Continuous Attractor Networks for Laplace Neural Manifolds

Bryan C. Daniels[1*] and Marc W. Howard[2*]

[1]School of Complex Adaptive Systems, Arizona State University, PO Box 872701, Tempe, 85287, AZ, USA.
[2*]Department of Psychological and Brain Sciences, Boston University, 610 Commonwealth Ave, Boston, 02215, MA, USA.

*Corresponding author(s). E-mail(s): bryan.daniels.1@asu.edu; marc777@bu.edu;

Abstract

Many cognitive models, including those for predicting the time of future events, can be mapped onto a particular form of neural representation in which activity across a population of neurons is restricted to manifolds that specify the Laplace transform of functions of continuous variables. These populations coding Laplace transform are associated with another population that inverts the transform, approximating the original function. This paper presents a neural circuit that uses continuous attractor dynamics to represent the Laplace transform of a delta function evolving in time. One population places an edge at any location along a 1-D array of neurons; another population places a bump at a location corresponding to the edge. Together these two populations can estimate a Laplace transform of delta functions in time along with an approximate inverse transform. Building the circuit so the edge moves at an appropriate speed enables the network to represent events as a function of log time. Choosing the connections appropriately within the edge network make the network states map onto Laplace transform with exponential change as a function of time. In this paper we model a learned temporal association in which one stimulus predicts another at some fixed delay T. Shortly after t=0 the first stimulus recedes into the past. The Laplace Neural Manifold representing the past maintains the Laplace transform $\exp(-st)$. Another Laplace Neural Manifold represents the predicted future. At t=0, the second stimulus is represented a time T in the future. At each moment between 0 and T, firing over the Laplace transform predicting the future changes as $\exp[-s(T-t)]$. Despite exponential growth in firing, the circuit is robust to noise, making it a practical means to implement Laplace Neural Manifolds in populations of neurons for a variety of cognitive models.

Many authors have proposed that the evolutionary adaptation of the brain is to predict the future (Clark, 2013; Friston, 2010; Friston & Kiebel, 2009; Palmer, Marre, Berry, & Bialek, 2015; Rao & Ballard, 1999). Our ability to predict the future depends on our ability to remember the past. A growing body of work has demonstrated that the nervous system of many animals uses a Laplace transform, with real s, to construct a temporal memory of the recent past (Atanas et al., 2023; Bright et al., 2020; Cao, Bright, & Howard, 2024; Tsao et al., 2018; Zuo et al., 2023). It has been hypothesized that populations of neurons representing the Laplace transform of functions of time (and other variables) are paired with another population that represents the inverse Laplace transform (Shankar & Howard, 2010). We refer to these paired representations of a continuous function as a *Laplace Neural Manifold*.

In addition to memory for the past, it has also been proposed that the brain uses a real Laplace transform to represent the time of predicted *future* events (Howard, Esfahani, Le, & Sederberg, 2023). Some neural evidence suggests that this could be the case (Affan et al., 2024; Cao et al., 2024, Fig. 1). However, constructing a neural circuit that represents the Laplace transform of the time of a predicted future event raises significant computational challenges that do not arise in representing the remembered past. Similar computational challenges arise in building realistic neural models for Laplace representations of spatial variables during allocentric navigation **?** and evidence accumulation Howard, Luzardo, and Tiganj (2018), so that a solution to this problem could have applications in many domains of computational cognitive neuroscience.

This paper addresses this challenge by demonstrating that continuous attractor neural networks are well suited to construct and maintain a neural population that represents the Laplace transform of future events. This proposal relies on two assumptions. First, the network is restricted to represent the Laplace transform of delta functions that can change in their location in time. Specifically, the Laplace transform of a delta function centered at τ is $e^{-s\tau}$. Second, the values of s across the population are distributed uniformly as a function of log time. That is, s_n, the rate constant for the nth neuron, scales as a^n for some constant a so that the change in $\log s$ for neighboring neurons is constant across all neurons. This choice for the distributions is consistent with a large body of work in psychology and neuroscience (Fechner, 1860/1912; Feigenson, Dehaene, & Spelke, 2004; Gallistel & Gelman, 2000; Van Essen, Newsome, & Maunsell, 1984) and is consistent with empirical findings for time constants in at least some brain regions (Cao, Bright, & Howard, 2023; Guo, Huson, Macosko, & Regehr, 2021). When rate constants s are chosen in this way, when τ changes to $c\tau$ the Laplace transform simply translates across the population of cells by an amount $\log_a c$ (Figure 2). For a delta function, translation in time also maps onto rescaling in time. If we translate in time by taking $\tau \rightarrow \tau + b$, this is equivalent to rescaling time $\tau \rightarrow c\tau$ with $c = \frac{\tau+b}{\tau}$. Thus, the time evolution of Laplace transform of a delta function can be represented as translation across a continuous attractor network if the rate of translation changes as time passes.

One extremely simple way to compute the Laplace transform of the past is to construct a bank of leaky integrators each with s chosen from a continuous spectrum.

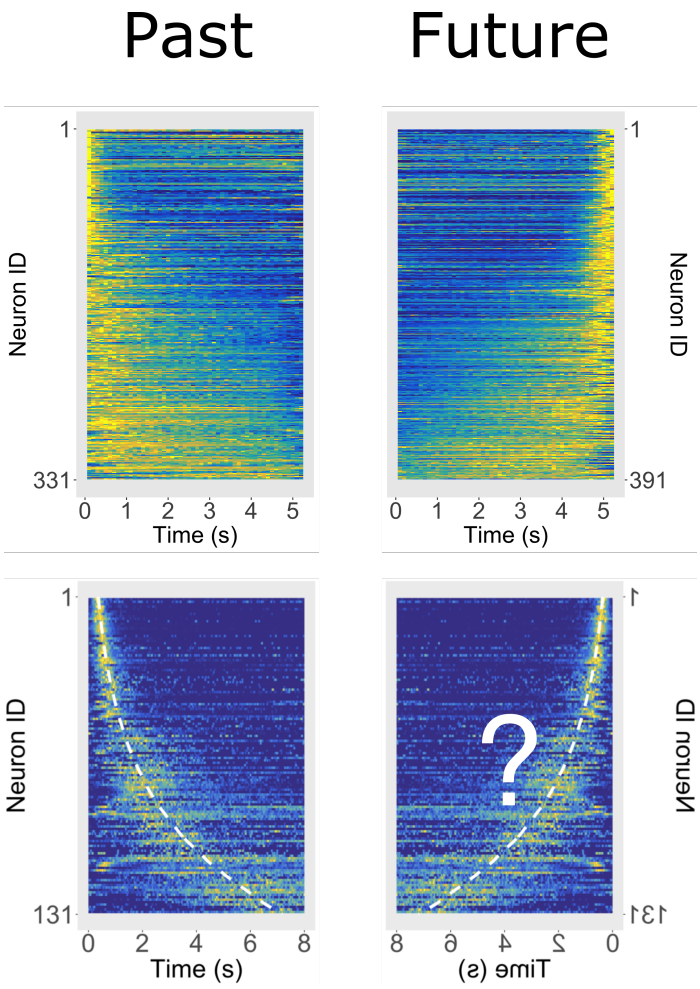

Fig. 1 Neural representations of the past (left) and future (right) in the brain. Top: Laplace transform of future and past in mPFC. Recordings are during an interval reproduction task in which the animal reproduces an interval of duration T. One group of cells decays exponentially e^{-st} with time during the reproduction phase (left). These neurons peak at about the same time but have a continuous spectrum of decay rates. This population codes for the Laplace transform of a delta function a time t in the past. Another population (right) ramps up as $e^{-s(T-t)}$ during the reproduction phase with a continuous spectrum of ramp rates. This group codes the Laplace transform of a delta function a time $T-t$ in the future. Data from Henke, et al., (2021). Analyses by Cao, et al., (2024). **Bottom: Inverse Laplace transform of the past in hippocampus.** Hippocampal time cells (left) fire in sequence following a task relevant stimulus. The sequence is understandable as an approximation of the inverse Laplace transform of time since the sequence began. After Cao, et al., 2023. Populations of neurons coding for the inverse of the future (right) would fire in in sequence approaching the predicted time of occurrence. This population is hypothesized, so the figure is simply a reflection of the data on the left.

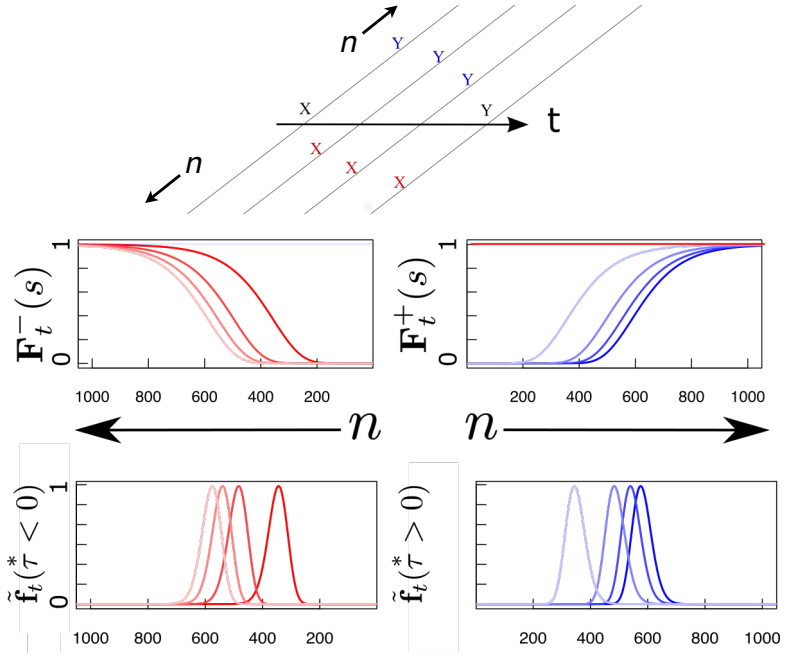

Fig. 2 Laplace Neural Manifolds representing delta functions can be implemented using continuous attractor neural networks. Top: Memory of the past and prediction of the future as a logarithmic timeline. Suppose that X and Y are experienced consistently with a lag of T seconds. As time unfolds (horizontal line) after presentation of X at $t = 0$, the observer will have a memory for the past (red, below the line) and prediction of the future (blue above the line). Because of the logarithmic compression of the timeline, the movement of the remembered/anticipated times do not change evenly as time passes. Middle: The function $e^{-s_n t}$ for evenly spaced choices of τ (red curves) as a function of $n \propto \log s$. If we assume a log distribution of rate constants, such that $s_n = s_o a^n$ and $n \propto \log s$, then evolving $e^{-s_n t}$ simply amounts to translation. At time $t = 0$, the future event is predicted a time T in the future. At time $t < T$, the future event is predicted a time $T - t$ in the future. Evolving $e^{-s(T-t)}$ simply requires moving the edge in the opposite direction (blue curves). Bottom: A bump that translates as a function of time attached to the edge is understandable as the inverse transform of the original function over log time convolved with some function of log time that describes the shape of the bump.

The dynamics for each integrator obey

$$\frac{dF}{dt} = -sF + f(t). \tag{1}$$

Let us use the notation $f_t(\tau)$ to refer the past leading up to the present at time t with $\tau = \infty$ referring to the distant past. The solution of Equation 1 is

$$F_t(s) = \int_0^\infty e^{-s\tau} f_t(\tau) d\tau \tag{2}$$

from which we can see that Eq. 1, with continuous values of s across neurons describes a population that codes for the real Laplace transform of the past. If at some moment

in time, the past consists only of a delta function at one moment τ in the past, then $F_t(s) = e^{-s\tau}$. Suppose that the stimulus is experienced at $t = 0$. As time passes, the stimulus will be $\tau = t$ in the past and each neuron will decay exponentially as e^{-st}. Now consider a population of neurons coding for the Laplace transform of a future event. Let us suppose that at $t = 0$ the event is predicted T seconds in the future. At any moment $0 < t < T$, the firing rate across the population represents a time $\tau = T - t$. As a function of time, the firing rate for neurons representing the time of a future event should be $e^{-s(T-t)}$. For this population, firing should ramp up exponentially as a function of time with a continuous spectrum of s. Naturally the exponential growth cannot continue indefinitely so at time $t = T$ we should observe a discontinuous change as either the predicted stimulus is observed or the prediction is violated.

The leaky integrators in Equation 1 could be implemented neurobiologically using one of many possible mechanisms to affect the functional time constant of individual neurons. Potential mechanisms include slow intrinsic currents (Fransén, Tahvildari, Egorov, Hasselmo, & Alonso, 2006; Tiganj, Hasselmo, & Howard, 2015), network recurrence (Dahmen, Grün, Diesmann, & Helias, 2019; Stern, Istrate, & Mazzucato, 2023), or synaptic mechanisms with a diversity of time constants (Barri, Wiechert, Jazayeri, & DiGregorio, 2022; Guo et al., 2021; Yoshida, Fransén, & Hasselmo, 2008). In principle one could adapt Equation 1 to code for the future by initializing the population to e^{-sT} then updating with Equation 1 but with positive s. However, this approach is unacceptable. Exponential growth implies that the population would be unstable in the presence of any amount of noise.

1 Continuous attractor model for Laplace/inverse representations

The primary contribution of this paper is demonstrating that continuous attractor networks can be used to evolve $F_t(s)$ in time if two conditions are met. First, at any particular moment we only need to represent the Laplace transform of a delta function. Second, the values of s_n are in a geometric series. If these two conditions are met, then evolving $F_t(s)$ in time amounts to translating an edge of differential activation across the population. Continuous attractor neural networks are well-suited to this problem.

To see why these conditions are sufficient, consider how the solution to Equation 1 changes between time t and $t + \Delta t$ (Figure 2). In general, $F_{t+\Delta t}(s) = e^{-s\Delta t}F_t(s)$. Let the population code a delta function a time τ in the past so that $F_t(s_n) = e^{-s_n\tau}$. Now, $F_{t+\Delta t}(s_{n+\Delta n}) = e^{-s_{n+\Delta n}(\tau+\Delta t)}$; if we want the time evolution to be a simple translation by Δn, we require that $\frac{s_{n+\Delta n}}{s_n} = \frac{\tau}{\tau+\Delta t}$. If Δn is a constant as a function of n, then we can say that the activity has translated across the population with the passage of time. Δn is a constant iff the s are in geometric series. Note that the magnitude of the translation along n depends not only on Δt, but on τ as well. Defining the ratio between adjacent s as $a = \frac{s_{n+1}}{s_n}$, we find that $\Delta n = \log_a \frac{\tau}{\tau+\Delta t}$.

The continuous attractor network proposed here consists of two populations. The population that represents the Laplace transform is an attractor network with interactions that favor neighbors in the same state of activation and the two ends of the array

clamped to different values. This network exhibits a stable "edge" at any point along the array. With appropriate choice of interactions this network maps precisely onto the Laplace transform of a delta function with values of the rate constant chosen in a geometric series. The "edge attractor" network representing the Laplace transform is coupled to a "bump" network that takes as input the derivative of the edge attractor network, placing the bump at a location that corresponds to the location of the edge. With appropriate feedback coupling between the networks, the bump network pushes the edge along at a pace that enables the location of the edge to be mapped onto $\log \tau$.

Consider the activity of an individual cell at a particular location in the middle in the bump network. Initially, the cell's firing is low. As the bump approaches the cell's location, its firing increases. As the bump moves past the cell's location its firing rate decreases again. As the bump moves monotonically from one end of the network to the other, each cell is activated in this way at a time that depends on their location in the network and the rate at which the bump moves. These sequentially activated cells have behavior that resembles that of "time cells" (Figure 1, bottom left). Time cells were initially reported in hippocampus (Cao et al., 2023; MacDonald, Lepage, Eden, & Eichenbaum, 2011; Pastalkova, Itskov, Amarasingham, & Buzsaki, 2008) and subsequently reported in many other brain regions (Jin, Fujii, & Graybiel, 2009; Mello, Soares, & Paton, 2015; Tiganj, Cromer, Roy, Miller, & Howard, 2018). If time cells result from movement of a bump of activity across a continuous attractor network then there is no mechanistic obstacle to constructing "future time cells" (Figure 1, bottom right) by simply having the bump travel in the opposite direction. These "future time cells" have not been characterized in the hippocampus (although spatial predictive sequences have been observed (Ferbinteanu & Shapiro, 2003; Gauthier & Tank, 2018; Sarel, Finkelstein, Las, & Ulanovsky, 2017)). There is some evidence for predictive sequences in other brain regions, including cerebellum (Wagner, Kim, Savall, Schnitzer, & Luo, 2017), although the empirical story is not nearly as clear as for standard time cells.

1.1 Overview of the circuit model

To generate the Laplace transform of a delta function using neurons with logarithmically distributed rate constants we require 1) a continuous attractor neural network with an edge attractor with the correct shape and 2) translation of the edge over the network in time with a speed $d\bar{n}/dt$ proportional to $1/\tau$. Inverting the transform can be accomplished with another continuous attractor neural network that places a bump of activity in an analogous location as the edge. Because the location of the edge is understandable as log time, so too the bump also translates as a function of log time. Because the bump should have the same shape at any moment in the past, neurons in the bump network will naturally inherit scale-invariant properties observed for hippocampal time cells (Cao, Bladon, Charczynski, Hasselmo, & Howard, 2022). Because the bump has the same location as the edge, the pattern over the bump network is understandable as the original function over log time, convolved with a function of log time that describes the shape of the bump. In this sense, the state over the bump network approximates the inverse Laplace transform, but as a function of logarithmic time.

We want an edge attractor to approximate $\mathcal{L}\{f\}(s)$ for delta functions $f(t-\tau)$, with real values of s sampled along a log scale. Our strategy will be to use two sets of neurons (Fig. 3) labeled by index n. We will treat n as continuous and align the two populations with one another. We will design the connections within and between these two sets of neurons $x(n)$ and $y(n)$ so that their stable states are simply related to the Laplace transform of f and the inverse transform respectively. This paper builds a minimal circuit to accomplish these goals and does not attempt a computationally optimal nor biologically detailed model.

The real Laplace transform of a delta function at a point on the positive line takes values between zero and one. The continuous attractor neural network $x(n)$ goes from $-x_{\max}$ to $x_{\max}$. We provide external input to clamp the two edges of the network at $n = 1$ and $n = N$ to these extreme values. Ferromagnetic interactions between units within $x(n)$ encourage neighboring units to have the same activation. These ferromagnetic interactions are implemented by connections w_r; the weights on the connections between units within x depend only on their distance in the network, $n-n'$. If the parameters of the network are chosen such that it resembles a ferromagnet at low temperature, then somewhere between the ends of the network, there should be a transition between maximal and minimal values. This boundary, or edge, can be at any location along the network as long as the range of the local interactions is far from the ends $n = 1$ and $n = N$. The requirement that the neurons code real Laplace transform requires a specific shape of the edge, which in turn requires a specific form for connection weights.

In this paper we adopt a minimal approach to the bump network. Rather than recurrent connections within $y(n)$ we simply take input to neurons in $y(n)$ to be large if there is an edge in the neighborhood of each n.

The input to $y(n)$ from $x(n)$ is provided by w_d, which simply computes the derivative of $x(n)$ in the neighborhood of n. Again, the connections between $x(n)$ and $y(n')$ depend only on $n - n'$. The bump then provides feedback to the $x(n)$ via connections $v(n)$, causing the edge to move over time. By choosing $v(n)$ to depend on n appropriately we can control the rate at which the edge moves at different points along the network.

1.2 Continuous attractor neural network with edge solutions

To define the neural dynamics, we use a simple neural firing rate model:

$$\frac{dx(n)}{dt} = -x(n) + I(n) + \sum_{n'} w_r(n-n')h\left[\frac{x(n')}{\sigma_r}\right] + w_f(n-n')h\left[\frac{y(n')}{\sigma_h}\right] + \xi \quad (3)$$

$$\frac{dy(n)}{dt} = -y(n) + \sum_{n'} w_d(n-n')h\left[\frac{x(n')}{\sigma_h}\right] + \xi. \quad (4)$$

Here x and y represent instantaneous relative neural firing rates. $I(n)$ consists of a large negative input current $-i$ applied from $n = 1$ to n_i and large positive input current $+i$ applied at the other end of the network from $n = N$ to $N - n_i$, which sets boundary conditions so that the edge has an orientation with small n neurons

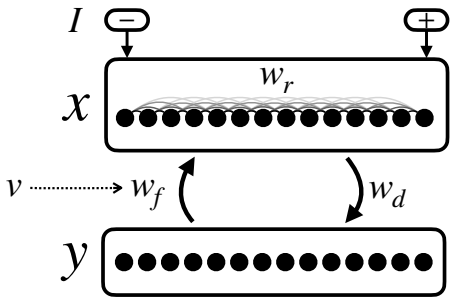

Fig. 3 Schematic of network setup. The edge attractor is distributed over one set of units $x(n)$, utilizing recurrent connections w_r within this set of units. Another set of units $y(n)$ supports a bump. Connections w_d from x provide input to y. These connections estimate the derivative with respect to n. Connections from x to y, w_f, move the location of the edge as a function of time. Because the strengths of these connections are a function of n, the velocity of the edge v depends on the location of the edge.

saturated at $-x_{\max}$ and large n neurons saturated at $+x_{\max}$. The kernel w_r defines connections among x neurons with varying n and determines the shape of the edge. The kernel w_d defines connections from x to y, and is set to implement an effective derivative of $x(n)$ being represented in $y(n)$. The kernel w_f defines feedback from y neurons to x neurons. The nonlinear function $h = \tanh$ defines the mapping from neural state to synaptic current, with width σ_r among x neurons, and σ_h for all other synaptic connections. Uncorrelated Gaussian noise is represented by ξ, parameterized such that variance is added at a rate σ_ξ^2.

In simulations, we initialize the neural states to a stationary state $x^*(n)$, $y^*(n)$, found numerically by setting feedback w_f and noise σ_ξ to zero, with the edge and bump set initially at location $\bar{n}_0$. Translation equivariance of w_r, w_d, and w_f implies that the edge can smoothly translate to any location $\bar{n}$ along the network. In implementing cognitive functions, one might imagine using input to set the initial location of the edge. Once the network is in a stable edge state, this input can be removed.

1.3 Mapping edge solutions to the real Laplace transform

We desire that there be a mapping from neuron index n to the Laplace variable s, $s(n)$. At any moment, the Laplace transform of a delta function at time τ gives $F(n) = e^{-s(n)\tau}$. Therefore, if the network has successfully represented Laplace transform of a delta function, we should be able to observe a particular unit as time evolves and see exponential changes in firing with rate constant $s(n)$. To ensure the time evolution of the network is simply translation of the edge, we require that $s(n) \propto a^n$ for some constant a. Equivalently, we require $n = \log_a \frac{s}{s(n=1)}$, so that the $s(n)$ form a logarithmic time scale for location of the edge. Thus we set the size of the scale factor between s of neighboring neurons to be a, with $a < 1$. In the limit that $a \to 1$, we get a continuum limit in which neighboring neurons represent infinitesimally distinct s.

The dynamics defined by Equation 3 produce x varying between $-x_{\max}$ and $+x_{\max}$. The Laplace transform of a delta function with $\tau > 0$ should vary between 0 and 1,

so we define

$$F(s) = \frac{1}{2}\left[\frac{x(s)}{x_{\max}} + 1\right]. \tag{5}$$

To get the stable states as a function of n to be identified with Laplace transform of a delta function, we need the edge to have the right shape and for the edge to move at the proper speed. Given that those problems have been solved (see below), the activity over $y(n)$ will map onto the inverse Laplace transform as long as the bump is located at an analogous n and does not change shape as the edge moves along n. If these conditions are met, then the bump state is understandable as the convolution of the delta function f as a function of $\log \tau$ and some function of $\log \tau$ that describes the shape of the bump over n.

1.4 Setting the interaction kernels

We will first set aside the problem of defining w_r, which sets the equilibrium shape of the edge, and merely assume that $x(n)$ is monotonically increasing in n, with an edge location defined as the (interpolated) $\bar{n}$ at which $x(\bar{n}) = 0$. To get such an edge state to move, we will first approximate the derivative of x with respect to n in the states of the y neurons, then use $y(n)$ to drive the edge at the desired speed.

1.4.1 Forming a bump in y

First, we want $y(n)$ to represent a (discrete) derivative of $x(n)$:

$$y(n) \to \frac{1}{2}\left[x\left(n+1\right) - x\left(n-1\right)\right]. \tag{6}$$

In equilibrium assuming constant input and neglecting noise, from Eq. 4,

$$y(n) \to \sum_{n'} w_d\left(n-n'\right) h\left[x\left(n'\right)/\sigma_h\right]. \tag{7}$$

If we set the scale of the nonlinearity σ_h large enough, then $h\left[x\left(n'\right)/\sigma_h\right] \approx x(n')/\sigma_h$, and we can implement the derivative simply by setting

$$w_d(n-n') = \frac{\sigma_h}{2}\left[\delta\left(n-n'-1\right) - \delta\left(n-n'+1\right)\right]. \tag{8}$$

1.4.2 Moving the edge to form a logarithmic time scale

With neural states $y(n)$ representing the derivative of the edge as a function of n (Eq. 6), this can be rescaled and applied as an input to the x neurons to move the edge. We then define w_f to feed y back to x with a scaling factor v that can depend on n:

$$w_f(n-n') = -\sigma_h v(n)\,\delta(n-n'). \tag{9}$$

A constant feedback strength $v(n) = v_c$ moves the edge with constant speed $\frac{d\bar{n}}{dt} = v_c$. To see this, consider solving for the input necessary to move an equilibrium edge with shape $x^*(n)$ a small distance $\Delta\bar{n}$ in time Δt. Since there is a continuous set of steady state edges with shape $x^*(n)$, the needed change in x as a function of n is $\Delta x = x^*(n - \Delta\bar{n}) - x^*(n) \approx -\Delta\bar{n}\frac{dx^*(n)}{dn}$. Then near equilibrium the necessary input term added to $\frac{dx}{dt}$ to produce motion with speed $v_c = \frac{\Delta\bar{n}}{\Delta t}$ is $-v_c\frac{dx^*(n)}{dn}$. Finally, with $y(n) \approx dx^*(n)/dn$, $h[y(n')/\sigma_h] \approx \sigma_h dx^*(n)/dn$, so we need $w_f = -\sigma_h v_c \delta(n - n')$.

Instead of having the edge move with constant speed, we want $\bar{n}$ to grow logarithmically in time:

$$\bar{n}(t) = \bar{n}_0 + \log_a \frac{t}{t_0}, \tag{10}$$

corresponding to edge speed

$$\frac{d\bar{n}}{dt} = \frac{1}{\ln a}\frac{1}{t}. \tag{11}$$

As a function of $\bar{n}$, this corresponds to

$$\frac{d\bar{n}}{dt} = \frac{1}{\ln a}\frac{1}{t_0}a^{-(\bar{n}-\bar{n}_0)} \tag{12}$$

because $t(\bar{n}) = t_0 a^{(\bar{n}-\bar{n}_0)}$.

If we make the assumption that the bump is sufficiently localized in n such that we can replace variations in $v(n)$ along the bump with the value at the edge location $v(\bar{n})$, then we get the desired speed by setting

$$v(n) = v_0\, a^{-(n-\bar{n}_0)} \tag{13}$$

where we have replaced $\bar{n}$ by n in the desired $\frac{d\bar{n}}{dt}$ and set the initial speed $v_0 = \frac{1}{t_0}\frac{1}{\ln a}$.

In our simulations, we must limit the size of v to avoid an instability that arises due to the feedback loop from edge to bump to edge neurons. We therefore limit v to a predefined $v_{\max}$. This does not affect the edge motion as long as the range of n over which the limitation acts does not significantly overlap the edge locations.

1.4.3 Reversing velocity of the edge and bump to represent events as a function of future time

To change the direction in which the edge moves, we can simply flip the sign of the desired velocity $v(n)$. Note that the dependence of the initial velocity v_0 on t_0 means that the edge moves in the correct direction when $t_0 > 0$ corresponds to representing past events, with the edge moving toward larger n, and $t_0 < 0$ corresponds to representing future events, with the edge moving toward smaller n. If at time $t = 0$ the future event is predicted at future time T, then at time $0 < t < T$, the $x(n)$ corresponds to Laplace transform of a delta function at $\tau = T - t$. In parallel, the network representing the past represents a delta function at $\tau = t$.

In our example simulations, we demonstrate the representation of an event in the past (at $\tau = 0$) that is going further in the past by setting $t_0 = 25$ and letting time t run forward to $t = 175$. We demonstrate the representation of an event predicted

to happen in the future (also at $\tau = 0$) by setting $t_0 = -175$ and letting time t run forward to $t = -25$.

1.5 Setting the shape of the edge to ensure time course of each unit in $x(n)$ is exponential

Finally, we set the edge shape by solving for the form of interactions w_r among x neurons. Note that an edge of any shape that moves with speed $\propto 1/t$ corresponds to x decaying in time for a given n at a rate $s(n)$ that decreases geometrically with n:

$$s(n) = \frac{1}{t_0\, a^{(n-\bar{n}_0)}}. \tag{14}$$

Here we set the constant prefactor by stipulating that $s(n = \bar{n}(t)) = t^{-1}$ for all t. Specifically, the neuron corresponding to the starting location $n = \bar{n}_0$ has rate constant $s = t_0^{-1}$. The shape of the decay of a given neuron's state with respect to time will be some function

$$x(t, n) = \mathcal{F}\left[s(n)\, t\right]. \tag{15}$$

$\mathcal{F}(s\, t)$ will be set by the shape of the edge as a function of n, which is in turn controlled by w_r. For $x(n)$ to correctly represent the Laplace transform $F(s)$ when rescaled as in Eq. 5, we want the state of individual neurons to decrease exponentially as a function of time. As a function of t and n, we want the output to be

$$e^{-A\, s(n)\, t} = \exp\left[-A \frac{t}{t_0} a^{-(n-\bar{n}_0)}\right], \tag{16}$$

with a constant A that we set to $A = \ln 2$ to fix the edge location $\bar{n} = \bar{n}_0$ at time $t = t_0$. At a fixed time t with corresponding edge location $\bar{n}$ (setting $t = t_0$ and $\bar{n} = \bar{n}_0$), this corresponds to a desired functional shape of

$$F_d(n, \bar{n}) = \exp\left[-A a^{-(n-\bar{n})}\right]. \tag{17}$$

In terms of neural states, we need, from Equation 5

$$\frac{x_d(n, \bar{n})}{x_{\max}} = 2F_d(n, \bar{n}) - 1. \tag{18}$$

The equilibrium shape of the edge as a function of n is the solution of

$$x^*(n) = \sum_{n'} w_r(n - n') h\left[x^*\left(n'\right)/\sigma_r\right]. \tag{19}$$

using Eq. 3 and neglecting terms corresponding to feedback, noise, and inputs at the boundaries. We then want to solve Eq. 19 for w_r given $x^*(n) = x_d(n)$ as in Eq. 18.

N	100
n_0	50
σ_h	20
σ_r	0.5
a	$e^{0.25}$
$x_{\max}$	2
$v_{\max}$	1
i	100
n_i	5
Δt	10^{-1}
σ_ξ^2	10^{-3}
δn	0.1256

Table 1
Simulation parameters.

Note that, in the limit $\sigma_r \to 0$, the tanh nonlinearity becomes a step function. In this limit, and assuming $x^*(n)$ has a single zero crossing at $\bar{n}$, Eq. 19 produces

$$x^*(n+1) - x^*(n) = -2w_r(n - \bar{n}). \tag{20}$$

Thus we can get the desired edge shape by setting the interaction kernel proportional to the derivative of the desired edge shape with respect to n (Amari, 1977):

$$w_r(n - n') \propto -\frac{dF_d(n, n')}{dn} \propto a^{-(n-n')} F_d(n, n'). \tag{21}$$

The overall scale factor of the interaction kernel sets the scale of the equilibrium neural states, $x_{\max}$. To set $x_{\max}$ precisely in the discrete case, we numerically normalize the interaction kernel, resulting in

$$w_r(\Delta n) = x_{\max} \frac{\tilde{w}_r(\Delta n)}{\sum_{\Delta n} \tilde{w}_r(\Delta n)}, \tag{22}$$

where we define $\Delta n = n - n'$ and

$$\tilde{w}_r(\Delta n) = a^{-\Delta n} \exp\left[-A\, a^{-\Delta n}\right]. \tag{23}$$

As can be seen from Eq. 23, these weights are asymmetric (they are not even in Δn). The network will give an edge for many choices of weights; this expression, coupled with the velocity of the edge as specified in Eq. 13, gives exponential decay in time.

1.5.1 Simulation methods

Here we illustrate properties of the continuous attractor neural network model with simulations, integrating the dynamics using a simple Euler timestep Δt. In our simulation, setting $\sigma_r = 0$, i.e., using a step function nonlinearity instead of tanh, does successfully create the desired edge shape, but the infinitely sharp interactions effectively pin the edge to the lattice, not allowing the edge to move smoothly along n.

That is, the continuous nature of the attractor states is interfered with by the discreteness of the lattice. To mitigate this effect while still keeping close to the desired edge shape, we reduce σ_r as much as possible before pinning becomes strong. We find that $\sigma_r = 0.5$ is a good compromise, and that values between 0.15 and 0.5 produce relatively indistinguishable results.

A second complication due to discretization comes from the fact that setting w_r using the above logic, which assumes continuity, on discrete lattices leads to an edge that moves slowly over time even with zero input feedback. To fix this, we numerically solve for an offset value δn that produces zero change in the edge's location after one application of the offset interaction kernel $w_r(n - n' + \delta n)$. Given our other simulation values, we find $\delta n = 0.1256$.

2 Results

Figure 4 shows the activity over the network at different evenly-spaced moments in time. At $t = 0$, a stimulus that predicts a future event at time T is presented. At time $t > 0$ neurons in the $x(n)$ mapping onto the Laplace transform of the past represent a time $\tau = t$ in the past; neurons in the $x(n)$ mapping onto Laplace transform of the future represent a time $\tau = T - t$ in the future. The simulation of the continuous attractor network exhibits the desired properties and gives results closely analogous to Figure 2. With the passage of time the edge/bump changes its position but retains its shape. When τ is bigger, the edge/bump complex is further from $n = 1$. As time passes, the edge/bump representing the past moves more slowly whereas the edge/bump representing the future moves more rapidly.

Rather than looking at the pattern of activity across the population at different points in time, Figure 5 describes activity of the edge network as a function of time. First, as can be seen from the top row, the location of the edge, operationalized as the value of n where activity passes through zero, moves as desired (Equation 10) for both representations of the past (left) and the future (right). This is a consequence of the choice for $v(n)$. Second, following the time course of individual units in the edge network, we find exponential firing as a function of time with a rate constant controlled by $s(n)$. The bottom panel shows the time course of individual cells rescaled by each cell's value of s. It is clear that individual cells follow the same time course up to a scaling factor. This is a consequence of the logarithmic rate at which the edge progesses. The inset plots the firing as a function of time on a logarithmic scale. The nearly straight lines are consistent with an exponential time course, allowing a mapping between the activity of $x(n)$ and the real Laplace transform. Units participating in the representation of the future (right) grow exponentially.

Because the representation of the time of future events requires exponential growth, tt would be impossible to construct an estimate of the Laplace transform of the time of future events using independent leaky integrators as in Equation 1. With independent neurons, even a small amount of noise would be rapidly amplified so that the pattern of activity can no longer reliably code for the time until the predicted event. Figure 6 repeats the simulation in Figure 5 but in the presence of noise. As can be seen from visual inspection, the noise in the network is stable over time. The location of the edge

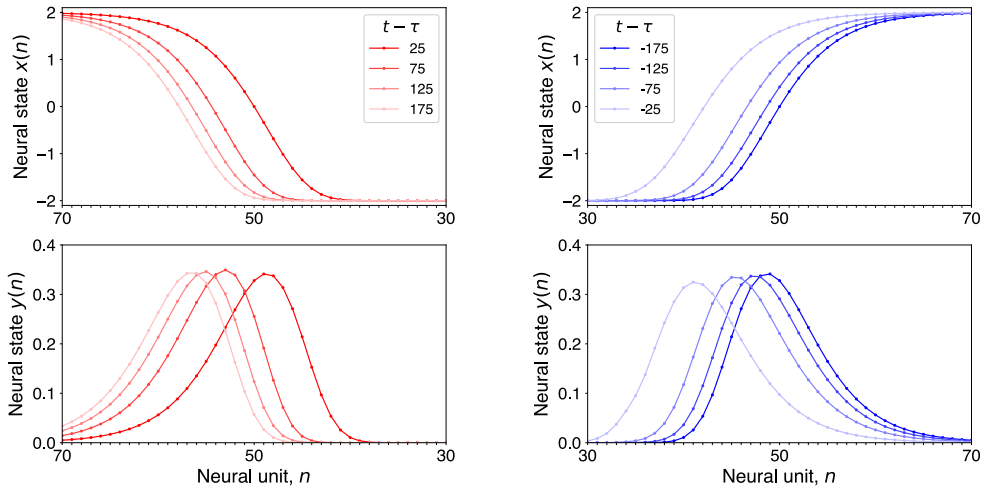

Fig. 4 Simulated dynamics of continuous attractor neural network model for Laplace transform of events in the past and the future. Layout is as in Figure 2. Top: Activation $x(n)$ over the network at four different time points. Bottom: Activation $y(n)$ at the same time points. Left: Networks representing the past $e^{-s\tau}$ at evenly-spaced τ. Note that the edges become closer together as events recede into the past. Right: Networks representing the future $e^{-s(T-\tau)}$ at evenly-spaced τ. Note that the edges become more widely separated as the event approaches from the future. Here noise $\sigma_\xi^2 = 0$, with other simulation parameters listed in Table 1

still moves as desired and the time course of individual neurons still follows exponential functions albeit with fluctuations around the mean. Critically, neurons representing the time until predicted future events grow smoothly even in the presence of noise.

The properties of the edge/bump networks in this paper are thus sufficient to describe the basic properties of time-sensitive neurons observed in the brain. Figure 7 plots the results of the same simulation in Figure 6 using the same conventions as used to show the neural data in Figure 1. In this plot, units are sorted as a function of n on the horizontal axis and the relative firing rate of each neuron is shown *via* color as a function of time. Neurons in x resemble the firing of exponentially decaying and ramping neurons in the mammalian brain (Cao et al., 2024). Neurons in y coding for the past (lower left) fire sequentially with a characteristic hook that resembles "time cells" observed in the hippocampus and other brain regions (Cao et al., 2022). The bump in the network associated with the future (lower right) provides a mirror image of the time cells. These hypothesized "future time cells" have more cells firing near the time of the predicted event and with narrower temporal receptive fields.

3 Discussion

This paper developed a simple continuous attractor neural network with stable states that form an edge at any position along a line of neurons. This attractor neural network is coupled to another set of units that express a bump at a location that aligns with the edge. The speed at which the edge moves is controlled by feedback

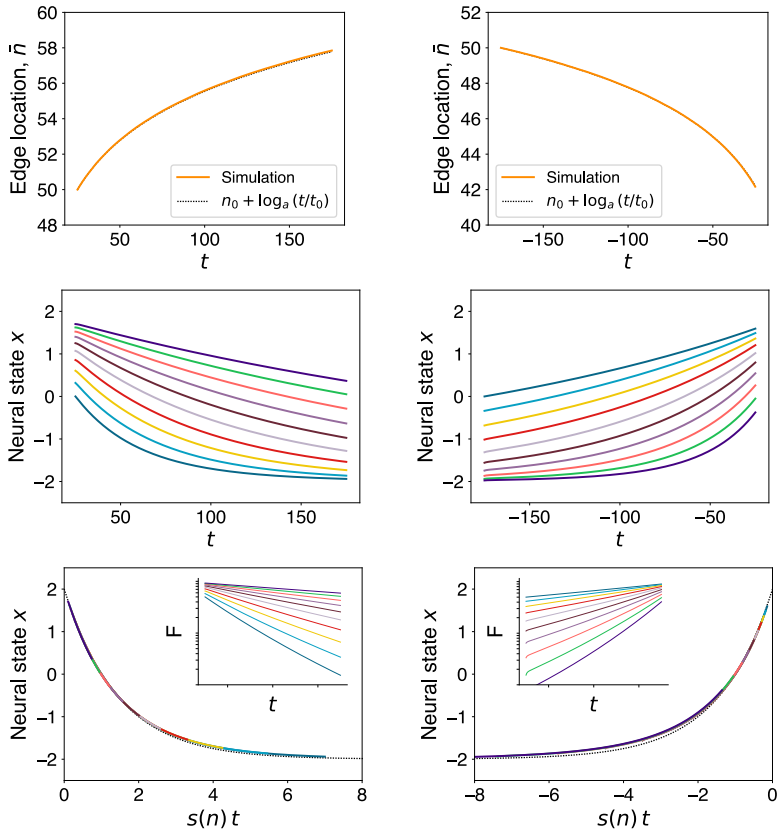

Fig. 5 Simulated dynamics of the continuous attractor neural network circuit model aligns with the theoretical properties of Laplace transform of the past and the future. Top: The location of the edge/bump as a function of τ moves as desired by the theory. Middle: Individual neurons in x decay/ramp exponentially in time, as required for the Laplace transform of a delta function. Bottom: Individual neurons in x decay/ramp according to the same function. The time dynamics of individual neurons in x differ only in their time scale. In this figure, noise $\sigma_\xi^2 = 0$.

from the bump neurons to the edge neurons. By allowing these connections to change as a function of position along the line, the edge moves at different speeds. Because we chose the velocity to be proportional to inverse time, the location of the edge/bump at any moment can be understood as the logarithm of elapsed time. Because the edge attractor has recurrent dynamics, this network is robust to noise. By changing the sign of the feedback signal we can allow the edge/bump to move in either direction.

The choices made in constructing the neural network were made to allow a mapping of the population $x(n)$ onto $\mathcal{L}\{f_t\}(s)$ where f_t is a delta function t seconds in the past and $s(n) \propto a^n$. This implies that the bump is interpretable as an approximation of the original function f projected onto log time. The bump is just f convolved with some function over log time describing the shape of the bump. While the precise shape of the connections within and between each layer affect the shape of this function,

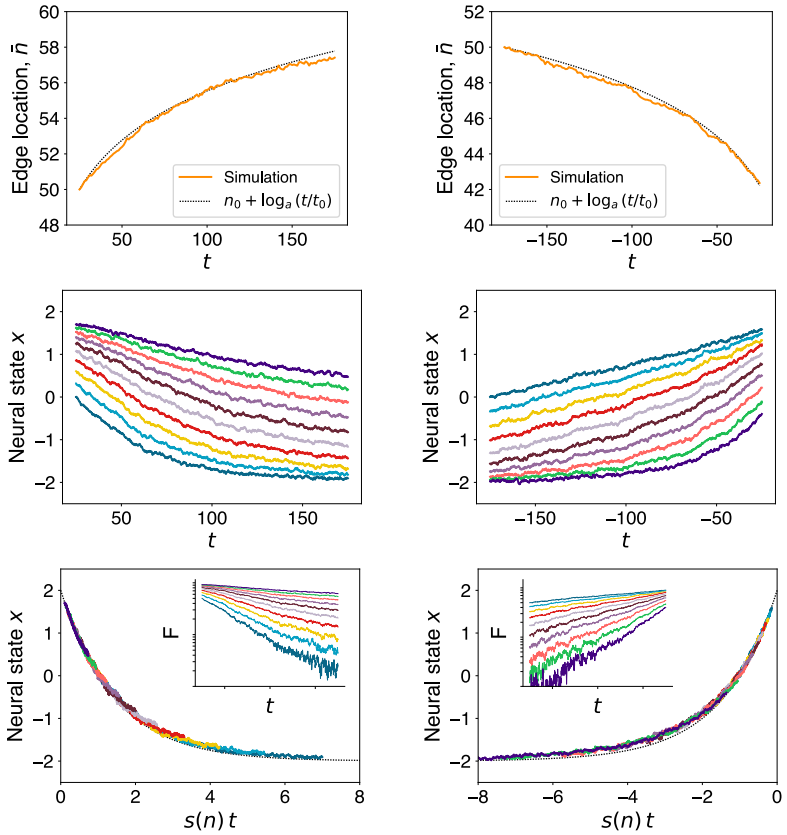

Fig. 6 Circuit model for Laplace transform is robust to noise. Layout analogous to Figure 5. Here, the noise parameter σ_ξ^2 was increased from 0 to 10^{-3}.

the interpretation of $y(n)$ as an approximation of the original function projected on log time depends only on the translation-equivariance of the stable states of $x(n)$ and the rate at which the edge moves as a function of time. This work allows a precise connection between abstract cognitive models using a Laplace Neural Manifold and circuit-level attractor neural network models.

This framework is sufficiently general to represent both memory for the past—delta functions f that move further from zero as time passes—and the future—delta functions f that approach zero as time passes. The function $v(n)$ controls the speed at which the edge/bump moves across the network. To represent the future the edge moves in the opposite direction and the sign of $v(n)$ is reversed. This results in units in $x(n)$ that *ramp upwards* exponentially in time with a variety of time constants (Figure 5). It would be extremely challenging to build a network with this property with independent units, as in Equation 1.

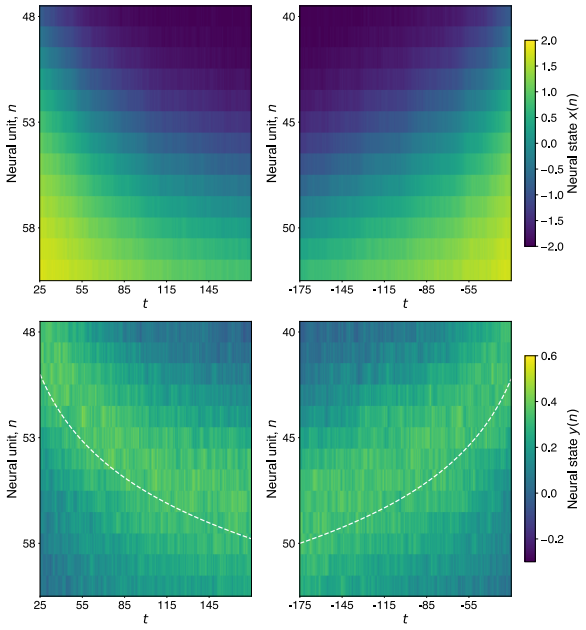

Fig. 7 Dynamics are similar to those observed in the brain. Layout is analogous to Figure 1. Top: The firing rate of each cell in an edge attractor network $x(n)$ as a function of time sorted on location in the network n. On the left, the edge moves from small n to large n with $v(n)$ positive. On the right, the edge moves from large n to small n with $v(n)$ negative. The cells in these networks emulate neurons coding for the Laplace transform of the past and the Laplace transform of the future respectively. Bottom: The activity of neurons in the bump network $y(n)$ as a function of time for the corresponding edge attractors. The neurons in these networks emulate time cells coding for the past (left) and hypothesized "future time cells" (right).

3.1 Limitations of this approach

Our attractor network acts as a good approximator of a Laplace transform within a certain limited regime. In particular, the approximation breaks down at small times $|t-\tau|$, near the singularity at $t = \tau$, due to the network's inability to support arbitrarily large time derivatives. In our setup, we limit the magnitude of $v(n)$ to $v_{\max}$, meaning that the edge velocity approaches a constant at $t \to \tau$ instead of diverging as in Eq. 11. This could correspond to a limitation in the ability to represent times in the very near past or future. At the other extreme, the approximation also breaks down at large $|t - \tau|$ because noise overwhelms the signal of small time derivatives. At large enough $|t - \tau|$, the edge would cease to move at the desired slow speed and would instead diffuse randomly. This could correspond to an inability to represent times in the very distant past or future.

3.2 Extensions of this approach

The present paper describes a minimal circuit to build the Laplace transform for remembered past and predicted future times. There are many possible elaborations

that would enhance its biological realism, computational stability, and application to a wider range of cognitive problems.

3.2.1 Symmetric weights; Alternatives to Laplace transform

In a continuous attractor network, recurrent excitatory connections are typically chosen to be symmetric as a function of distance within the network. In this paper we chose the weights w_r to give a shape of the edge that results in each unit's activation evolving in time as an exponential function. This choice required weights that are not precisely symmetric as a function of distance within the network (Eq. 22). Different weights would have resulted in a different-shaped edge and the time course of the individual neurons would no longer follow an exponential function. Although the activity over that network would no longer be directly related to the Laplace transform it would still be effective as a working memory representation over a logarithmic timeline.

The Appendix shows results analogous to Figures 4 and 5 computed with a network identical to the one in the main text except that the connections w_r are chosen to be a symmetric function of the distance between units in x, as is more typical in continuous attractor neural networks. The edge still moves in such a way that the location of the edge $\bar{n}$ changes with log time. This is a consequence of the velocity controlled by $v(n)$. With symmetric w_r, the time courses of individual units are still rescaled versions of one another (mapping onto a single scaling function as in Eq. 15) with a time constant that goes like $1/s_n$ (Figure 5 bottom). This means that the activity across $x(n)$ can still be understood as an integral transform of the delta function f.

The symmetric edge model described in the Appendix would be very difficult to distinguish empirically from the model that maps onto the Laplace transform described in the main text. The symmetric model results in $y(n)$ being understandable as a representation of the delta function f as a function of log time, although convolved with a slightly different bump shape. Cognitive models of retrieval time in working memory tasks rely on this logarithmic timeline moreso than the shape of the bump of activity (Howard, Shankar, Aue, & Criss, 2015; Tiganj, Tang, & Howard, 2021). The distribution of time constants that would be estimated from units in $x(n)$ with the symmetric model is identical to the distribution estimated from the Laplace model.

3.2.2 Including recurrent bump dynamics

A great deal of theoretical (Amari, 1977; Redish & Touretzky, 1997; Zhang, 1996) and empirical (Kim, Rouault, Druckmann, & Jayaraman, 2017; Taube, 1998) work has studied the properties of continuous bump attractor neural networks for use in neuroscience and psychology (Schöner & Spencer, 2016). In these networks, recurrent excitatory connections within a population of neurons fall off as a function of distance within the network. In addition, some form of inhibition, perhaps divisive normalization or global inhibition, places an upper limit on the activity of the network. These two constraints mean that the network can support stable states that take the form of "bumps" of activity across the population. Because the strength of the excitatory connections between neurons depends only on distance in the network, the stable bump can be localized at effectively continuous locations.

The activity over $y(n)$ showed bumps of activity. However, these did not depend on recurrent connections between the units in y but were inherited from input from the $x(n)$ *via* connections w_d, which computes the derivative of $x(n)$ with respect to n. Inclusion of recurrent weights within $y(n)$ would not have had a major impact on the properties of this network. We would expect such a network to be more resistant to noise as recurrent weights within both x and y would tend to stabilize the network. Because the bump attractor would be stable at all locations, inputs from $x(n)$ that estimate the derivative should place the bump attractor in the appropriate location. Although the shape of the bump would certainly change depending on the form of the recurrent weights within units in y, as long as the location of the bump coincides with the edge, activity over $y(n)$ is interpretable as the original function f over log time convolved with some function describing the shape of the bump attractor.

3.2.3 Dynamic control of edge/bump location

The speed at which the edge/bump complex moves in this model is controlled by $v(n)$. Choosing $v(n)$ to be exponentially decreasing in n allows the location of the edge/bump to be interpretable as log time. Suppose, however, that it were possible to modulate all of the connections by a time varying external signal, something like $v(n) \propto \alpha(t)$. If $\alpha(t) = 0$ the edge/bump would stop moving, regardless of its location. If $\alpha(t) > 0$, then when it is bigger the edge/bump moves faster. If $\alpha(t)$ is controlled by the time rate of change of some external variable such that $\alpha(t) = \frac{dz}{dt}$, then the edge/bump moves like $\log_a z$.

This property is extremely useful in computational neuroscience. For instance, the place code in the hippocampus and related regions is believed to result from integration of velocity signals. In the context of spatial navigation, neurons participating in an edge attractor driven by spatial velocity would behave like "border cells" (Campbell et al., 2018; Solstad, Boccara, Kropff, Moser, & Moser, 2008). Critically, the edge attractor model proposed here predicts that border cells should have a continuous distribution of space constants controlling the width of their spatial receptive fields. Cells participating in the bump attractor would behave in one dimension as boundary vector cells (Barry et al., 2006; Lever, Burton, Jeewajee, O'Keefe, & Burgess, 2009; Sheehan, Charczynski, Fordyce, Hasselmo, & Howard, 2021); in two dimensions conjunctions of boundary vector cells would have properties like canonical hippocampal place cells (Burgess & O'Keefe, 1996).

Similarly, cognitive and neural models of decision-making accumulate evidence over macroscopic periods of time (Gold & Shadlen, 2007; Ratcliff, 1978). At each moment, evidence accumulator models keep track of the distance in an abstract space to decision bounds. When the distance to one of the bounds reaches zero, a behavioral decision is made. In this case, the velocity signal $\alpha(t)$ is the evidence available at each moment t (Howard et al., 2018). Neurons in $x(n)$ participating in evidence accumulation should behave more or less like classical evidence accumulation cells ramping until a threshold when a decision is made (Hanes & Schall, 1996; Shadlen & Newsome, 2001). Different units in $x(n)$ state ramp at different rates depending on their value of n, exhibiting different "evidence constants." Neurons participating in the bump during evidence accumulation should have receptive fields at different locations along the evidence axis,

not unlike observations in rodent cortex (Koay, Charles, Thiberge, Brody, & Tank, 2022; Morcos & Harvey, 2016).

3.3 Cognitive models using edge/bump attractors

In many physical systems a coarse-grained description of the problem gives rise to different physics than a finer-grained description. In this sense the levels of description each require their own models (Anderson, 1972). Ideally, each of the levels should seamlessly connect to one another. The major challenge of computational cognitive neuroscience is mapping neural circuits onto cognitive models. Cognitive models provide a low-dimensional description of behavior; if neural circuits can be mapped directly onto cognitive models, then this provides a link from that circuit model to the behavior of the organism. For instance, the diffusion model (Ratcliff, 1978) describes the degrees of freedom necessary to provide a more or less complete description of behavior in many evidence accumulation tasks. Insofar as that is true, to the extent an RNN or an attractor model can be mapped onto the diffusion model (Daniels, Flack, & Krakauer, 2017) it can describe all the relevant behavior, but may or may not accurately describe neural-level mechanisms. The flexibility of Laplace/inverse representations for cognitive models of a wide range of behavioral tasks (Howard et al., 2018, 2015; Tiganj et al., 2021) suggests that the circuit developed here could be adapted for many cognitive operations. The Laplace Neural Manifold also provides an alternative hypothesis for constraining neural mechanisms, which makes distinct predictions about the mapping onto neural data.

Most of contemporary computational neuroscience starts from neural circuits and attempts to map onto behavior as an emergent property. The present approach—mapping cognitive models specified as Laplace transform and inverse onto neural circuits—is conceptually distinct. Rather than starting from neural circuits that seem biologically reasonable, this paper starts with a hypothesis for the collective behavior of neurons and then constructs a neural circuit to satisfy those requirements. Because the form of representation is motivated by constraints from cognition, the mapping onto cognitive models is assured.

3.3.1 Neural models for evidence accumulation

The traditional way of interpreting the diffusion model is that each neuron in a population provides a noisy estimate of the decision variable (Zandbelt, Purcell, Palmeri, Logan, & Schall, 2014). This then suggests a dynamical system view where the two options in the decision are associated with mutually exclusive attractors (Bogacz, Brown, Moehlis, Holmes, & Cohen, 2006; Wang, 2008). Distance to bound can then be derived by a linear projection of the network state along the vector connecting the two attractors. This approach makes several empirical predictions that are distinct from the present edge/bump system. Most notably, there is no natural way to map this idea onto the observation of decision-related sequences. Conversely, circuit models that create decision-related sequences do not provide an account for why in other brain regions and/or behavioral tasks, monotonic decision-related ramps are observed (Brown et al., 2023). Starting from the assumption that the decision variable is mapped

onto a Laplace Neural Manifold naturally generates both of these two functional cell types. Neurons in $x(n)$ should show classical ramping behavior as a function of the decision variable; neurons in $y(n)$ should show sequential activation as a function of the decision variable.

Rather than voting for the time since an item was experienced—or the time until a planned action is executed or a decision variable etc— with their firing rates, in this view, neurons collectively represent information. This collective encoding differs from homogeneous voting when decomposing the unique and redundant contributions of individual neurons (Daniels, Ellison, Krakauer, & Flack, 2016). Specifically, each neuron in $x(n)$ spends most of its time in a high state of activation or a low state of activation, providing information about whether τ (or $T - \tau$) is greater or lesser than $1/s_n$. Although the total activation across $x(n)$ is *correlated* with $\log \tau$, more precise information about time can be read off by observing the pattern across the entire population. Similarly, individual neurons in $y(n)$ provide information about the number τ by means of the location of the bump; the average firing rate over the network is not correlated with τ.

The Laplace manifold mechanism also differs from fixed-point attractor models of decision making in the origin of slow variables (Khona & Fiete, 2022; Langdon, Genkin, & Engel, 2023). Models that represent a binary decision using two attractors must tune one collective variable to a critical point in order to produce critical slowing down that allows for evidence accumulation at a slower timescale than intrinsic neural dynamics (Daniels et al., 2017; Wang, 2002). The Laplace manifold mechanism instead has built-in slow dynamics from the continuous symmetry of the attractor, which requires tuning of all N neurons.

3.3.2 Data-independent operators for cognitive computation

Finally, we note that properties of the Laplace transform make it well-suited to describe data-independent operators for *manipulating* information (Howard & Hasselmo, 2020; Howard et al., 2015). The network $x(n)$ represents the Laplace transform of a delta function—essentially a single continuous number—on a logarithmically-compressed number line. There is a great deal of evidence that the nonverbal number system is also logarithmically-compressed (Dehaene & Brannon, 2011; Gallistel & Gelman, 2000; Nieder & Dehaene, 2009). If we could construct neural operators to, say, add and subtract any arbitrary pair of numbers, this would be extremely powerful for cognitive computation (Fodor & Pylyshyn, 1988; Gallistel & King, 2011). If it were possible to combine information from two networks representing numbers with the representation used by $x(n)$, it would be straightforward to write out networks for addition and subtraction. For instance, note that if we have two delta functions f and g centered at locations a and b, then the convolution $f * g$ is centered at $a + b$. Thus, adding two numbers amounts to convolving the two delta functions. The Laplace transform is extremely efficient for computing convolutions due to the fact that the transform of the convolution of two functions is simply the pointwise product of the two transforms. In this way neural circuits for numerical cognition can be constructed by elaborating the edge/bump attractor network developed here.

4 Acknowledgments

We gratefully acknowledge Sarah Marzen and Jim Crutchfield for organizing the workshop "Sensory Prediction, Engineered And Evolved" at the Santa Fe Institute where this work germinated, and all the participants in that workshop for stimulating presentations and discussion.

5 Code availability

All code for simulating the continuous attractor network and creating the simulation figures can be found at https://github.com/Collective-Logic-Lab/laplace-decisions.

References

Affan, R.O., Bright, I.M., Pemberton, L., Cruzado, N.A., Scott, B.B., Howard, M. (2024). Ramping dynamics in the frontal cortex unfold over multiple timescales during motor planning. *bioRxiv*, 2024–02,

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics*, *27*(2), 77–87,

Anderson, P.W. (1972). More is different. *Science*, *177*(4047), 393–396,

Atanas, A.A., Kim, J., Wang, Z., Bueno, E., Becker, M., Kang, D., ... others (2023). Brain-wide representations of behavior spanning multiple timescales and states in c. elegans. *Cell*, *186*(19), 4134–4151,

Barri, A., Wiechert, M., Jazayeri, M., DiGregorio, D. (2022). Synaptic basis of a sub-second representation of time in a neural circuit model. *Nature Communications*, *13*(1), 7902,

Barry, C., Lever, C., Hayman, R., Hartley, T., Burton, S., O'Keefe, J., ... Burgess, N. (2006). The boundary vector cell model of place cell firing and spatial memory. *Reviews in Neuroscience*, *17*(1-2), 71-97,

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., Cohen, J.D. (2006). The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, *113*(4), 700-765,

Bright, I.M., Meister, M.L.R., Cruzado, N.A., Tiganj, Z., Buffalo, E.A., Howard, M.W. (2020). A temporal record of the past with a spectrum of time constants in the monkey entorhinal cortex. *Proceedings of the National Academy of Sciences*, *117*, 20274-20283,

Brown, L.S., Cho, J.R., Bolkan, S.S., Nieh, E.H., Schottdorf, M., Tank, D.W., ... Goldman, M.S. (2023). Neural circuit models for evidence accumulation through choice-selective sequences. *bioRxiv*, 2023–09,

Burgess, N., & O'Keefe, J. (1996). Neuronal computations underlying the firing of place cells and their role in navigation. *Hippocampus.*, *6*(6), 749-62,

Campbell, M.G., Ocko, S.A., Mallory, C.S., Low, I.I.C., Ganguli, S., Giocomo, L.M. (2018). Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. *Nature Neuroscience*, *21*(8), 1096-1106, https://doi.org/10.1038/s41593-018-0189-y

Cao, R., Bladon, J.H., Charczynski, S.J., Hasselmo, M., Howard, M. (2022). Internally generated time in the rodent hippocampus is logarithmically compressed. *eLife*, *https://doi.org/10.7554/eLife.75353*, ,

Cao, R., Bright, I.M., Howard, M.W. (2023). The time of future actions is represented with continuous time constants in mpfc. *Society for neuroscience annual meeting planner* (p. 4432). Washington, DC.

Cao, R., Bright, I.M., Howard, M.W. (2024). Ramping cells in rodent mpfc encode time to past and future events via real laplace transform. *bioRxiv*, ,

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(03), 181–204,

Dahmen, D., Grün, S., Diesmann, M., Helias, M. (2019). Second type of criticality in the brain uncovers rich multiple-neuron dynamics. *Proceedings of the National Academy of Sciences*, *116*(26), 13051–13060,

Daniels, B.C., Ellison, C.J., Krakauer, D.C., Flack, J.C. (2016). Quantifying collectivity. *Current Opinion in Neurobiology*, *37*, 106–113, https://doi.org/http://dx.doi.org/10.1016/j.conb.2016.01.012

Daniels, B.C., Flack, J.C., Krakauer, D.C. (2017). Dual coding theory explains biphasic collective computation in neural decision-making. *Frontiers in Neuroscience*, *11*, 313,

Dehaene, S., & Brannon, E. (2011). *Space, time and number in the brain: Searching for the foundations of mathematical thought.* Academic Press.

Fechner, G. (1860/1912). *Elements of psychophysics. Vol. I.* Houghton Mifflin.

Feigenson, L., Dehaene, S., Spelke, E. (2004). Core systems of number. *Trends in Cognitive Sciences*, *8*(7), 307–314,

Ferbinteanu, J., & Shapiro, M.L. (2003). Prospective and retrospective memory coding in the hippocampus. *Neuron*, *40*(6), 1227-39,

Fodor, J.A., & Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1), 3–71,

Fransén, E., Tahvildari, B., Egorov, A.V., Hasselmo, M.E., Alonso, A.A. (2006). Mechanism of graded persistent cellular activity of entorhinal cortex layer V neurons. *Neuron*, *49*(5), 735-46,

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, *11*, 127-138,

Friston, K., & Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1521), 1211-21, https://doi.org/10.1098/rstb.2008.0300

Gallistel, C.R., & Gelman, R. (2000). Non-verbal numerical cognition: From reals to integers. *Trends in Cognitive Sciences*, *4*(2), 59–65,

Gallistel, C.R., & King, A.P. (2011). *Memory and the computational brain: Why cognitive science will transform neuroscience* (Vol. 6). John Wiley & Sons.

Gauthier, J.L., & Tank, D.W. (2018). A dedicated population for reward coding in the hippocampus. *Neuron*, *99*, 179-193.e7,

Gold, J.I., & Shadlen, M.N. (2007). The neural basis of decision making. *Annual Review Neuroscience*, *30*, 535–574,

Guo, C., Huson, V., Macosko, E.Z., Regehr, W.G. (2021). Graded heterogeneity of metabotropic signaling underlies a continuum of cell-intrinsic temporal responses in unipolar brush cells. *Nature Communications*, *12*(1), 1–12,

Hanes, D.P., & Schall, J.D. (1996). Neural control of voluntary movement initiation. *Science*, *274*(5286), 427–430,

Howard, M.W., Esfahani, Z.G., Le, B., Sederberg, P.B. (2023). Foundations of a temporal RL. *arXiv preprint arXiv:2302.10163*, ,

Howard, M.W., & Hasselmo, M.E. (2020). Cognitive computation using neural representations of time and space in the Laplace domain. *arXiv preprint arXiv:2003.11668*, ,

Howard, M.W., Luzardo, A., Tiganj, Z. (2018). Evidence accumulation in a Laplace decision space. *Computational Brain and Behavior*, *1*, 237-251,

Howard, M.W., Shankar, K.H., Aue, W., Criss, A.H. (2015). A distributed representation of internal time. *Psychological Review*, *122*(1), 24-53,

Jin, D.Z., Fujii, N., Graybiel, A.M. (2009). Neural representation of time in cortico-basal ganglia circuits. *Proceedings of the National Academy of Sciences*, *106*(45), 19156–19161,

Khona, M., & Fiete, I.R. (2022). Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 1–23,

Kim, S.S., Rouault, H., Druckmann, S., Jayaraman, V. (2017). Ring attractor dynamics in the drosophila central brain. *Science*, *356*(6340), 849–853,

Koay, S.A., Charles, A.S., Thiberge, S.Y., Brody, C.D., Tank, D.W. (2022). Sequential and efficient neural-population coding of complex task information. *Neuron*, *110*(2), 328–349,

Langdon, C., Genkin, M., Engel, T.A. (2023). A unifying perspective on neural manifolds and circuits for cognition. *Nature Reviews Neuroscience*, 1–15,

Lever, C., Burton, S., Jeewajee, A., O'Keefe, J., Burgess, N. (2009). Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, *29*(31), 9771-7,

MacDonald, C.J., Lepage, K.Q., Eden, U.T., Eichenbaum, H. (2011). Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, *71*(4), 737-749,

Mello, G.B., Soares, S., Paton, J.J. (2015). A scalable population code for time in the striatum. *Current Biology*, *25*(9), 1113–1122,

Morcos, A.S., & Harvey, C.D. (2016). History-dependent variability in population dynamics during evidence accumulation in cortex. *Nature Neuroscience*, *19*(12), 1672–1681,

Nieder, A., & Dehaene, S. (2009). Representation of number in the brain. *Annual Review of Neuroscience*, *32*, 185-208, https://doi.org/10.1146/annurev.neuro.051508.135550

Palmer, S.E., Marre, O., Berry, M.J., 2nd, Bialek, W. (2015). Predictive information in a sensory population. *Proceedings of the National Academy of Sciences USA*, *112*(22), 6908-13, https://doi.org/10.1073/pnas.1506855112

Pastalkova, E., Itskov, V., Amarasingham, A., Buzsaki, G. (2008). Internally generated cell assembly sequences in the rat hippocampus. *Science*, *321*(5894), 1322-7,

Rao, R.P., & Ballard, D.H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79-87, https://doi.org/10.1038/4580

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*, 59-108,

Redish, A.D., & Touretzky, D.S. (1997). Cognitive maps beyond the hippocampus. *Hippocampus*, *7*, 15-35,

Sarel, A., Finkelstein, A., Las, L., Ulanovsky, N. (2017). Vectorial representation of spatial goals in the hippocampus of bats. *Science*, *355*(6321), 176–180,

Schöner, G., & Spencer, J.P. (2016). *Dynamic thinking: A primer on dynamic field theory.* Oxford University Press.

Shadlen, M.N., & Newsome, W.T. (2001). Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey. *Journal of Neurophysiology*, *86*(4), 1916–1936,

Shankar, K.H., & Howard, M.W. (2010). Timing using temporal context. *Brain Research*, *1365*, 3-17,

Sheehan, D.J., Charczynski, S., Fordyce, B.A., Hasselmo, M.E., Howard, M.W. (2021). A compressed representation of spatial distance in the rodent hippocampus. *bioRxiv*, ,

Solstad, T., Boccara, C.N., Kropff, E., Moser, M.B., Moser, E.I. (2008). Representation of geometric borders in the entorhinal cortex. *Science*, *322*(5909), 1865-8,

Stern, M., Istrate, N., Mazzucato, L. (2023). A reservoir of timescales emerges in recurrent circuits with heterogeneous neural assemblies. *Elife*, *12*, e86552,

Taube, J.S. (1998). Head direction cells and the neurophysiological basis for a sense of direction. *Progress in Neurobiology*, *55*(3), 225-256,

Tiganj, Z., Cromer, J.A., Roy, J.E., Miller, E.K., Howard, M.W. (2018). Compressed timeline of recent experience in monkey lPFC. *Journal of Cognitive Neuroscience*, *30*, 935-950,

Tiganj, Z., Hasselmo, M.E., Howard, M.W. (2015). A simple biophysically plausible model for long time constants in single neurons. *Hippocampus*, *25*(1), 27-37,

Tiganj, Z., Tang, W., Howard, M. (2021). A computational model for simulating the future using a memory timeline. *Proceedings of the annual meeting of the cognitive science society* (Vol. 43).

Tsao, A., Sugar, J., Lu, L., Wang, C., Knierim, J.J., Moser, M.-B., Moser, E.I. (2018). Integrating time from experience in the lateral entorhinal cortex. *Nature*, *561*, 57-62,

Van Essen, D.C., Newsome, W.T., Maunsell, J.H. (1984). The visual field representation in striate cortex of the macaque monkey: asymmetries, anisotropies, and individual variability. *Vision Research*, *24*(5), 429-48,

Wagner, M.J., Kim, T.H., Savall, J., Schnitzer, M.J., Luo, L. (2017). Cerebellar granule cells encode the expectation of reward. *Nature*, *544*(7648), 96–100,

Wang, X.-J. (2002). Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, *36*(5), 955–968,

Wang, X.-J. (2008). Decision making in recurrent neuronal circuits. *Neuron*, *60*(2), 215–234,

Yoshida, M., Fransén, E., Hasselmo, M.E. (2008). mGluR-dependent persistent firing in entorhinal cortex layer III neurons. *European Journal of Neuroscience*, *28*(6), 1116-26,

Zandbelt, B., Purcell, B.A., Palmeri, T.J., Logan, G.D., Schall, J.D. (2014). Response times from ensembles of accumulators. *Proceedings of the National Academy of Sciences*, *111*(7), 2848–2853,

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, *16*(6), 2112-26,

Zuo, S., Wang, C., Wang, L., Jin, Z., Kusunoki, M., Kwok, S.C. (2023). Neural signatures for temporal-order memory in the medial posterior parietal cortex. *bioRxiv*, 2023–08,

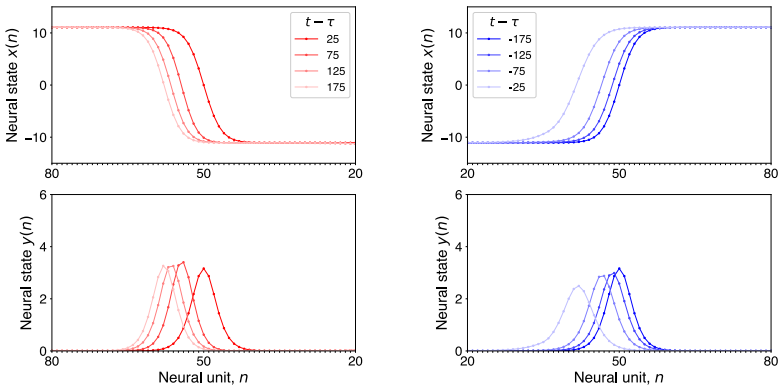

Fig. A1 The motion of symmetric edges. Layout analogous to Figure 4. However, this uses the symmetric recurrent weight kernel in Eq. A1 instead of Eq. 23.

A1 Appendix

A1.1 Using a symmetric edge

Using a recurrent weight kernel $w_r(n - n')$ that is symmetric (even in $\Delta n = n - n'$) produces symmetric edge shapes. As discussed in the Methods, such a setup will still produce an edge that moves with the desired speed, with dynamics of individual neurons that are still simply rescaled in time. Yet the symmetric case does not produce a representation of the Laplace transform, as the state of individual neurons does not depend exponentially on time.

To demonstrate this case explicitly, we simulate the dynamics using a Gaussian kernel:

$$w_r(\Delta n) = J \exp(-\Delta n/\sigma_k^2). \tag{A1}$$

The resulting dynamics are shown in Figs. A1 and A2, using $J = 22$, $\sigma_k = 1$ and other parameters as in Table 1.

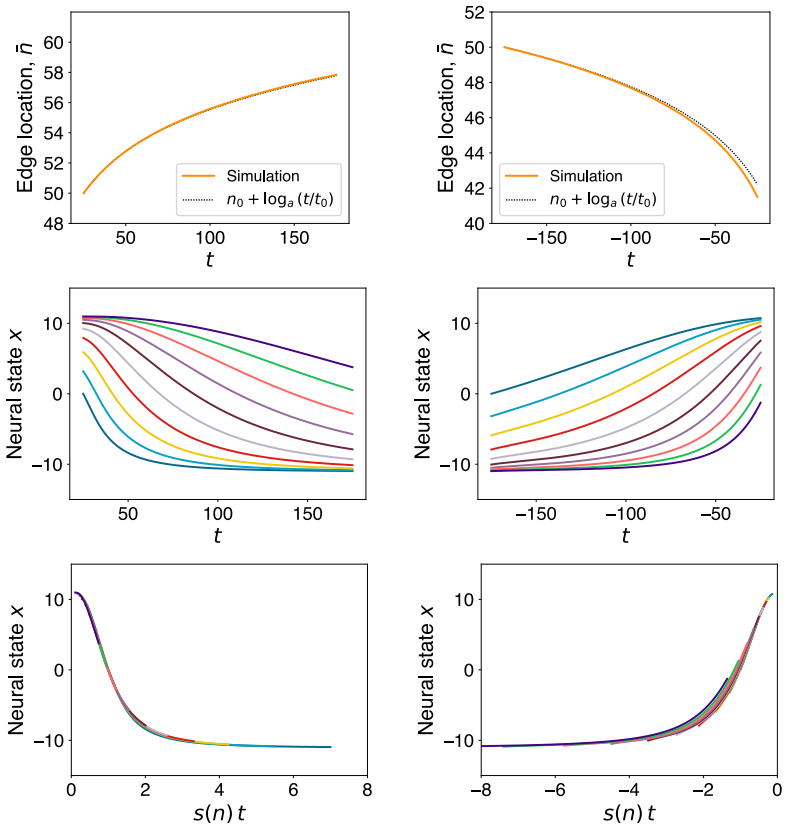

Fig. A2 Symmetric case reproduces time scaling properties except for exponential decay. Layout analogous to Figure 5. However, this uses the symmetric recurrent weight kernel in Eq. A1 instead of Eq. 23.