discussion of related issues, and Drs. James Hyman and Jeremy Seamans for lending me their in-vivo electrophysiological recordings from rat ACC as an analysis testbed.

Author Contributions

Conceptualization: DD.

Formal analysis: DD.

Funding acquisition: DD.

Methodology: DD.

Software: DD.

Writing – original draft: DD.

Writing – review & editing: DD.

References

1. Durstewitz D. Self-organizing neural integrator predicts interval times through climbing activity. J Neurosci. 2003; 23: 5342–5353. PMID: 12832560
2. Wang XJ. Probabilistic decision making by slow reverberation in cortical circuits. Neuron. 2002; 36: 955–968. PMID: 12467598
3. Izhikevich EM. Dynamical Systems in Neuroscience. 2007; MIT Press.
4. Rabinovich MI, Huerta R, Varona P, Afraimovich VS. Transient cognitive dynamics, metastability, and decision making. PLoS Comput Biol. 2008; 2: e1000072.

5. Pillow JW, Shlens J, Chichilnisky EJ, Simoncelli EP. A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings. PLoS One. 2013; 8: e62123 https://doi.org/10.1371/journal.pone.0062123 PMID: 23671583
6. Stevens CF. Neurotransmitter release at central synapses. Neuron. 2003; 40: 381–388. PMID: 14556715
7. Pouget A1, Beck JM, Ma WJ, Latham PE, Probabilistic brains: knowns and unknowns. Nat Neurosci 2013,1170–8. https://doi.org/10.1038/nn.3495 PMID: 23955561
8. Orbán G, Wolpert DM, Representations of uncertainty in sensorimotor control. Curr Opin Neurobiol 2011, 629–35. https://doi.org/10.1016/j.conb.2011.05.026 PMID: 21689923
9. Körding KP, Wolpert DM, Bayesian integration in sensorimotor learning. Nature 2004, 427: 244–7. https://doi.org/10.1038/nature02169 PMID: 14724638
10. Durstewitz D, A few important points about dopamine's role in neural network dynamics. Pharmacopsychiatry 2006, 39 Suppl 1: S72–S75.
11. Balaguer-Ballester E, Lapish CC, Seamans JK, Durstewitz D. Attractor Dynamics of Cortical Populations During Memory-Guided Decision-Making. PLoS Comput Biol. 2011; 7: e1002057. https://doi.org/10.1371/journal.pcbi.1002057 PMID: 21625577
12. Lapish CC, Balaguer-Ballester E, Seamans JK, Phillips AG, Durstewitz D. Amphetamine Exerts Dose-Dependent Changes in Prefrontal Cortex Attractor Dynamics during Working Memory. J Neurosci. 2015; 35: 10172–10187. https://doi.org/10.1523/JNEUROSCI.2421-14.2015 PMID: 26180194
13. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci U S A. 2016; 113: 3932–3937. https://doi.org/10.1073/pnas.1517384113 PMID: 27035946
14. Wood SN. Statistical inference for noisy nonlinear ecological dynamic systems. Nature. 2010; 466: 1102–1104. https://doi.org/10.1038/nature09319 PMID: 20703226
15. Smith AC, Brown EN. Estimating a state-space model from point process observations. Neural Comput. 2003; 15: 965–991. https://doi.org/10.1162/089976603765202622 PMID: 12803953
16. Paninski L, Ahmadian Y, Ferreira DG, Koyama S, Rahnama RK, Vidne M, et al. A new look at state-space models for neural data. J Comput Neurosci. 2010; 29: 107–126. https://doi.org/10.1007/s10827-009-0179-x PMID: 19649698
17. Paninski L, Vidne M, DePasquale B, Ferreira DG. Inferring synaptic inputs given a noisy voltage trace via sequential Monte Carlo methods. J Comput Neurosci. 2012; 33: 1–19. https://doi.org/10.1007/s10827-011-0371-7 PMID: 22089473
18. Pillow JW, Shlens J, Paninski L, Sher A, Litke AM, Chichilnisky EJ, et al. Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature. 2008; 454: 995–999. https://doi.org/10.1038/nature07140 PMID: 18650810
19. Pillow JW, Ahmadian Y, Paninski L. Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. Neural Comput. 2011; 23: 1–45. https://doi.org/10.1162/NECO_a_00058 PMID: 20964538
20. Buesing L, Macke JH, Sahani M. Learning stable, regularised latent models of neural population dynamics. Network. 2012; 23: 24–47. https://doi.org/10.3109/0954898X.2012.677095 PMID: 22663075
21. Latimer KW, Yates JL, Meister ML, Huk AC, Pillow JW. Single-trial spike trains in parietal cortex reveal discrete steps during decision-making. Science. 2015; 349: 184–187. https://doi.org/10.1126/science.aaa4056 PMID: 26160947
22. Macke JH, Buesing L, Sahani M. Estimating State and Parameters in State Space Models of Spike Trains. In: Chen Z editor. Advanced State Space Methods for Neural and Clinical Data. Cambridge: University Press; 2015.
23. Yu BM, Afshar A, Santhanam G, Ryu SI, Shenoy KV. Extracting Dynamical Structure Embedded in Neural Activity. Adv Neural Inf Process Syst. 2005; 18: 1545–1552.
24. Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI Meng TH, et al. Mixture of trajectory models for neural decoding of goal-directed movements. J Neurophysiol. 2007; 5: 3763–3780.
25. Yu BM, Cunningham JP, Santhanam G, Ryu SI, Shenoy KV, Sahani M. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. J Neurophysiol. 2009; 102: 614–635. https://doi.org/10.1152/jn.90941.2008 PMID: 19357332
26. Durbin J, Koopman SJ. Time Series Analysis by State Space Methods. Oxford Statistical Science; 2012.
27. Roweis ST, Ghahramani Z. An EM algorithm for identification of nonlinear dynamical systems. In: Haykin S, editor. Kalman Filtering and Neural Networks; 2001

28. Amit DJ, Brunel N. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. Cereb Cortex. 1997; 7: 237–252. PMID: 9143444

29. Durstewitz D, Seamans JK, Sejnowski TJ. Neurocomputational models of working memory. Nat Neurosci. 2000; 3; Suppl: 1184–1191.

30. Durstewitz D. Implications of synaptic biophysics for recurrent network dynamics and active memory. Neural Netw. 2009; 22: 1189–1200. https://doi.org/10.1016/j.neunet.2009.07.016 PMID: 19647396

31. Brunel N, Wang XJ. Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. J Comput Neurosci. 2001; 11: 63–85. PMID: 11524578

32. Wang XJ. Synaptic basis of cortical persistent activity: the importance of NMDA receptors to working memory. J Neurosci. 1999; 19: 9587–9603. PMID: 10531461

33. Durstewitz D, Gabriel T. Dynamical basis of irregular spiking in NMDA-driven prefrontal cortex neurons. Cereb Cortex. 2007; 17: 894–908. https://doi.org/10.1093/cercor/bhk044 PMID: 16740581

34. Funahashi KI, Nakamura Y. Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks. Neural Netw. 1993; 6: 801–806.

35. Kimura M, Nakano R. Learning dynamical systems by recurrent neural networks from orbits. Neural Netw. 1998; 11: 1589–1599. PMID: 12662730

36. Chow TWS, Li XD. Modeling of Continuous Time Dynamical Systems with Input by Recurrent Neural Networks. Trans Circuits Syst I Fundam Theory Theory Appl. 2000; 47: 575–578.

37. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521: 436–444. https://doi.org/10.1038/nature14539 PMID: 26017442

38. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518: 529–533. https://doi.org/10.1038/nature14236 PMID: 25719670

39. Schmidhuber J. Deep learning in neural networks. Neural Networks 2015, 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003 PMID: 25462637

40. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997; 9: 1735–1780. PMID: 9377276

41. Hyman JM, Whitman J, Emberly E, Woodward TS, Seamans JK. Action and outcome activity state patterns in the anterior cingulate cortex. Cereb Cortex. 2013; 23: 1257–1268. https://doi.org/10.1093/cercor/bhs104 PMID: 22617853

42. Sussillo D, Abbott LF. Generating coherent patterns of activity from chaotic neural networks. Neuron. 2009; 63: 544–557. https://doi.org/10.1016/j.neuron.2009.07.018 PMID: 19709635

43. Song HF, Yang GR, Wang XJ. Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible Framework. PLoS Comput Biol 12, e1004792. https://doi.org/10.1371/journal.pcbi.1004792 PMID: 26928718

44. Fan J, Yao Q. Nonlinear Time Series: Nonparametric and Parametric Methods. Springer, New York. 2003

45. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning (Vol. 2, No. 1) Springer, New York 2009

46. Park M, Bohner G, Macke J. Unlocking neural population non-stationarity using a hierarchical dynamics model In: Advances in Neural Information Processing Systems 28, Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS 2015); 2016. pp.1-9.

47. Beer RD. Parameter Space Structure of Continuous-Time Recurrent Neural Networks. Neural Computation 2006; 18: 3009–3051. https://doi.org/10.1162/neco.2006.18.12.3009 PMID: 17052157

48. Koyama S., Pérez-Bolde L.C., Shalizi C.R., Kass R.E.: Approximate Methods for State-Space Models. J. Am. Stat. Assoc. 2010; 105: 170–180. https://doi.org/10.1198/jasa.2009.tm08326 PMID: 21753862

49. Brugnano L, Casulli V. Iterative solution of piecewise linear systems. SIAM J Sci Comput. 2008; 30: 463–472.

50. Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural Computat. 1990; 1: 256–263

51. Hertz J, Krogh AS, Palmer RG. Introduction to the theory of neural computation. 1991; Addison-Wesley Pub Co.

52. Zhang K, Hyvärinen A. A General Linear Non-Gaussian State-Space Model: Identifiability, Identification, and Applications. JMLR: Workshop and Conference Proceedings 2011; 20: 113–128.

53. Auger-Méthé M, Field C, Albertsen CM, Derocher AE, Lewis MA, Jonsen ID, et al. State-space models' dirty little secrets: even simple linear Gaussian models can have estimation problems. Sci Rep. 2016; 6: 26677. https://doi.org/10.1038/srep26677 PMID: 27220686

54. Wu CFJ. On the Convergence Properties of the EM Algorithm. Ann Statist. 1983; 11: 95–103.

55. Boutayeb M, Rafaralahy H, Darouach M. Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems. IEEE Trans Autom Control. 1997; 42: 581–586.

56. Megiddo. Advances in Economic Theory. Fifth World Congress. Edited by Bewley Truman F. Cambridge University Press 1987

57. Durstewitz D, Vittoz NM, Floresco SB, Seamans JK (2010) Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning. Neuron 66: 438–48. https://doi.org/10.1016/j.neuron.2010.03.029 PMID: 20471356

58. Shimazaki H, Shinomoto S. Kernel Bandwidth Optimization in Spike Rate Estimation. J Comp Neurosci. 2010; 29: 171–182.

59. Janson et al. Effective Degrees of Freedom: A Flawed Metaphor. Lucas Janson, Will Fithian, Trevor Hastie. Biometrika. 2015, 102: 479–485. https://doi.org/10.1093/biomet/asv019 PMID: 26977114

60. Hyman JM, Ma L, Balaguer-Ballester E, Durstewitz D, Seamans JK. Contextual encoding by ensembles of medial prefrontal cortex neurons. Proc Natl Acad Sci USA. 2012, 109: 5086–5091 https://doi.org/10.1073/pnas.1114415109 PMID: 22421138

61. Latham PE, Nirenberg S. Computing and stability in cortical networks. Neural Comput. 2004; 16: 1385–1412. https://doi.org/10.1162/089976604323057434 PMID: 15165395

62. Durstewitz D, Seamans JK. Beyond bistability: biophysics and temporal dynamics of working memory. Neuroscience. 2006; 139: 119–133. https://doi.org/10.1016/j.neuroscience.2005.06.094 PMID: 16326020

63. Friston KJ, Harrison L, Penny W. Dynamic causal modelling. Neuroimage. 2003; 19: 1273–1302. PMID: 12948688

64. Walter E., & Pronzato L. (1996). On the identifiability and distinguishability of nonlinear parametric models. Mathematics and Computers in Simulation, 42(2–3), 125–134.

65. Strogatz SH. Nonlinear dynamics and chaos. Addison-Wesley Publ; 1994.

66. Durstewitz D, Kelc M, Güntürkün O. A neurocomputational theory of the dopaminergic modulation of working memory functions. J Neurosci. 1999; 19: 207–222.

67. Brunel N. Dynamics of Sparsely Connected Networks of Excitatory and Inhibitory Spiking Neurons. J Comput Neurosci. 2000; 8: 183–208. PMID: 10809012

68. Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature. 2013; 503: 78–84. https://doi.org/10.1038/nature12742 PMID: 24201281

69. Hertäg L, Durstewitz D, Brunel N. Analytical approximations of the firing rate of an adaptive exponential integrate-and-fire neuron in the presence of synaptic noise. Front Comput Neurosci. 2014; 8: 116. https://doi.org/10.3389/fncom.2014.00116 PMID: 25278872

70. Sussillo D, Barak O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. Neural Comput. 2013; 25: 626–649. https://doi.org/10.1162/NECO_a_00409 PMID: 23272922

71. Takens F. Detecting strange attractors in turbulence. Lecture Notes in Mathematics 898. Springer Berlin;1981: 366–381.

72. Sauer TD, Sauer K, Davies DG. Embedology. J Stat Phys. 1991; 65: 579–616.

73. Sauer T. Reconstruction of dynamical systems from interspike intervals. Phys Rev Lett. 1994; 72: 3811–3814. https://doi.org/10.1103/PhysRevLett.72.3811 PMID: 10056303

74. So P, Francis JT, Netoff TI, Gluckman BJ, Schiff SJ. Periodic Orbits: A New Language for Neuronal Dynamics. Biophys J. 1998; 74: 2776–2785 https://doi.org/10.1016/S0006-3495(98)77985-8 PMID: 9635732

75. Takahashi S, Anzai Y, Sakurai Y. A new approach to spike sorting for multi-neuronal activities recorded with a tetrode—how ICA can be practical. Neurosci Res. 2003a; 46: 265–272.

76. Takahashi S, Anzai Y, Sakurai Y. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. J Neurophysiol. 2003b; 89: 2245–2258.

77. Hille B. Ion channels of excitable membranes. 3rd ed. Sinauer Assoc Inc; 2001.

78. Kantz H, Schreiber T. Nonlinear Time Series Analysis. Cambridge University Press; 2004.

79. Schreiber T, Kantz H. Observing and predicting chaotic signals: Is 2% noise too much? In: Kravtsov YA, Kadtke JB, editors. Predictability of Complex Dynamical Systems, Springer, New York;1996.

80. Zhao Park 2016. Interpretable Nonlinear Dynamic Modeling of Neural Trajectories Yuan Zhao, Il Memming Park. Advances in Neural Information Processing Systems 29. NIPS 2016

81. Daunizeau J, Stephan KE, Friston KJ. Stochastic dynamic causal modelling of fMRI data: Should we care about neural noise? Neuroimage. 2012; 2: 464–481.

82. Huys QJM, Paninski L. Smoothing of, and Parameter Estimation from, Noisy Biophysical Recordings. PLoS Comput Biol. 2009; 5: e1000379. https://doi.org/10.1371/journal.pcbi.1000379 PMID: 19424506

83. Durstewitz D. Advanced Data Analysis in Neuroscience: Integrating statistical and computational models. Heidelberg: Springer; in press.

84. Wang XJ. Neural dynamics and circuit mechanisms of decision-making. Curr Opin Neurobiol. 2012, 22: 1039–1046. https://doi.org/10.1016/j.conb.2012.08.006 PMID: 23026743

85. Insabato A, Pannunzi M, Deco G. Multiple Choice Neurodynamical Model of the Uncertain Option Task. PLoS Comput Biol 2017, 13: e1005250. https://doi.org/10.1371/journal.pcbi.1005250 PMID: 28076355

86. Rigotti M, Barak O, Warden MR, Wang XJ, Daw ND, Miller EK, Fusi S. The importance of mixed selectivity in complex cognitive tasks. Nature. 2013, 497: 585–590. https://doi.org/10.1038/nature12160 PMID: 23685452

87. Stephan KE, Kasper L, Harrison LM, Daunizeau J, den Ouden HE et al. Nonlinear dynamic causal models for fMRI. Neuroimage. 2008; 42: 649–662. https://doi.org/10.1016/j.neuroimage.2008.04.262 PMID: 18565765

88. Toth BA, Kostuk M, Meliza CD, Margoliash D, Abarbanel HD. Dynamical estimation of neuron and network properties I: variational methods. Biol Cybern. 2011; 105: 217–237. https://doi.org/10.1007/s00422-011-0459-1 PMID: 21986979

89. Kostuk M, Toth BA, Meliza CD, Margoliash D, Abarbanel HD. Dynamical estimation of neuron and network properties II: path integral Monte Carlo methods. Biol Cybern. 2012; 106: 155–167. https://doi.org/10.1007/s00422-012-0487-5 PMID: 22526358

90. Whiteway M. R., Butts D. A. Revealing unobserved factors underlying cortical activity with a rectified latent variable model applied to neural population recordings. J Neurophysiol 2017, 117: 919–936 https://doi.org/10.1152/jn.00698.2016 PMID: 27927786

91. Yi Z, Tan KK, Lee TH. Multistability analysis for recurrent neural networks with unsaturating piecewise linear transfer functions. Neural Comput. 2003; 15: 639–662. https://doi.org/10.1162/089976603321192112 PMID: 12620161

92. Tang HJ, Tan KC, Zhang W. Analysis of cyclic dynamics for networks of linear threshold neurons. Neural Comput. 2005; 17: 97–114. https://doi.org/10.1162/0899766052530820 PMID: 15563749

93. Yu J, Yi Z, Zhang L. Representations of continuous attractors of recurrent neural networks. IEEE Trans Neural Netw. 2009; 20: 368–372. https://doi.org/10.1109/TNN.2008.2010771 PMID: 19150791

94. Zhang L Yi Z, Yu J. Multiperiodicity and attractivity of delayed recurrent neural networks with unsaturating piecewise linear transfer functions. IEEE Trans Neural Netw. 2008; 19: 158–167. https://doi.org/10.1109/TNN.2007.904015 PMID: 18269947

95. Ruder S. An overview of gradient descent optimization algorithms. arXiv:1609.04747, 2016.

96. Mandic DP, Chambers JA. Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability 2001 Wiley.

97. Zipser D, Kehoe B, Littlewort G, Fuster J. (1993) A spiking network model of short-term active memory. J Neurosci. 1993; 13: 3406–3420. PMID: 8340815

98. Duchi J, Hazan E, Singer Y: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research 2011, 12:2121–2159.

99. Fuster JM. Prefrontal Cortex. 5th ed. Academic Press; 2015.

100. Fuster JM. Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory. J Neurophysiol. 1973; 36: 61–78. PMID: 4196203

101. Funahashi S, Bruce CJ, Goldman-Rakic PS. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. J Neurophysiol. 1989; 61: 331–349. PMID: 2918358

102. Miller EK, Erickson CA, Desimone R. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. J Neurosci. 1996; 16: 5154–5167. PMID: 8756444

103. Nakahara H, Doya K. Near-saddle-node bifurcation behavior as dynamics in working memory for goal-directed behavior. Neural Comput. 1998; 10: 113–132. PMID: 9501506

104. Baeg EH, Kim YB, Huh K, Mook-Jung I, Kim HT, Jung MW. Dynamics of population code for working memory in the prefrontal cortex. Neuron. 2003; 40: 177–188. PMID: 14527442

105. Mongillo G, Barak O, Tsodyks M. Synaptic theory of working memory. Science. 2008; 319: 1543–1546. https://doi.org/10.1126/science.1150769 PMID: 18339943

106. Fahrmeir L, Tutz G. Multivariate Statistical Modelling Based on Generalized Linear Models. Springer; 2010.

107. Eaves BC. "Solving Piecewise Linear Convex Equations," Mathematical Programming, Study 1, November; 1974. pp. 96–119.

108. Eaves BC, Scarf H. The solution of systems of piecewise linear equations. Math Oper Res. 1976; 1: 1–27.

109. Cottle RW, Dantzig GB. Complementary pivot theory of mathematical programming.Linear Algebra Appl. 1968; 1: 103–125.

110. Crisan D, Doucet A. A Survey of Convergence Results on Particle Filtering Methods for Practitioners. IEEE Trans Signal Process. 2002; 50: 736–746.

111. Lee A, Whitley N. Variance estimation in the particle filter. arXiv:1509.00394v2

112. Hyman JM, Ma L, Balaguer-Ballester E, Durstewitz D, Seamans JK. Contextual encoding by ensembles of medial prefrontal cortex neurons. Proc Natl Acad Sci USA. 2013; 109: 5086–5091.