

1 A theory and recipe to construct general and biologically
2 plausible integrating continuous attractor neural networks

3 Federico Claudi^{1*}, Sarthak Chandra^{1*}, and Ila R. Fiete^{1,2}

4 ¹Department of Brain and Cognitive Sciences & McGovern Institute, MIT

5 ²Integrative Computational Neuroscience Center and Yang-Tan Collective, MIT

6 *Both authors contributed equally to this work.

7 **Abstract**

8 Across the brain, circuits with continuous attractor dynamics underpin the representation and storage in memory of continuous variables for motor control, navigation, and mental computations. The represented variables have various dimensions and topologies (lines, rings, euclidean planes), and the circuits exhibit continua of fixed points to store these variables, and the ability to use input velocity signals to update and maintain the representation of unobserved variables, effectively integrating the incoming velocity signal. Integration constitutes a general computational strategy that enables variable state estimation when direct observation of the variable is not possible, suggesting that it may play a critical role in other cognitive processes. While some neural network models for integration exist, a comprehensive theory for constructing neural circuits with a given topology and integration capabilities is lacking. Here, we present a theoretically-driven design framework, Manifold Attractor Direct Engineering (MADE), to automatically, analytically, and explicitly construct biologically plausible continuous attractor neural networks with diverse user-specified topologies. We show how these attractor networks can be endowed with accurate integration functionality through biologically realistic circuit mechanisms. MADE networks closely resemble biological circuits where the attractor mechanisms have been characterized. Additionally, MADE offers innovative and minimal circuit models for uncharacterized topologies, enabling a systematic approach to developing and testing mathematical theories related to cognition and computation in the brain.

9 **Introduction**

10 The brains of species from insects to mammals contain circuits specialized to represent
11 and integrate continuous variables (Figure 1A) [1, 2]: the head direction circuits in mammals
12 [3, 4, 5], fish [6], and flies [7, 8, 9, 10], the oculomotor system of vertebrates [11, 12, 13, 14, 15],
13 and grid cell networks in mammals [16, 17, 18] (see Figure 1B,C,D). These circuits receive
14 velocity inputs, representing the rate of change of the represented variable, and update their
15 internal state in proportion to the instantaneous velocity [1]. The oculomotor circuit integrates
16 head velocity signals to counter-rotate the eyes and hold the gaze fixed during head movements
17 [15, 11]; it also integrates saccadic velocity commands to generate stable fixations at different
18 gaze angles between saccades [13]. In the head direction and grid cell circuits for spatial
19 navigation, self-movement cues from turning and walking update the internal pose estimates
20 [5, 19, 20, 21, 22]. This so-called path integration (PI) computation underpins behaviors that
21 are core for survival [23, 24].

22 Integration may also underlie the representation and mapping of other continuous domains
23 including auditory sound spaces, parametric image variations, and emotional/aggression states

24 [25, 26, 27, 28, 29, 30, 31, 32], and thus support inference, reasoning, planning, and imagination
25 in all these domains.

26 Neural network models of these integrator circuit fall under the category of continuous
27 attractor networks (CANs) [33, 1, 34, 35, 36, 15, 18]. All continuous attractor models posit
28 recurrent circuitry to generate a continuous set of states that persist in the absence of external
29 inputs (continuous attractors). However, not all CAN models are integrators: integrators must
30 additionally contain a mechanism for updating the internal state based on velocity inputs.
31 CAN models generate extensive predictions about circuit connectivity, activity, population
32 dynamics, and lesions, and have stimulated extensive experimental work across species and
33 circuits to test their predictions. Core novel predictions of these models have subsequently
34 been validated via physiology, imaging, and connectomics: the dynamics and connectivity of
35 the oculomotor integrator [37, 14, 12, 38] have been shown to match the hypothesized circuit
36 model in considerable detail. The one-dimensional ring attractor dynamics, including fixed
37 point dynamics, isometric representation in the head direction circuit in mammals matches
38 [3] the predicted population dynamics of the ring integrator models in detail [39, 35, 34]. In
39 insects, the connectivity and physical layout of the head direction circuit form an actual ring
40 [7, 10] and exhibit some of the shift-like asymmetries hypothesized by a subset of the ring
41 attractor models [39, 35]. In the grid cell system, the invariant two-dimensional population
42 dynamics [40, 41, 42, 43] and its localization [44] to the predicted torus of fixed point states
43 [18, 45, 46] has been directly observed in experiments. Thus, when available, circuit models
44 have propelled a conceptual understanding of the structure and function of the mechanisms
45 involved in integration, memory, and control of continuous variables, and driven experiments
46 that have confirmed their mechanistic hypotheses

47 These models have been hand-crafted through intuition and insight, individually for each
48 circuit or system in the brain. It is remarkable that the corresponding biological circuits
49 have been found possess a structure, in the population dynamics and when direct physical
50 comparisons have been possible in the circuit architecture, that closely matches these models
51 [37, 14, 12, 40, 41, 42, 43, 7, 10, 38, 1]. This suggests that mathematically guided and
52 conceptually minimal models are well-matched to the biology of the brain. Yet we lack a
53 general mathematical theory to allow researchers to automatically construct such models
54 for other continuous variables of a given dimension and topology, to generate predictions for
55 future experiments and for potential use in machine learning applications involving such input
56 variables.

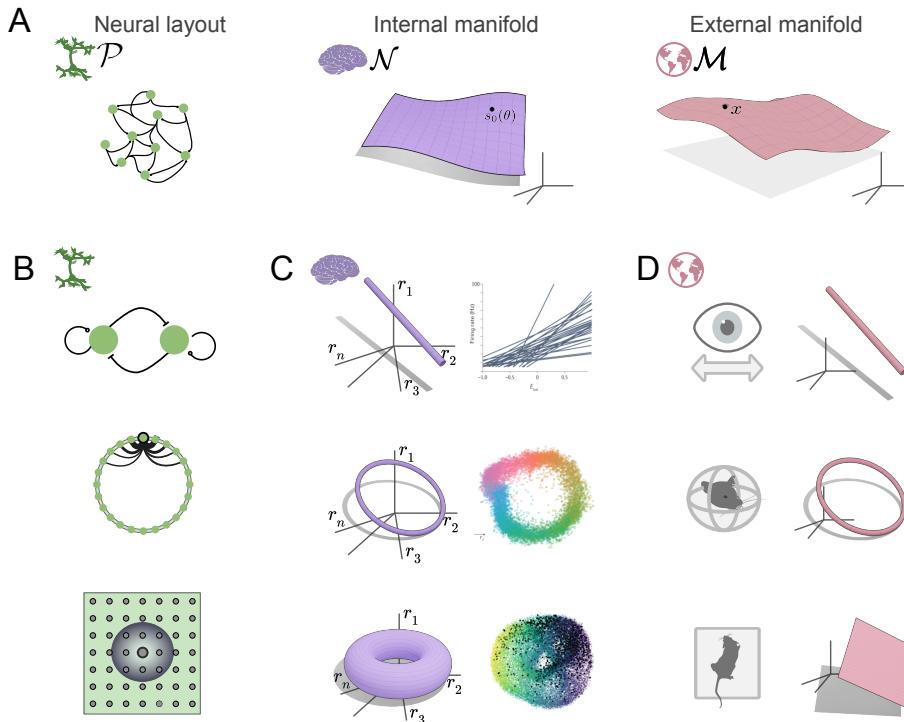


Figure 1: Existing continuous attractor network models and the biological systems where they are found.

A. Left, schematic representation of a spatially embedded set of neurons and their connections. The neural connectivity constrains the patterns of neural co-activation, thus determining the dimensionality and topology of neural activity in the state space. Center, schematic representation of neural activity states, in this case forming a continuous manifold in state space. Right, schematic representation of the states of a (latent) variable in the external world. **B,C,D.** Examples of integrator circuits. Top row, integration in the oculomotor system. Center row, head direction system. Bottom row shows the grid cell system. **B.** Schematic representation of CAN models architecture for line, ring and torus attractors. **C.** Schematic illustration of the continuous manifolds of fixed points predicted and found to exist in the corresponding circuits, adapted from published work [15, 14, 34, 47, 35, 3, 48, 18, 46, 44]. **D.** Schematic illustration of variable manifolds.

57 Recent efforts to overcome this limitation center on training networks via gradient learning
58 to perform continuous integration tasks on the desired variable [49, 50, 51, 52]. However, the
59 difficulties of this approach for the formation of continuous attractor networks is that it is in-
60 efficient, and the results are not usually interpretable. Specifically, training on M -dimensional
61 manifolds requires of order k^M samples [53, 54], scaling exponentially with manifold dimen-
62 sion. In the few cases where the results become interpretable, it is only through mapping
63 onto the original “hand designed” models. The combination of these factors and the striking
64 match between biology and the minimal hand-crafted models suggests that a set of simple
65 and general mathematical principles are used by biology to build such circuits and if discov-
66 ered, can be used to directly construct circuit models for integration of arbitrary continuous
67 variables.

68 Here, we present such a small set of mathematical principles to directly construct minimal,
69 interpretable continuous attractor networks that integrate variables of rich topologies and ge-
70ometries. The theoretical framework converts directly into a practical recipe for constructing
71 integrating attractor networks of desired dimension and topology. Existing integration net-
72 works known from biology appear as special cases of this framework. We name the method
73 MADE (Manifold Attractor Direct Engineering). Thus, MADE can serve as a generally useful
74 framework for making circuit predictions about connectivity and function in neural integrators
75 not yet discovered, including in high-level areas that perform various cognitive tasks.

76 Results

77 Integration is the task of estimating the value of some (potentially latent) continuous
78 variable $x(t)$, based on an initial condition and inputs conveying information about $\dot{x}(t) \equiv$
79 $dx(t)/dt$, its instantaneous rate of change. For a variable to be integrable, it must be contin-
80 uous and lie on a ‘differentiable manifold’: a smooth, continuous space that at small scales is
81 similar to Euclidean space, though globally it may be non-Euclidean, with complex topology.
82 For a neural circuit to integrate, its representations must form a differential manifold, and if
83 the velocity signal is zero then the read out state should not change over time. In constructing
84 a neural circuit that can integrate a given variable, we therefore need two components: a net-
85 work that possesses a manifold of states that support a stable readout value, whose dimension
86 and topology matches the variable, and a mechanism to allow velocity inputs to move states
87 along the manifold. In what follows, we derive a general theory for achieving both with neural
88 circuits, assuming that the stable readouts are stable population states on the manifold.

89 Theory: Continuous attractor manifolds of desired dimension and topology

90 Here we describe the theoretical elements sufficient to construct a neural network possessing
91 a continuous set of attractor states with desired intrinsic dimensionality d (e.g., $d = 1$ for a
92 ring lattice and $d = 2$ for a plane) and desired topology specified by a manifold \mathcal{P} .

93 Consider a set of N neurons and spatially embed them, equally spaced (in a lattice),
94 according to the desired manifold topology \mathcal{P} . With this embedding, each neuron has a
95 unique d -dimensional coordinate θ_i . This spatial organization is used for the specification of
96 network connectivity, $W_{ij} = W(\theta_i, \theta_j)$; it may but need not mirror the actual locations of
97 neurons in neural tissue [18]. We use rate-based neurons with standard recurrent weighted
98 sums and point-wise neural nonlinearity given by the function f . The activation of the neuron
99 at θ_i is denoted s_{θ_i} . For better analytical transparency — so that weights and activations
100 can be written as functions instead of lists of numbers — we follow others [34, 55] and take
101 the continuum neural field limit. The discrete lattice of positions on the neural manifold \mathcal{P}
102 and neural activations become $\theta_i \rightarrow \theta$, $s_{\theta_i} \rightarrow s(\theta)$, respectively. Additionally, $\sum_i \rightarrow \int d\theta$,

103 $\sum_j W_{ij} s_j \rightarrow \int W(\theta, \theta') s(\theta') d\theta'$, so that the neural network equations are:

$$\tau \frac{ds(\theta)}{dt} + s(\theta) = f \left[\int_{-\infty}^{\infty} W(\theta, \theta') s(\theta') d\theta' + b \right]. \quad (1)$$

104 We will use the rectifying nonlinearity, $f(x) = x$ if $x > 0$ and $f(x) = 0$ if $x \leq 0$. Derivations
105 that follow are conceptually and qualitatively independent of this continuum limit.

106 We seek interaction weights consistent with the formation, through symmetry breaking, of
107 a single activity bump state that can be positioned anywhere on the neural manifold \mathcal{P} . The
108 set of such bump states will form the continuous set of attractor states of desired dimension
109 and topology.

110 Let W be a kernel function, $W(\theta, \theta') = k(d(\theta, \theta'))$, where $d(\theta, \theta')$ is a distance metric
111 defined on \mathcal{P} , and k is a continuous scalar function that is symmetric about the origin (see
112 Figure 2A). Analogous to prior work [34, 48, 56], we set k to be locally excitatory and globally
113 inhibitory. To avoid runaway excitability, we make it strictly inhibition-dominated ($k(d) \leq 0$
114 for all d) as in [57, 18]; network activity can be non-zero because of a compensatory spatially-
115 and temporally-constant excitatory feed-forward drive $b > 0$. Specifically, $k(d) = -k_0 + k_1(d)$,
116 where $k_0 > 0$ is a positive number and $k_1(d) \rightarrow 0$ as $d \rightarrow \infty$ with $k_1(0) = k_0$.

117 Let the kernel's length scale be given by σ , i.e., $k_1(d) \approx 0$ for $d \geq \sigma$, with σ selected to be
118 much smaller than the distances L over which the manifold \mathcal{P} has curvature. Thus, within
119 any ball V_l of radius l such that $\sigma \ll l \ll L$, \mathcal{P} is flat. Since σ is the only spatial scale being
120 introduced in the dynamics, we qualitatively expect that a localized bump state within the
121 ball will have a spatial scale of $\mathcal{O}(\sigma)$. The conditions for the formation of a stable single bump
122 state are thus the same as those for a globally flat manifold.

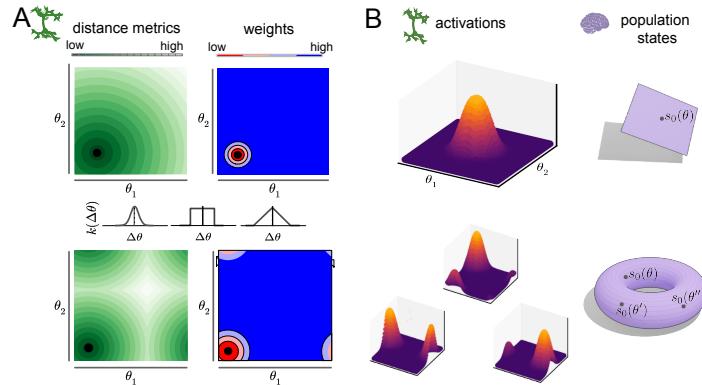


Figure 2: **CAN construction and activity manifolds.** **A.** Left, neural lattice \mathcal{P} for the Plane (top) and Torus (bottom) attractor networks. Black circles indicate the location of an example neuron, shades of green represent distance from other points on the lattice. Bottom right, inhibitory connectivity strength between the example neurons and all other points on the neural lattice. Middle inset, three examples of valid connectivity kernel functions k . **B.** Neural manifold in state space (top,right) and activity patterns on the neural lattice \mathcal{P} (top, left). Bottom row shows three activity patterns with bumps at different locations corresponding to different points on the activity manifold \mathcal{N} .

123 Since W is symmetric, Eq. 1 can be described through an energy function [58], and a
124 stable steady state must exist. If the homogeneous state (all neurons equally active) were
125 unstable, there must exist some other stable state, with broken symmetry. If the symmetry
126 broken state is localized, we would refer to it as a bump state. Thus, we seek conditions under
127 which the homogeneous steady state is unstable. The homogeneous steady state $s(x) = s_0$
128 must satisfy

$$s_0/\tau = s_0 \int W(\theta - \theta') d\theta' + b. \quad (2)$$

129 We derive the existence and stability of the homogeneous state (Appendix 1) following the
130 analysis in Ref. [59], to obtain two requirements for the formation of a stable bump state:

131 first, the Fourier transform of the kernel $k_1(d)$, which we denote as $\tilde{k}_1(\omega)$, must be maximized
 132 at $\omega = 0$; and second, this maximum must be larger than $1/\tau$. If k attains a positive maximum
 133 value at $\omega = 0$, a rescaling can always make this maximum larger than $1/\tau$.

134 A broad sufficiency condition for the first requirement is if $k_1(d) \geq 0$ for all d , then its
 135 Fourier transform is maximized at zero (proof in Appendix 1). This condition does not include
 136 all interaction kernels k_1 whose Fourier transforms are maximized at zero, but is a sufficiently
 137 broad class.

138 Thus, up to a rescaling of the strength of the interaction, an interaction $W(d(\theta, \theta'))$ will
 139 lead to the formation of a bump state if it can be rewritten as $W(d(\theta, \theta')) = k_1(d(\theta, \theta')) - k_0$
 140 for: $k_0 \geq 0$; a kernel k_1 that satisfies $k_1(d) \geq 0$ and $k_1(d) \rightarrow 0$ for $d \geq \sigma$; and sufficiently small
 141 σ over which the manifold \mathcal{P} is approximately flat. As a result, there is a set of stable fixed
 142 points corresponding to activity profiles that are in one-to-one correspondence with points on
 143 \mathcal{P} : every stable single-bump activity pattern is centered at some point in \mathcal{P} , and every point
 144 in \mathcal{P} forms the center of some stable single-bump state (see Figures 2B). Thus, the set of
 145 stable states of the dynamics in Eq. (1) form a continuous attractor manifold \mathcal{N} that has a
 146 bijection with the manifold of the neural layout \mathcal{P} and thus to the target manifold. Moreover,
 147 importantly for representation and integration of continuous variables, we show in Appendix
 148 2 that \mathcal{P} and \mathcal{N} are isometric to each other, with respect to their intrinsic geodesic metrics.

149 Theory: Integration on manifolds

150 The theoretical and practical frameworks outlined above show how to construct neural
 151 networks whose activity states possess a set of attractors forming a manifold \mathcal{N} of desired
 152 dimension and topology. Here, given the desired manifold \mathcal{N} , we describe how the constructed
 153 attractor network with states matching the topology and dimension of \mathcal{N} can be augmented
 154 to endow them with the ability to perform velocity integration.

155 Note that to perform velocity integration of an external observed variable, the desired
 156 manifold \mathcal{N} may, but need not, coincide in dimension and topology with the manifold on
 157 which the observed variable states lie. This possibility is exemplified by grid cells, where the
 158 manifold \mathcal{N} of a grid module is $\mathcal{N} = \mathbb{T}^2$ and is used to integrate animal velocities as animals
 159 move about in physical 2D space (thus $\mathcal{M} = \mathbb{R}^2$). In a future work, we will consider the
 160 question of which internal manifolds \mathcal{N} , not necessarily of the same topology or dimension
 161 as \mathcal{M} , permit accurate integration of velocities on \mathcal{M} . Here we show how to equip networks
 162 with attractor manifold \mathcal{N} with accurate path integration functionality for velocity inputs of
 163 matching dimensionality.

164 Previous models [34, 35, 18] constructed offset interactions between multiple copies of a
 165 continuous attractor network to permit external inputs to drive the state along the mani-
 166 fold. Here, we analytically derive the conditions required for an external input that has no
 167 knowledge about the structure and state of the continuous attractor network to generate
 168 appropriate movements along the nonlinear attractor manifolds of given topology, and show
 169 that offset interactions are necessary solutions.

170 For simplicity, consider a one-dimensional manifold with linear transfer function f . The
 171 stable bump states are fixed points of Eq. 1:

$$s(\theta) = \int W(\theta - \theta') s(\theta') d\theta' + b, \quad (3)$$

172 where $s(\theta)$ denotes an activity bump centered at any point in \mathcal{P} . Consider two such activity
 173 bump states: $s_0(\theta)$ centered at θ_0 and $s_0(\theta + \epsilon)$ centered at $\theta_0 - \epsilon$. For the neural state to
 174 move from $s_0(\theta)$ to $s_0(\theta + \epsilon)$ in time Δt , the time derivative $\partial s / \partial t$ must equal

$$\frac{\partial s(\theta, t)}{\partial t} = \frac{s_0(\theta + \epsilon) - s_0(\theta)}{\Delta t} \approx \frac{\epsilon}{\Delta t} \frac{\partial s_0(\theta)}{\partial \theta}.$$

175 The movement speed is $v = \epsilon/\Delta t$. Multiplying by τ on both sides, we have

$$\tau \frac{\partial s(\theta, t)}{\partial t} = \frac{\tau\epsilon}{\Delta t} \frac{\partial s_0(\theta)}{\partial \theta}. \quad (4)$$

176 We can add 0 to the equation above, in the form $(-s_0 + \int W(\theta - \theta')s_0(\theta')d\theta' + b)$, which is
177 zero because of the equality of Eq. 3), to obtain:

$$\tau \frac{\partial s(\theta, t)}{\partial t} = -s_0 + \int W(\theta - \theta')s_0(\theta')d\theta' + b + \frac{\tau\epsilon}{\Delta t} \frac{\partial s_0(\theta)}{\partial \theta}. \quad (5)$$

178 Comparing this expression to Eq. 1, we see that moving the bump with velocity v can be
179 achieved by adding a feedforward input drive $\frac{\tau\epsilon}{\Delta t} \frac{\partial s_0(\theta)}{\partial \theta}$ to the continuous attractor network.
180 Though this appears to be a simple way to drive the activity bump on the manifold, it
181 would require the external input to “know” the current value of $\frac{\partial s_0(\theta)}{\partial \theta}$, which varies along the
182 manifold. Thus, the external input would need to know both the shape and current state on
183 the internal neural activity manifold.

184 Observing that $\frac{\partial s_0(\theta)}{\partial \theta} = \int \frac{\partial W(\theta - \theta')}{\partial \theta} s_0(\theta')d\theta'$ (from Eq. 3), and grouping like terms, we
185 obtain

$$\tau \frac{\partial s(\theta, t)}{\partial t} = -s_0 + \int \left(W(\theta - \theta') + \frac{\tau\epsilon}{\Delta t} \frac{\partial W(\theta - \theta')}{\partial \theta} \right) s_0(\theta')d\theta' + b. \quad (6)$$

186 This expression has now “internalized” the desired input to move the bump, converting it
187 into the weight asymmetry term $\frac{\partial W(\theta - \theta')}{\partial \theta}$, similar to [34]. The weight asymmetry is internal
188 to the network, thus the velocity external input would not need to be aware of the internal
189 state or shape on the attractor manifold to drive the bump. However, the external input
190 would be required to dynamically modulate the degree of weight asymmetry, a biologically
191 unrealistic requirement. As a final step, observe that for small $\tau\epsilon/\Delta t \equiv \delta$, by Taylor expansion,
192 $W(\theta - \theta') + \delta \frac{\partial W(\theta - \theta')}{\partial \theta} = W(\theta - \theta' + \delta)$. Thus, we obtain

$$\tau \frac{\partial s(\theta, t)}{\partial t} + s(\theta) = \int W(\theta - \theta' + \delta) s(\theta')d\theta' + b \quad (7)$$

193 Because we have that $\delta = \tau\epsilon/\Delta t = \tau v$, the equation above results in a moving bump along
194 the internal state-space manifold of fixed points \mathcal{N} with speed $v = \delta/\tau$, without any external
195 velocity input or temporally varying modulation of network weights. The network corresponds
196 to the original continuous attractor network constructed in the previous section, with the
197 modification that the weights, instead of being symmetric, have a small offset in a particular
198 direction δ along the neural circuit manifold \mathcal{P} . The speed of bump movement on \mathcal{N} is
199 proportional to the magnitude of the offset, $|\delta|$, and inversely proportional to the neural
200 time-constant.

201 This continuous-speed flow may form a periodic cycle on specific manifolds (e.g. Ref.[18,
202 35]). In these cases, the network is a limit cycle attractor. On generic manifolds, however,
203 this flow need not close periodically on itself. The result will be a quasiperiodic attractor
204 dynamics [60]. We therefore refer to these as Quasiperiodic Attractor Networks (QANs). The
205 flow of activity patterns in a QAN defines a constant vector field Ψ on \mathcal{N} .

206 For several attractor manifolds \mathcal{N} of dimension d (in particular, ‘parallelizable manifolds’
207 such as the Euclidean spaces \mathbb{R}^d and the Torii \mathbb{T}^d) it is possible to construct d QANs with
208 linearly independent flows, and $2d$ QANs with two mutually opposing flows in each of d
209 dimensions (defined by weight matrices $W(\theta - \theta' \pm \delta_m)$, where δ_m is a displacement vector of
210 norm $|\delta|$ along the m^{th} manifold dimension). Each sets up a constant vector field $\Psi_{\pm m}$ on \mathcal{N} .
211 For these manifold topologies [34, 35, 18], opposing-pair QANs numbering $2d$, where d is the

212 manifold dimension, can generate smooth non-vanishing flows of any direction at every point
 213 and are thus sufficient to construct integrators. The combined dynamics is given by:

$$\tau \frac{\partial s_{\sigma m}(\theta, t)}{\partial t} + s_{\sigma m}(\theta) = \int \sum_{\sigma' \in \{1, -1\}} \sum_{m'=1}^d W(\theta - \theta' + \sigma' \delta_{m'}) s_{\sigma' m'}(\theta') d\theta' + b + \dot{x}_{\sigma m} \quad (8)$$

214 where $s_{\sigma m}$ indicates neural activities in the individual QANs and $\dot{x}_{\sigma m}$ is an input carrying
 215 information about the rate of change of the external variable in the m^{th} direction.

216 Coupled in this way, the QANs form a network whose combined activity state moves on
 217 \mathcal{N} in a way controlled by the velocity inputs \dot{x} , which modulate the activity levels of the
 218 individual QANs. When $\dot{x}_{\sigma m} = 0$ for all σ, m , the action of the opposing QANs along each
 219 dimension restores the symmetry of the system and s remains stationary (it does not flow
 220 along \mathcal{N}). Otherwise, the terms $\dot{x}_{\sigma m}$ differentially modulate the activation of the QANs,
 221 causing the activity bump on \mathcal{P} to flow in the direction of the positively modulated QANs.
 222 The result is a time-varying vector field Ψ_t . For accurate path integration, the component
 223 vector fields must be smooth and the set of QANs must generate a complete basis set of
 224 non-vanishing vector fields at every point on \mathcal{N} . This condition is satisfied by using $2d$ QANs
 225 for Euclidean spaces \mathbb{R}^d and Torii \mathbb{T}^d , thus the prescription above is sufficient for integration
 226 on these manifolds.

227 On other manifolds, $2d$ opposing QANs for the d manifold dimensions are not sufficient
 228 for accurate integration. For instance, in the case of even-dimensional spheres, the hairy
 229 ball theorem states that every continuous tangent vector field must vanish at some point(s)
 230 [61, 62, 63]. In other words, a continuous vector field $\Psi_{\pm m}$ generated by the QAN prescription
 231 above will be zero somewhere on the sphere; at that location, the QAN will not be able to
 232 drive bump movement; thus, d QAN pairs will not suffice for good integration everywhere.
 233 Further, on non-orientable manifolds such as the Möbius band, it is not possible to define
 234 continuous vector fields that are globally orthogonal everywhere and smooth. Thus, while the
 235 approach above provides a unified way to construct integrating continuous attractor networks
 236 — including all those with a single bump state currently found in the neuroscience literature
 237 [34, 35, 47, 48, 56] — it needs to be further generalized for manifolds that do not permit
 238 non-vanishing continuous tangent vector fields everywhere.

239 **Generalization: Killing vector fields.** To enable accurate path integration over a sig-
 240 nificantly wider set of manifolds (excluding the Klein bottle), we now broaden and further
 241 generalize the concepts developed above. The approach replaces the constant weight offset
 242 vector fields $\Psi_{\pm m}$ with the more generally applicable Killing vector fields [62]: Killing fields
 243 are vector fields on a manifold whose flows preserve the structure of the manifold, i.e., they are
 244 continuous isometries on the manifold. Conceptually, if each point of an object on the mani-
 245 fold is displaced by the corresponding Killing vector, it will move without distortion. Killing
 246 fields form a ‘vector space’, such that linear combinations of Killing fields are also Killing
 247 fields. The manifold isometric property of Killing fields means that activity patterns are
 248 rigidly translated over \mathcal{P} through the flow Ψ_t without changes in area, a necessary condition
 249 for accurate integration [34].

250 To generate Killing fields in each QAN, the constant weight offsets are replaced by an
 251 appropriate position-dependent offset:

$$k(\theta + \delta_{\pm m}) \rightarrow k(d(\theta + \delta_{\pm m}(\theta))), \quad (9)$$

252 where $\pm \delta_m(\theta)$ is the offset vector of the σ, m^{th} QAN at coordinates θ on \mathcal{P} . This allows for
 253 weight offsets to vary at different locations on the manifold \mathcal{N} consistent with non-constant

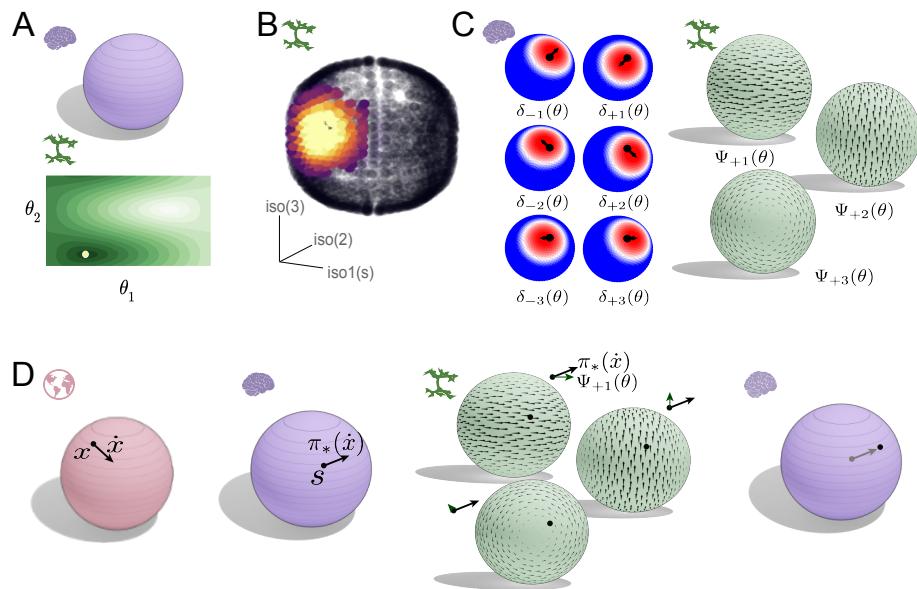


Figure 3: Quasiperiodic Attractor Networks for Path Integration. **A.** Schematic representation of a desired 2D spherical set of fixed points in state space and corresponding connectivity on \mathcal{P} . **B.** Example activity bump plotted on the neural manifold \mathcal{P} . **C.** Schematic illustration of Killing vector fields for the sphere manifold, left, and resulting offset connectivity weights on \mathcal{P} , right. **D.** Schematic illustration of the QAN approach to velocity integration. Left two panels, relationship between changes in the variable on \mathcal{M} and on the neural \mathcal{N} manifold, and associated tangent vectors. Center, each QAN receives a velocity-dependent input based on the tangent vectors at left projected onto its Killing fields, and the activity of all networks is combined. Right: this results in a trajectory in the state-space \mathcal{N} , which corresponds to velocity integration of inputs from \mathcal{M} .

Killing fields required on the sphere (Figure 3C). This simple change, and allowing the number of QANs to be larger than $2d$, endows a much broader class of continuous attractor manifolds including spheres and Möbius band with integration functionality. For a two-dimensional sphere, three basis Killing fields ($d_{kill} = 3$) are required (each corresponding to rotational symmetry along one principal axis; Figure 3C). Although each field vanishes at two points on the sphere, at least two fields are non-vanishing and point in independent directions along the manifold at any point, forming an overcomplete basis such that it is possible for the network to perform accurate path integration.

Finally, we generalize how an external manifold \mathcal{M} may be mapped to the internal integrating manifold \mathcal{N} , by mapping velocity vectors in the external space to the QANs within the network. Throughout, our construction seeks to make \mathcal{P} and \mathcal{N} isometric, and indeed they are, as shown in Appendix 2. However, as noted at the start of this section, \mathcal{N} need not exactly match the topology of the external variable: $\mathcal{N} = \mathbb{T}^2$ of a grid module represents positions on $\mathcal{M} = \mathbb{R}^2$ of the externals partial variable. Similarly, the dimensionality of \mathcal{N} could equal or exceed that of \mathcal{M} : a planar integrator network is capable of integrating an external one-dimensional variable if the velocity inputs are one-dimensional. For instance, grid cell responses on a linear track appear to be generated as a slice through their 2D manifold of states [41, 64].

Define π as the mapping of \mathcal{M} to \mathcal{N} (which can be the identity map or the isomorphism map when \mathcal{M} and \mathcal{N} are isomorphic, such as when head direction is represented in a ring attractor, or a many-to-one map as when spatial position is represented in a single grid module). The Jacobian π_* is a map from the tangent space of \mathcal{M} to the tangent space of \mathcal{N} : it is the operator that maps tangent vectors from \mathcal{M} (i.e. \dot{x}) to tangent vectors of \mathcal{N} [63, 65]. In other words, the velocity vector \dot{x} is ‘pushed forward’ through the map π into

278 $\pi_*(\dot{x})$ (Figure 3D). The coupled system dynamics can be written as

$$\tau \frac{ds_{\sigma m}(\theta, t)}{dt} + s_{\sigma m}(\theta, t) = f \left[\int \sum_{\sigma' \in \{0,1\}} \sum_{m'=1}^{d_{kill}} W_{\sigma' m'} s_{\sigma' m'}(\theta', t) d\theta' + b + \delta_{\sigma m}(\theta) \cdot \pi_*(\dot{x}) \right]. \quad (10)$$

279 where $W_{\sigma m}$ refers to the Killing-field weights from Eq. 9, d_{kill} defines the minimal number
 280 of independent Killing fields. The term $\delta_{\sigma m} \cdot \pi_*(\dot{x})$ refers to the projection of the velocity
 281 pushed through π onto the $(\sigma, m)^{th}$ QAN. Note that in general, the Jacobian π_* maps the
 282 tangent space at a specific point on \mathcal{M} to the tangent space at a specific point on \mathcal{N} , making
 283 it dependent in principle on both $x \in \mathcal{M}$ and $s \in \mathcal{N}$. Thus, neural circuits generating $\pi_*(\dot{x})$
 284 would require access to both the integrator's neural state and the external variable. While
 285 the neural state s is available to the brain, x cannot be directly observed. However, if the
 286 integrator network maintains an accurate estimate of this variable — an expected property
 287 of a reliable integrator — then the brain can instead evaluate π_* at the integrator's state on
 288 \mathcal{M} as a proxy for x .

289 The constant vector fields on the ring and torus manifolds described above (and effectively
 290 discovered in previous work) are Killing fields. Therefore, this approach encompasses previous
 291 work and provides a broader general framework for constructing minimal biologically plau-
 292 sible continuous attractor neural networks capable of path integration on spaces of various
 293 dimension and topology. Next, we demonstrate how to practically construct the networks,
 294 the examine the effectiveness of the approach through extensive numerical simulations of path
 295 integration in MADE integrator networks.

296 Practical construction of CAN integrators with MADE

297 With the complete conceptual and mathematical frameworks in place, we now illustrate
 298 through numerical simulation how to apply the MADE prescription to construct various
 299 CANs and integrators of desired dimension and topology. The simulations also allow us to
 300 validate the functionality of the resulting CANs and integrators. For simplicity, here we focus
 301 our description on one and two-dimensional surfaces, allowing us to construct line, ring, plane,
 302 cylinder, torus, sphere, Möbius band and Klein bottle topologies and geometries (Figure 4A).
 303 The procedures outlined here can be straightforwardly generalized to apply to manifolds of
 304 different dimensionality.

305 We first construct a neural surface \mathcal{P} that is isometric to the target state-space manifold
 306 \mathcal{N} . For the sphere attractor, we construct \mathcal{P} as an embedding of the two-dimensional unit
 307 sphere in \mathbb{R}^3 , and for the Klein bottle attractor \mathcal{P} was an embedding of a finite cylinder
 308 manifold with appropriate identification of the cylinder end-points to each other in \mathbb{R}^4). For
 309 several other manifolds (including all others from Fig. 4), which admit a flat metric, we define
 310 a rectangular two-dimensional space $[0, L_1] \times [0, L_2]$ (Figure 4B) and provide an appropriate
 311 distance function on the rectangular space. For example, for the torus manifold, $L_1 = L_2 =$
 312 2π , and distances are computed respecting the periodic boundary conditions that identify 0
 313 and 2π as the same point.

314 Given \mathcal{P} , we next approximately evenly place neurons on the surface. For manifolds with a
 315 flat metric, this involved placing neurons on an $n \times n$ rectangular lattice on this space, where
 316 n^2 is the total number of neurons. For the sphere, we spaced neurons at regular intervals
 317 along a Fibonacci spiral over the unit sphere (see Methods) to approximate an even placement
 318 on the sphere. Thus, for each neuron we define their \mathcal{P} coordinates θ_i .

319 Next, we computed the connectivity of the network W_{ij} , which depends on the (geodesic)
 320 distances $d(\theta_i, \theta_j)$ between pairs of neurons with coordinates θ_i and θ_j on \mathcal{P} . With appropriate
 321 coordinate parametrization for the neurons, these geodesic distances can be computed via

322 analytical expressions (for instance, as Euclidean distance with periodic boundary conditions
323 on a torus attractor), or via a simple numerical computation (see Methods). Connectivity
324 is then given by the $n^2 \times n^2$ matrix with entries $W_{i,j} = k(d(\theta_i, \theta_j))$, where k is a kernel
325 function (Figures 2A, 3A, 4B) satisfying the requirements described earlier for the formation
326 of activity bump states (see also Appendix 1). We used a scaled Gaussian kernel such that the
327 connectivity between pairs of neurons was strictly negative and $W_{ij} = 0$ if $d(\theta_i, \theta_j) = 0$ (see
328 Methods). Other choices of kernels yield similar results (data not shown). Neural activity is
329 simulated based on these weights according to Eq. 2. We will provide Python and Julia code
330 that implements the MADE prescription for CANs (see Methods).

331 Validation of CAN states and dynamics

332 To validate the MADE CANs, we first characterize where the states of the constructed
333 networks localize. To do so, we sample population activity data from each model by randomly
334 initializing each network and allowing the initial state to settle to a stationary state (see
335 Methods). This state forms one population vector sample; we repeat the process 2500 times
336 for each network. We apply nonlinear dimensionality reduction via ISOMAP, which has
337 proven useful for the visualization of nonlinear low-dimensional manifolds in real data [3], to
338 the resulting point cloud of stationary population activity states. The resulting structures
339 (Figure 4C) visually matched the desired manifolds (Figure 4A): the population responses of
340 the MADE CANs localize to low-dimensional sets of states that appear homeomorphic to \mathcal{N} .

341 To quantify the structure of the resulting population states, we use persistent homology, a
342 Topological Data Analysis [66, 67] technique that has been applied with success in neuroscience
343 [68, 3, 44]. Persistent homology supplies Betti numbers that characterize the topology of the
344 set of stationary states of each network (see Methods). Betti numbers catalog the number of
345 “cavities” of each dimension present on a manifold; the first three Betti numbers correspond
346 to the number of connected components, rings and two dimensional cavities, respectively.
347 Betti numbers don’t provide a complete or unique description of manifold structure (e.g., the
348 ring and the cylinder share the same Betti numbers while having different dimensionality),
349 but they provide a quantitative confirmation that the MADE CANs match their intended
350 targets. The Betti numbers of all MADE CANs population states match those of their target
351 manifolds (Figure 4 D).

352 We next visualize the instantaneous population activity states as functions on the neural
353 lattice. The localized kernel connectivity on the manifold was expected to stabilize single
354 activity bump states on the manifold. A stationary population activity state can be directly
355 visualized on the neural lattice by coloring neurons according to their activity level. Indeed,
356 we see that the stationary population states correspond to localized bumps of activation on
357 the neural lattice \mathcal{P} and activity manifold \mathcal{N} (Figure 4E).

358 Next, we characterize the intrinsic dimensionality [3, 69] of the stationary states of the
359 MADE CANs. Intrinsic dimensionality at a point on a manifold is the numbers of degrees of
360 freedom of movement along the manifold at that point. Intrinsic dimensionality would allow
361 one to distinguish, for example, a ring (one dimensional) from a cylinder (two dimensional).
362 Dimensionality is generally a difficult (and ill-posed) quantity to estimate in noisy data, and
363 existing works use various methods [3, 70, 71, 72, 73]. For MADE CANs, which we can run
364 in a noiseless setting, intrinsic manifold dimension is well-defined.

365 We adopt an approach [73, 72] based on estimating the dimensionality of the tangent space
366 to a manifold (see Methods)(Figure 5 A, left). The tangent space $T_s\mathcal{N}$ at a point $s \in \mathcal{N}$ is the
367 best linear approximation of the manifold at that point and has the same dimensionality as
368 the underlying manifold [74, 63, 65]. We consider the set S of points in a small neighborhood
369 of s (see Methods)and apply PCA to determine the number of large principal components

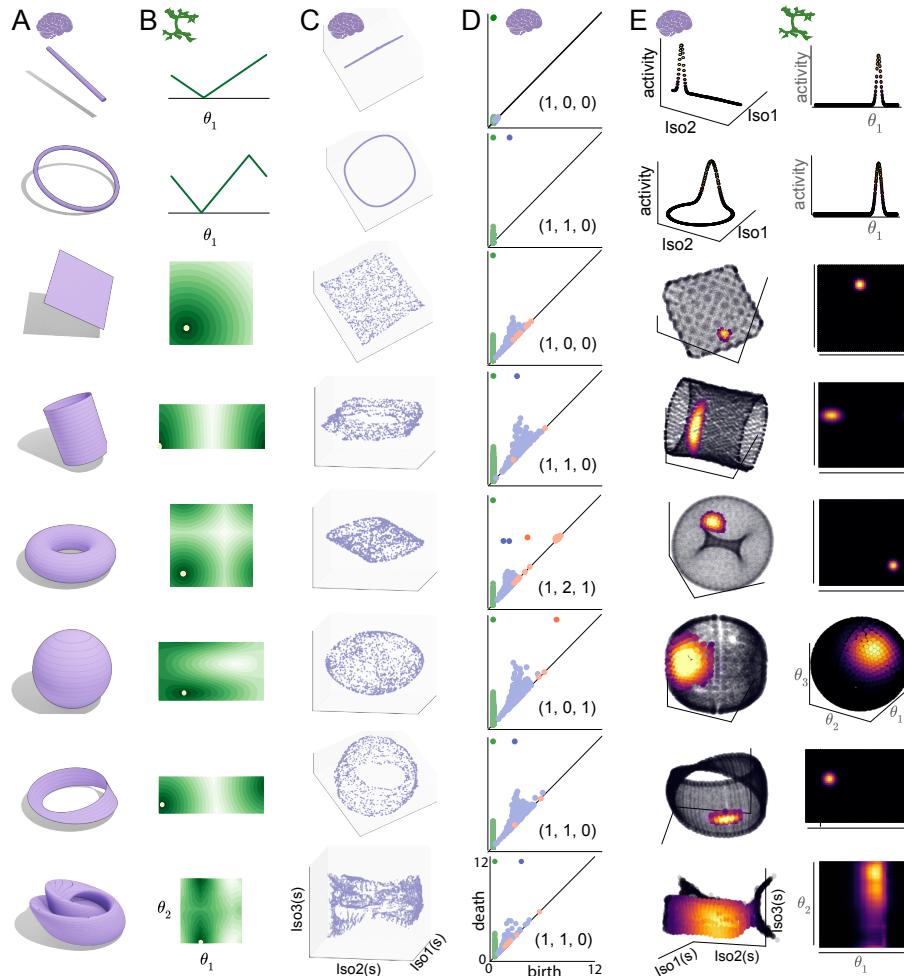


Figure 4: Stationary states and manifold topologies of the MADE CANs **A.** Desired population activity manifold topology for CANs constructed with MADE for several manifolds (from top to bottom): line, ring, plane, cylinder, torus, sphere, Möbius band and Klein bottle. **B.** Distance functions over the neural lattice \mathcal{P} for selected example neurons. **C.** Low dimensional embedding of the neural activity manifold \mathcal{N} . **D.** Betti number and persistent homology bar code for each CAN's neural population states (in \mathcal{N}). **E.** Left: Activity of one example neuron over \mathcal{N} (low dimensional embedding). Right: Stationary population activity states form localized bumps on the neural lattice \mathcal{P} .

needed to describe the data. This gives us the dimensionality of the tangent space at s and thus the local intrinsic dimension of the manifold.

Repeating this analysis across multiple randomly selected sample points s for each MADE CAN, we confirmed that all manifolds had the expected intrinsic dimensionality given their topology: line 1 ± 0.0 (mean \pm standard deviation, across multiple repeats), ring: 1 ± 0.0 , torus: 2 ± 0.0 , sphere: 2 ± 0.0 , Möbius band: 1.96 ± 0.16 , cylinder: 2 ± 0.0 and plane: 2.05 ± 0.23 (Figure 5 A). By contrast to the small intrinsic dimensionality of the constructed CAN manifolds, their extrinsic linear dimensionality, estimated by the minimum number of principal components required to represent the manifold as a whole, is large (Figure 5 B).

Finally, we examined whether the stationary manifolds of the MADE CANs are neutral attractor states, with rapid decay of off-manifold perturbations, together with no state drift along the manifold in the absence of noise and external inputs [34, 1]. First, we consider manifold stability by computing Betti numbers of the population states in networks simulated with varying noise conditions, and find that except in the most severe noise case, we recover the same Betti numbers for the noisy dynamics – indirectly showing that the manifold is attractive and robust to noise (see Methods)(Figure S1). Second, we more directly

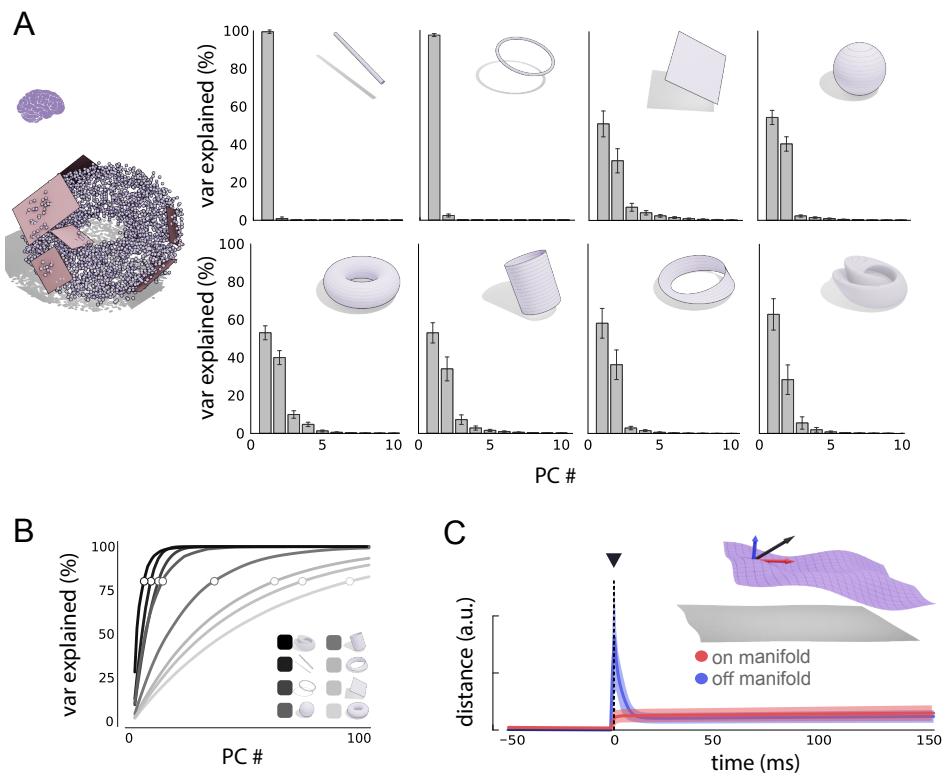


Figure 5: Dimensionality and attractor dynamics of the MADE CANs. **A**, Left, tangent planes approach to computing the intrinsic manifold dimension (schematic) of \mathcal{N} . Right, estimated tangent space dimension for each manifold, which estimates the low intrinsic dimensionality of the CAN networks. **B**, Cumulative manifold variance explained by global PCA analysis: the slow saturation of the curves shows that the linear (embedding) dimension of the manifolds can be large. **C**, Numerical simulations to probe attractor dynamics. Inset: activity manifold, perturbation vector (black) and on-manifold (red) and off-manifold (blue) components of the perturbation. Main plot: Time-varying distance from the starting point in the off-manifold and along-manifold dimensions.

386 perturb the neural population state with a randomly oriented vector of fixed magnitude (see
 387 Methods), repeating this experiment for multiple initial states and random perturbations,
 388 and observe the dynamics by which the perturbed state evolves. To quantify on- and off-
 389 manifold dynamics following perturbation, we again used PCA to estimate the manifold's
 390 tangent space in the neighborhood of the initial state. The distance between the perturbed
 391 and initial (pre-perturbation) states along the tangent space dimension was considered the
 392 on-manifold perturbation component; the rest (along the remaining $N - d$ dimensions) was the
 393 off-manifold perturbation (see Methods). We find very limited on-manifold drift and strong
 394 decay of the off-manifold component of the perturbation, as intended (Figure 5 C).

395 Practical construction of integrators with MADE

396 To generate the QANs that combine to create neural integrator circuits, we slightly modify
 397 the connectivity structure of MADE CANs. We start with the same procedure as before to
 398 construct \mathcal{P} and compute the distance function d . For a QAN indexed by σ, m we simply
 399 apply a shift $\delta_{\sigma m}^\theta$ to the i^{th} neurons coordinates before computing d such that $W_{i,j} = k(d(\theta_i +$
 400 $\delta_{\sigma m}^\theta, \theta_j))$. For some manifolds with a flat metric (e.g. plane, torus) $\delta_{\sigma m}^\theta$ was identical for all
 401 points $\theta \in \mathcal{P}$ and was taken to be a vector of magnitude $|\delta|$ oriented along the m^{th} direction
 402 on \mathcal{P} . In others (e.g. the sphere), the offset vector varied as a function of position along the
 403 manifold. For each dimension m we defined a Killing Vector field $\Psi_{\pm m}$ and evaluated it at θ_i
 404 to obtain the offset vector (see Methods). Given an external velocity signal for a trajectory
 405 on \mathcal{M} , we use the map π from \mathcal{M} to \mathcal{N} to obtain the inputs to each QAN. Network activity

406 is simulated based on weights and these inputs according to Eq. 10. We will provide Python
407 and Julia code to implement the MADE prescription for neural integrators (see Methods).

408 Validation of MADE integrators

409 To examine the performance of each MADE integrator in representing and tracking time-
410 varying external variables, we provide the circuit with the velocity of a simulated random
411 trajectories of the variable $x(t) \in \mathcal{M}$ and track how the network's internal state changes.
412 We first consider how the firing of a single cell varies with the external variable, by plotting
413 its tuning curve or firing response as a function of the external variable, estimated over a
414 long velocity trajectory (Figure 6A). The existence of a localized activity bump (Figure 6A,
415 top three panels) means that the circuit has correctly inferred external position: the cell
416 fires at a specific position and not other random positions, and the network has transferred
417 the internal bump activity pattern into a corresponding pattern as a function of location on
418 the external manifold \mathcal{M} . In cases where the external manifold \mathcal{M} is not isomorphic to the
419 internal manifold \mathcal{N} , such as when a plane in \mathcal{M} is represented by a cylinder or a torus in \mathcal{N} , a
420 continued linear trajectory along one direction in \mathcal{M} corresponds to a periodic traversal on \mathcal{N} ,
421 and thus one would expect repeating bumps in the tuning curve along that dimension, as we
422 find (Figure 6A, panels 4-5). Note that based on the details of how we periodically connected
423 the boundaries of our rectangular neural lattice to obtain a torus, we would obtain a square
424 grid tuning curve (as shown) or a triangular grid tuning curve (as previously described for
425 grid cells in [48, 56]). Finally, the tuning curves for the sphere and Möbius strip are single
426 bumps, as expected (Figure 6A, last two panels).

427 We can more directly quantify how closely the network tracks the external variable $x(t)$ by
428 decoding it from the network's internal state $s(t)$, as $\hat{x}_t = \theta_{\text{argmax}(s(t))} + \omega$ where ω is an offset
429 used to account for the fact that, in some cases, \mathcal{N} was periodic while \mathcal{M} was not (e.g. torus
430 and plane, respectively) (see Methods). When \mathcal{M} and \mathcal{N} are chosen such that π is either an
431 identity map or a periodic mapping, the networks show very accurate integration over periods
432 of several seconds of simulated activity (Figure 6B,C). Decoding error remains low even in
433 the presence of moderate noise (Figure 6D) (see Methods). Thus, MADE networks support
434 accurate integration, even in non-trivial scenarios such as the cylinder-torus manifold pairing
435 and even on the Möbius band manifold, which have not been described previously.

436 We performed additional experiments on circuits requiring Killing vector fields to integrate.
437 To show the necessity of Killing fields, we built torus (\mathcal{N} = torus, \mathcal{M} = plane) and sphere
438 (\mathcal{M} = \mathcal{N} = sphere) integrator networks, but varied the QAN weight offsets relative to the
439 Killing field prescription. For the torus, we varied the orientation of the offset vectors, while
440 for the sphere we changed their lengths to be of constant magnitude everywhere (except at
441 two poles, where the magnitudes were left at 0), (see Figure 6E, left), (see Methods). The
442 constant-magnitude non-Killing field on the sphere may be considered a direct extension of
443 the constant offset vector fields used for flat manifolds and used in all prior work in the
444 construction of neural integrators. In both cases, we observed a dramatic deterioration in
445 integration accuracy, Figure 6E (right). The result underscores the importance of Killing
446 fields for integration on manifolds with a non-flat metric.

447 Finally, we considered integrating velocities from a cylindrical external variable on a net-
448 work with Möbius band topology. Both manifolds are two-dimensional with one periodic and
449 one non-periodic dimension. However, while a rectangle is glued without a twist to make a
450 cylinder (which has two surfaces, inner and outer), it is glued with a twist to make a Möbius
451 band (which has a single surface) with the consequence that there is no simple continuous
452 mapping between the two. Proceeding naively by simply mapping the two manifolds onto
453 each other by ignoring the flipped boundary of the Möbius band, it is unsurprising that in-

454 tegration is significantly less accurate (Figure 6F). In future work, it will be interesting to
 455 consider which pairings of external to neural manifolds will provably permit accurate path
 456 integration.

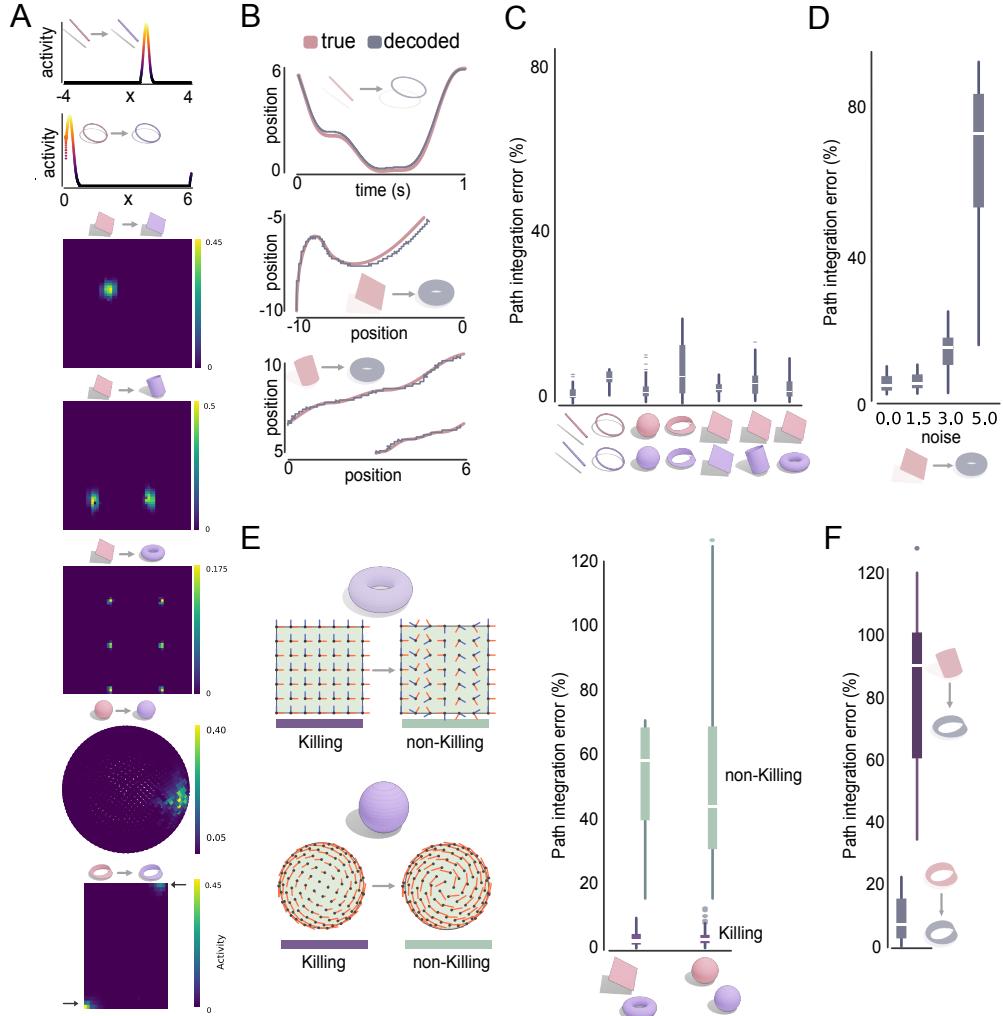


Figure 6: Numerical simulations of path integration performance with MADE path integrators. **A.** Tuning curves of single example neurons as a function of the external (latent) variable x . Insets show the manifold topologies of the external variable (red) and neural population states (blue): these pairings might be of identical manifolds, or e.g. a 2D Euclidean manifold in x could be mapped to a cylinder or torus, etc. in the neural population states. **B.** Example input trajectory (red) and decoded trajectory from the neural population response (blue). **C.** Decoding error across multiple simulations for various external-neural manifold pairs. Decoding error is shown as percentage of trajectory length over \mathcal{M} . Colored boxes show the interquartile range, white lines the mean, circles outliers and vertical lines the 95th percentile confidence interval. **D.** Same as **B** but for torus attractors with varying amounts of noise. **E.** Left: Killing and non-Killing weight offsets for the torus (top) and sphere (bottom). Right: Same as **C** for integrators correctly constructed with Killing weight offsets, and with the non-Killing weight offsets from the left. **F.** Same as **C** for Möbius to Möbius (left) and cylinder to Möbius mappings (right).

457 Discussion

458 *Summary* Here, we have presented MADE, a mathematical theory and recipe for con-
 459 structing biologically plausible neural CANs and integrator networks of desired topologies
 460 and geometries, with single- or multi-bump tuning curves. The mathematical theory unifies
 461 existing biologically plausible continuous attractor and integrator models involving bump-like
 462 activation functions, which emerge as specific cases of the MADE theory.

463 The theory provides a first-principles derivation showing that multiple copies of a basic

464 network must be coupled together for integration with biological constraints, in part to relieve
465 demands for rapid synaptic modulation and in part to remove the need for velocity estimating
466 regions from knowing the full nonlinear structure and current state of the integrator network.
467 It also predicts that manifolds without a flat metric will require an overcomplete set of network
468 pairs in the form of QAN networks, relative to the intrinsic dimensionality of the manifold:
469 thus, integration on a two-dimensional spherical surface requires more than 2 QAN pairs.

470 We envision MADE to be useful to distinct fields: for deep-learning models that might
471 require accurate low-dimensional neural network attractors and integrators, and for neuro-
472 science, where MADE provides de novo models and novel circuit-level mechanistic predictions
473 for the structure of other possible integrators in brain that may be uncovered in the future.

474 Indeed, given recent discoveries that path-integrating neural circuits generalizably repre-
475 sent multiple cognitive variables, it is likely that such circuits are used by the brain to perform
476 cognitive tasks in which variables of interest are not directly observed and only information
477 about their rate of changes is available (e.g., mental object rotation) [25, 75, 27]. MADE mod-
478 els could then act as test beds to generate mechanistic hypotheses for the network dynamics
479 underpinning integration computation in such cognitive tasks.

480 *Activity bumps and tuning curves* MADE provides a basic prescription for the construc-
481 tion of continuous attractor and integrator networks of a desired dimension and topology. We
482 numerically implemented a particular (Gaussian) kernel shape to illustrate the framework.
483 The shape of the population activity bumps that result will depend on the kernel shape,
484 which can be varied and selected as desired, according to the constraints supplied by our
485 theory. Recent theoretical work on symmetry breaking for pattern formation also suggests
486 that the set of potential kernels forms a large function space.

487 The tuning curve shapes of single cells depends both on the population activity bump shape
488 as well as on the mapping from the external variable manifold to the internal neural state
489 space manifold. As we have seen, if the external manifold is unbounded in some dimension
490 but the internal representation is compact and periodic, then the spatial tuning curve will be
491 periodic in that dimension. More subtle details of the mapping can affect the geometry of the
492 periodic mapping, as we have described above.

493 We have focused our illustrations on simple and non-trivial manifolds of intrinsic dimension
494 ≤ 2 for visualization and convenience. However, the theory and recipe for continuous attractor
495 and integrator network construction generalizes in a straightforward manner to manifolds of
496 higher dimension and different topologies.

497 *Related work* Computational models first described attractor networks [76, 36, 34, 35,
498 15, 11] and the mechanisms by which they could enable velocity integration [34, 47, 35,
499 15, 11, 46, 48, 18] long before experimental data verified the existence of such mechanisms.
500 Intriguingly and surprisingly, in every case experimentally probed to date, the proposed neural
501 circuit models closely resemble the hand-designed attractor models [40, 3, 42, 43, 44, 38, 15,
502 37, 14, 13]. Why is this the case? Presumably this match arises because the models were
503 minimal in the sense that they implemented the essential elements and symmetries required
504 to form the desired attractor, and circuits in the brain evolving under efficiency pressures
505 arrived at similarly minimal models. MADE adopts a very similar mathematically minimal
506 approach, recovering all of the known integrator models with bump-like tuning (except for
507 the oculomotor integrator, which does not have bump-like responses).

508 An alternative approach to building models of integrating circuits in brains is to train
509 artificial neural networks to perform tasks requiring integration [50, 49, 51]. After training, the
510 networks' solution is analyzed to reverse engineer the relation between network connectivity,
511 neural dynamics and task performance [50, 77, 78, 79]. However, such approaches often fail to
512 provide novel testable predictions or interpretable mechanisms to guide further experimental
513 investigations, unless there was already a hand-crafted model available to which the trained

514 network could be compared.

515 The network engineering approach [80, 81, 82, 83, 84, 82, 85, 86, 87] constructs circuits
516 starting from the detailed desired dynamics of a system (precise states, fixed points, or specific
517 tuning curves), then directly searching or solving for some network connectivity with those
518 dynamics. Typically, these works further constrain the problem to make it well-posed by
519 searching for low-rank weights or the lowest-dimensional embedding space for the dynamics
520 while satisfying the desired properties. These methods are complementary to our approach:
521 they permit construction of a broader set of dynamical systems, for instance trajectories
522 ending in discrete fixed points, stable and unstable fixed points, etc., while our focus is
523 specifically on biologically plausible continuous attractors that integrate. Conversely, those
524 approaches do not provide a framework for building biologically realistic continuous attractor
525 networks that integrate and lack known matches or easy interpretability to compare with
526 biological circuits in known cases.

527 In conclusion, MADE allows for easy generation of interpretable, mechanistic, models of
528 CAN networks that can integrate. We hope that MADE will endow researches with tools
529 required to generate detailed, testable, hypotheses about the neural underpinnings of integra-
530 tion in diverse settings and in various cognitive processes, accelerating our understanding of
531 the critical role that this class of computations play in many aspects of brain function and
532 allowing for easy incorporation of such circuits in deep learning applications.

533 References

- 534 [1] Mikail Khona and Ila R Fiete. Attractor and integrator networks in the brain. *Nat. Rev. Neurosci.*, 23(12):744–766, December 2022.
- 535
- 536 [2] James J Knierim and Kechen Zhang. Attractor dynamics of spatially correlated neural activity in the limbic system. *Annu. Rev. Neurosci.*, 35:267–285, March 2012.
- 537
- 538 [3] Rishidev Chaudhuri, Berk Gercek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.*, 22(9):1512–1520, September 2019.
- 539
- 540
- 541 [4] J S Taube, R U Muller, and J B Ranck, Jr. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *J. Neurosci.*, 10(2):420–435, February 1990.
- 542
- 543
- 544 [5] Jeffrey S Taube. The head direction signal: origins and sensory-motor integration. *Annu. Rev. Neurosci.*, 30:181–207, 2007.
- 545
- 546 [6] Luigi Petrucco, Hagar Lavian, You Kure Wu, Fabian Svara, Vilim Štih, and Ruben Portugues. Neural dynamics and architecture of the heading direction circuit in zebrafish. *Nat. Neurosci.*, 26(5):765–773, May 2023.
- 547
- 548
- 549 [7] Daniel B Turner-Evans, Kristopher T Jensen, Saba Ali, Tyler Paterson, Arlo Sheridan, Robert P Ray, Tanya Wolff, J Scott Lauritzen, Gerald M Rubin, Davi D Bock, and Vivek Jayaraman. The neuroanatomical ultrastructure and function of a biological ring attractor. *Neuron*, 108(1):145–163.e10, October 2020.
- 550
- 551
- 552
- 553 [8] Brad K Hulse, Hannah Haberkern, Romain Franconville, Daniel Turner-Evans, Shin-Ya Takemura, Tanya Wolff, Marcella Noorman, Marisa Dreher, Chuntao Dan, Ruchi Parekh, Ann M Hermundstad, Gerald M Rubin, and Vivek Jayaraman. A connectome of the drosophila central complex reveals network motifs suitable for flexible navigation and context-dependent action selection. *Elife*, 10:e66039, October 2021.
- 554
- 555
- 556
- 557
- 558 [9] Jonathan Green, Atsuko Adachi, Kunal K Shah, Jonathan D Hirokawa, Pablo S Magani, and Gaby Maimon. A neural circuit architecture for angular integration in drosophila. *Nature*, 546(7656):101–106, June 2017.
- 559
- 560
- 561 [10] Sung Soo Kim, Hervé Rouault, Shaul Druckmann, and Vivek Jayaraman. Ring attractor dynamics in the drosophila central brain. *Science*, 356(6340):849–853, May 2017.
- 562
- 563 [11] S C Cannon and D A Robinson. An improved neural-network model for the neural integrator of the oculomotor system: more realistic neuron behavior. *Biol. Cybern.*, 53(2):93–108, 1985.
- 564
- 565
- 566 [12] E Godaux and G Cheron. The hypothesis of the uniqueness of the oculomotor neural integrator: direct experimental evidence in the cat. *J. Physiol.*, 492 (Pt 2)(Pt 2):517–527, April 1996.
- 567
- 568
- 569 [13] A F Fuchs, C A Scudder, and C R Kaneko. Discharge patterns and recruitment order of identified motoneurons and internuclear neurons in the monkey abducens nucleus. *J. Neurophysiol.*, 60(6):1874–1895, December 1988.
- 570
- 571
- 572 [14] E Aksay, R Baker, H S Seung, and D W Tank. Anatomy and discharge properties of pre-motor neurons in the goldfish medulla that have eye-position signals during fixations. *J. Neurophysiol.*, 84(2):1035–1049, August 2000.
- 573
- 574

- 575 [15] H Sebastian Seung. Continuous attractors and oculomotor control. *Neural Netw.*,
576 11(7):1253–1258, October 1998.
- 577 [16] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser.
578 Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806,
579 August 2005.
- 580 [17] Hanne Stensola, Tor Stensola, Trygve Solstad, Kristian Frøland, May-Britt Moser, and
581 Edvard I Moser. The entorhinal grid map is discretized. *Nature*, 492(7427):72–78, De-
582 cember 2012.
- 583 [18] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network
584 models of grid cells. *PLoS Comput. Biol.*, 5(2):e1000291, February 2009.
- 585 [19] Stanley Heinze, Ajay Narendra, and Allen Cheung. Principles of insect path integration.
586 *Curr. Biol.*, 28(17):R1043–R1058, September 2018.
- 587 [20] Malcolm G Campbell, Alexander Attinger, Samuel A Ocko, Surya Ganguli, and Lisa M
588 Giocomo. Distance-tuned neurons drive specialized path integration calculations in me-
589 dial entorhinal cortex. *Cell Rep.*, 36(10), September 2021.
- 590 [21] P E Sharp, H T Blair, and J Cho. The anatomical and computational basis of the rat
591 head-direction cell signal. *Trends Neurosci.*, 24(5):289–294, May 2001.
- 592 [22] P E Sharp, Amanda Tinkelman, and Jeiwon Cho. Angular velocity and head direction
593 signals recorded from the dorsal tegmental nucleus of gudden in the rat: implica-
594 tions for path integration in the head direction cell circuit. *Behav. Neurosci.*, 115(3):571–588,
595 June 2001.
- 596 [23] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the
597 brain’s spatial representation system. *Annu. Rev. Neurosci.*, 31:69–89, 2008.
- 598 [24] J O’Keefe. A computational theory of the hippocampal cognitive map. *Prog. Brain Res.*,
599 1990.
- 600 [25] Dmitriy Aronov and David W Tank. Engagement of neural circuits underlying 2D spatial
601 navigation in a rodent virtual reality system. *Neuron*, 84(2):442–456, October 2014.
- 602 [26] Nathaniel J Killian, Michael J Jutras, and Elizabeth A Buffalo. A map of visual space
603 in the primate entorhinal cortex. *Nature*, 491(7426):761–764, November 2012.
- 604 [27] Alexandra O Constantinescu, Jill X O'Reilly, and Timothy E J Behrens. Organizing
605 conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468,
606 June 2016.
- 607 [28] James C R Whittington, David McCaffary, Jacob J W Bakermans, and Timothy E J
608 Behrens. How to build a cognitive map. *Nat. Neurosci.*, 25(10):1257–1272, October
609 2022.
- 610 [29] Honi Sanders, Matthew Wilson, Mirko Klukas, Sugandha Sharma, and Ila Fiete. Efficient
611 inference in structured spaces. *Cell*, 183(5):1147–1148, November 2020.
- 612 [30] Ling L Dong and Ila R Fiete. Grid cells in cognition: Mechanisms and function. *Annu.*
613 *Rev. Neurosci.*, 47(1):345–368, August 2024.

- 614 [31] Abhiram Iyer, Sarthak Chandra, Sugandha Sharma, and Ila R Fiete. Flexible mapping
615 of abstract domains by grid cells via self-supervised extraction and projection of gener-
616 alized velocity signals. In *The Thirty-eighth Annual Conference on Neural Information
617 Processing Systems*, 2024.
- 618 [32] Mengyu Liu, Aditya Nair, Nestor Coria, Scott W Linderman, and David J Anderson.
619 Encoding of female mating dynamics by a hypothalamic line attractor. *Nature*, pages
620 1–9, August 2024.
- 621 [33] Peter E Welinder, Yoram Burak, and Ila R Fiete. Grid cells: the position code, neural
622 network models of activity, and the problem of learning. *Hippocampus*, 18(12):1283–1300,
623 2008.
- 624 [34] K Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-
625 direction cell ensemble: a theory. *J. Neurosci.*, 16(6):2112–2126, March 1996.
- 626 [35] Xiaohui Xie, Richard H R Hahnloser, and H Sebastian Seung. Double-ring network
627 model of the head-direction system. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 66(4
628 Pt 1):041902, October 2002.
- 629 [36] R Ben-Yishai, R L Bar-Or, and H Sompolinsky. Theory of orientation tuning in visual
630 cortex. *Proc. Natl. Acad. Sci. U. S. A.*, 92(9):3844–3848, April 1995.
- 631 [37] H S Seung, D D Lee, B Y Reis, and D W Tank. Stability of the memory of eye position in
632 a recurrent network of conductance-based model neurons. *Neuron*, 26(1):259–271, April
633 2000.
- 634 [38] Ashwin Vishwanathan, Kayvon Daie, Alejandro D Ramirez, Jeff W Lichtman, Emre
635 R F Aksay, and H Sebastian Seung. Electron microscopic reconstruction of functionally
636 identified cells in a neural integrator. *Curr. Biol.*, 27(14):2137–2147.e3, July 2017.
- 637 [39] W E Skaggs, J J Knierim, H S Kudrimoti, and B L McNaughton. A model of the neural
638 basis of the rat’s sense of direction. *Adv. Neural Inf. Process. Syst.*, 7:173–180, 1995.
- 639 [40] Kijung Yoon, Michael A Buice, Caswell Barry, Robin Hayman, Neil Burgess, and Ila R
640 Fiete. Specific evidence of low-dimensional continuous attractor dynamics in grid cells.
641 *Nat. Neurosci.*, 16(8):1077–1084, August 2013.
- 642 [41] Kijung Yoon, Sam Lewallen, Amina A Kinkhabwala, David W Tank, and Ila R Fiete.
643 Grid cell responses in 1D environments assessed as slices through a 2D lattice. *Neuron*,
644 89(5):1086–1099, March 2016.
- 645 [42] Sean G Trettel, John B Trimper, Ernie Hwaun, Ila R Fiete, and Laura Lee Colgin. Grid
646 cell co-activity patterns during sleep reflect spatial overlap of grid fields during active
647 behaviors. *Nat. Neurosci.*, 22(4):609–617, April 2019.
- 648 [43] R J Gardner, L Lu, T Wernle, M B Moser, and E I Moser. Correlation structure of grid
649 cells is preserved during sleep. *Nat. Neurosci.*, 2019.
- 650 [44] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Ben-
651 jamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population
652 activity in grid cells. *Nature*, 602(7895):123–128, February 2022.
- 653 [45] Mark C Fuhs and David S Touretzky. A spin glass model of path integration in rat
654 medial entorhinal cortex. *Journal of Neuroscience*, 26(16):4266–4276, 2006.

- 655 [46] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-
656 Britt Moser. Path integration and the neural basis of the 'cognitive map'. *Nat. Rev.
657 Neurosci.*, 7(8):663–678, August 2006.
- 658 [47] A David Redish, Adam N Elga, and David S Touretzky. A coupled attractor model of
659 the rodent head direction system. *Network: computation in neural systems*, 7(4):671,
660 1996.
- 661 [48] Y Burak and I Fiete. Do we understand the emergent dynamics of grid cell activity? *J.
662 Neurosci.*, 26:9352–9354, September 2006.
- 663 [49] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr
664 Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degrif, Joseph Modayil, Greg
665 Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan
666 Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray
667 Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharrshan Kumaran. Vector-based
668 navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433,
669 May 2018.
- 670 [50] Christopher J Cueva and Xue-Xin Wei. Emergence of grid-like representations by training
671 recurrent neural networks to perform spatial localization. *Arxiv*, March 2018.
- 672 [51] Ben Sorscher, Gabriel C Mel, Samuel A Ocko, Lisa Giocomo, and Surya Ganguli. A
673 unified theory for the computational and mechanistic origins of grid cells. December
674 2020.
- 675 [52] A Nayebi, A Attinger, M Campbell, and others. Explaining heterogeneity in medial
676 entorhinal cortex with task-driven neural networks. *Advances in*, 2021.
- 677 [53] Akhilan Boopathy, Sunshine Jiang, William Yue, Jaedong Hwang, Abhiram Iyer, and Ila
678 Fiete. Breaking neural network scaling laws with modularity. *arXiv [cs.LG]*, September
679 2024.
- 680 [54] Qiya Liang, Ziming Liu, Mitchell Ostrow, and Ila Fiete. How diffusion models learn to
681 factorize and compose. *arXiv [cs.AI]*, August 2024.
- 682 [55] Si Wu, K Hamaguchi, and S Amari. Dynamics and computation of continuous attractors.
683 *Neural Comput.*, 20:994–1025, April 2008.
- 684 [56] Alexis Guanella, Daniel Kiper, and Paul Verschure. A model of grid cells based on a
685 twisted torus topology. *Int. J. Neural Syst.*, 17(4):231–240, August 2007.
- 686 [57] Albert Compte, Nicolas Brunel, Patricia S Goldman-Rakic, and Xiao-Jing Wang. Synap-
687 tic mechanisms and network dynamics underlying spatial working memory in a cortical
688 network model. *Cerebral cortex*, 10(9):910–923, 2000.
- 689 [58] John J Hopfield. Neurons with graded response have collective computational proper-
690 ties like those of two-state neurons. *Proceedings of the national academy of sciences*,
691 81(10):3088–3092, 1984.
- 692 [59] Mikail Khona, Sarthak Chandra, and Ila Fiete. Spontaneous emergence of topologically
693 robust grid cell modules: A multiscale instability theory. October 2021.
- 694 [60] Edward Ott. *Quasiperiodicity*, page 212–245. Cambridge University Press, 2 edition,
695 2002.

- 696 [61] Katsumi Nomizu. On local and global existence of killing vector fields. *Annals of Mathematics*,
697 pages 105–120, 1960.
- 698 [62] Sharief Deshmukh and Olga Belova. On killing vector fields on riemannian manifolds.
699 *Mathematics*, 9(3):259, 2021.
- 700 [63] John M Lee and John M Lee. *Smooth manifolds*. Springer, 2012.
- 701 [64] John H Wen, Ben Sorscher, Emily A Aery Jones, Surya Ganguli, and Lisa M Giocomo. One-shot entorhinal maps enable flexible navigation in novel environments. *Nature*,
702 635(8040):943–950, November 2024.
- 704 [65] John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.
- 705 [66] Louis Kang, Boyan Xu, and Dmitriy Morozov. State space discovery in spatial represen-
706 tation circuits with persistent cohomology. October 2020.
- 707 [67] Gard Spreemann, Benjamin Dunn, Magnus Bakke Botnan, and Nils A Baas. Using
708 persistent homology to reveal hidden information in neural data. October 2015.
- 709 [68] Robert Ghrist. Barcodes: The persistent topology of data. *Bull. Am. Math. Soc.*,
710 45(01):61–76, October 2007.
- 711 [69] Mehrdad Jazayeri and Srdjan Ostojic. Interpreting neural computations by examining
712 intrinsic and embedding dimensionality of neural activity. July 2021.
- 713 [70] Cátia Fortunato, Jorge Bennasar-Vázquez, Junchol Park, Joanna C Chang, Lee E Miller,
714 Joshua T Dudman, Matthew G Perich, and Juan A Gallego. Nonlinear manifolds underlie
715 neural population activity during behaviour. July 2023.
- 716 [71] Ege Altan, Sara A Solla, Lee E Miller, and Eric J Perreault. Estimating the dimension-
717 ality of the manifold underlying multi-electrode neural recordings. *PLoS Comput. Biol.*,
718 17(11):e1008591, November 2021.
- 719 [72] Anna V Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for
720 estimating intrinsic dimension. http://lcs1.mit.edu/papers/lit_mag_ros_2011.pdf.
721 Accessed: 2022-9-16.
- 722 [73] K Fukunaga and D R Olsen. An algorithm for finding intrinsic dimensionality of data.
723 *IEEE Trans. Comput.*, C-20(2):176–183, February 1971.
- 724 [74] Thomas F Banchoff and Stephen Lovett. *Differential geometry of curves and surfaces*.
725 CRC Press, 2022.
- 726 [75] Miriam L R Meister and Elizabeth A Buffalo. Neurons in primate entorhinal cortex
727 represent gaze position in multiple spatial reference frames. *J. Neurosci.*, 38(10):2430–
728 2441, March 2018.
- 729 [76] Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields.
730 *Biological cybernetics*, 27(2):77–87, 1977.
- 731 [77] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics
732 in high-dimensional recurrent neural networks. *Neural Comput.*, 25(3):626–649, March
733 2013.

- 734 [78] Niru Maheswaranathan, Alex H Williams, Matthew D Golub, Surya Ganguli, and David
735 Sussillo. Universality and individuality in neural dynamics across large populations of
736 recurrent networks. *Adv. Neural Inf. Process. Syst.*, 2019:15629–15641, December 2019.
- 737 [79] Rylan Schaeffer, Mikail Khona, Leenoy Meshulam, International Brain Laboratory, and
738 Ila Rani Fiete. Reverse-engineering recurrent neural network solutions to a hierarchical
739 inference task for mice. June 2020.
- 740 [80] Francesca Mastrogiovanni and Srdjan Ostojic. Linking connectivity, dynamics, and com-
741 putations in Low-Rank recurrent neural networks. *Neuron*, 99(3):609–623.e29, August
742 2018.
- 743 [81] Federico Claudi and Tiago Branco. Differential geometry methods for constructing
744 Manifold-Targeted recurrent neural networks. *Neural Comput.*, 34(8):1790–1811, July
745 2022.
- 746 [82] Chris Eliasmith. A unified approach to building and controlling spiking attractor net-
747 works. *Neural Comput.*, 17(6):1276–1314, June 2005.
- 748 [83] Ran Darshan and Alexander Rivkind. Learning to represent continuous variables in
749 heterogeneous neural networks. *Cell Rep.*, 39(1):110612, April 2022.
- 750 [84] Chris Eliasmith and Charles H Anderson. *Neural Engineering: Computation, Represen-
751 tation, and Dynamics in Neurobiological Systems*. MIT Press, 2003.
- 752 [85] Omri Barak and Sandro Romani. Mapping Low-Dimensional dynamics to High-
753 Dimensional neural activity: A derivation of the ring model from the neural engineering
754 framework. *Neural Comput.*, 33(3):827–852, March 2021.
- 755 [86] Eli Pollock and Mehrdad Jazayeri. Engineering recurrent neural networks from task-
756 relevant manifolds and dynamics. *PLoS Comput. Biol.*, 16(8):e1008128, August 2020.
- 757 [87] Manuel Beiran, Alexis Dubreuil, Adrian Valente, Francesca Mastrogiovanni, and Srdjan
758 Ostojic. Shaping dynamics with multiple populations in Low-Rank recurrent networks.
759 *Neural Comput.*, 33(6):1572–1615, May 2021.
- 760 [88] Seth D Axen, Mateusz Baran, Ronny Bergmann, and Krzysztof Rzecki. Manifolds.jl: An
761 extensible julia framework for data analysis on manifolds. June 2021.
- 762 [89] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework
763 for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.
- 764 [90] Matija Čufar. Ripserer.jl: flexible and efficient persistent homology computation in julia.
765 *Journal of Open Source Software*, 5(54):2614, 2020.

766 Methods

767 All simulations and figures were implemented in custom Julia code available at GeneralAt-
768 tractorsTheory. We will provide a minimal Python package for creating CANs and QANs
769 using MADE: MADE-Python.

770 CAN construction

771 In MADE, CAN engineering depends on computations of the pair-wise on-manifold dis-
772 tances between neurons in a lattice \mathcal{P} . Thus, we begin by specifying a set of n equally spaced
773 points on \mathcal{P} . For the Line attractor, $n = 256$ and \mathcal{P} was taken to be the interval $[-6, 6]$. For
774 the Ring attractor, $n = 256$ and \mathcal{P} was taken to be the interval $[0, 2\pi]$ with the two ends
775 identified (i.e. we ensured not to have a neuron at $\theta_i = 0$ and one at $\theta_i = 2\pi$). For all remain-
776 ing networks, $n = 48^2$ was used. The following rectangular intervals were used: for the plane
777 attractor $\mathcal{P} = [-10, 10] \times [-10, 10]$, cylinder: $\mathcal{P} = [-5, 5] \times [0, 2\pi]$, torus: $\mathcal{P}[0, 2\pi] \times [0, 2\pi]$,
778 Möbius band: $\mathcal{P} = [-2, 2] \times [0, 2\pi]$ and Klein Bottle: $\mathcal{P}[0, 2\pi] \times [0, 2\pi]$. For the sphere at-
779 tractor, the n points were chosen to be on a Fibonacci spiral on the unit sphere embedded in
780 \mathbb{R}^3 .

781 Next, to implement custom manifold-specific distance metrics d we used the Julia package
782 Distances.jl. The standard Euclidean metric was used for the line and plane attractor, for
783 the ring a one dimensional periodic Euclidean metric (period 2π) was used, for the torus a
784 two dimensional periodic Euclidean metric (period 2π in each direction) and for the Cylinder
785 a heterogeneous periodic and standard Euclidean metric for the periodic and non-periodic
786 dimensions respectively. For the sphere the great arc spherical distance for points on the
787 unit sphere (implemented in the Manifolds.jl package [88]) was used. For the Möbius band a
788 custom metric function was used to account for the non-orientable nature of the manifold.

789 For the Klein Bottle, a different approach was used. First, we defined an embedding of
790 the Klein Bottle in \mathbb{R}^4 mapping each lattice point $\theta = (u, v)$ to a point $q \in \mathbb{R}^4$:

$$\begin{aligned} q_1 &= (2 + \cos(v)) \cos(u) \\ q_2 &= (2 + \cos(v)) \sin(u) \\ q_3 &= \sin(v) \\ q_4 &= \sin(v) \cos\left(\frac{u}{2}\right) \end{aligned}$$

791 Next, we computed the pairwise Euclidean Distance in \mathbb{R}^4 for the embedded points and
792 selected the 8 nearest neighbors of each point. We then constructed a graph where each
793 node was a lattice point and two nodes were connected if one belong to the neighborhood
794 of the other. Each edge was assigned a weight equal to the Euclidean distance between the
795 two points. Thus, the graph structure was taken to represent the local topological structure
796 (connectivity) of the Klein Bottle. Given two points θ_i, θ_j then, their on-manifold distance
797 was given by summing the edge weights (local distances) along the shortest path on the graph
798 from the node corresponding to θ_i to the one corresponding to θ_j as a way to numerically
799 approximate the geodesic distance between them.

Following computation of pairwise distances, the connection weights between two neurons
was computed using as kernel function:

$$k(x) = \alpha \exp(-x^2/(2\sigma^2)) - \alpha$$

800 yielding strictly non-positive values for the connection strength. This gave a connectivity
 801 pattern characterized by global, long-distance inhibition, and no, or reduced, inhibition locally
 802 such that a localized pattern of activation on the neural lattice \mathcal{P} would remain localized and
 803 not result in activation of all neurons in the network. The parameters α, σ were varied based
 804 on the CAN topology and are indicated in table 1

	line	ring	plane	cylinder	torus	Möbius	Sphere	Klein Bottle
α	1	1	2.5	2.5	2.5	2.5	2.5	2.5
σ	1	1	25	25	2	2.5	40.5	150

Table 1: Kernel function parameters

805 CAN simulation

806 Network dynamics were approximate to discrete time using forward Euler integration with
 807 $\Delta t = 0.5\text{ms}$ using:

$$\mathbf{s}(t + \Delta t) = \frac{f[W\mathbf{s}(t) + b + \eta(t)] - \mathbf{s}(t)}{\tau}$$

808 where $\mathbf{s}(t)$ is a vector representing the activity of each neuron in the network at time
 809 t , $\tau = 5\text{ms}$ was used as time constant. The constant input $b = 0.5$ was used throughout.
 810 The term $\eta(t)$ was used to simulate Poisson noise in the recurrent dynamics, it represents a
 811 vector of length n whose entries are given by: $\eta_i(t) = \text{rand}(-0.5, 0.5) * \sigma_{noise} \sqrt{(s_i)}$ where
 812 $\sigma_{noise} \in \{0, 1.5, 3, 5\}$. Unless explicitly stated, $\sigma_{noise} = 0$ was used.

813 For each CAN, 2500 simulations of 25ms in duration were performed to generate data for
 814 the analysis of the activity manifold topology. We chose 25ms since we observed this to be
 815 sufficient for the network to settle into a steady state (i.e. one in which the network's activity
 816 does not change between simulation steps).

817 For the first 15ms of each simulation, the activity of neurons at a distance $d > 0.5$ from a
 818 selected neuron θ_0 (randomly selected for each simulation) was artificially set to 0 to induce
 819 the formation of a stable bump of activity around θ_0 to promote uniform coverage of the entire
 820 manifold. The final activation vector $\mathbf{s}(T)$ for each simulation was then stored for subsequent
 821 analysis. For the torus attractor network, additional simulations were performed varying the
 822 noise parameter to assess the effect of noise on the attractor dynamics.

823 Attractor manifold analysis.

824 The final activation vector of each of 2500 CAN simulations for each manifold were collected
 825 into a matrix of dimensionality $n \times 2500$ with n being the number of neurons in the network.
 826 For networks other than the line and ring attractors in which $n > 400$ a first dimensionality
 827 reduction step using PCA was performed to reduce the data to a 400×2500 dimensional
 828 matrix. Then, further reduction to three dimensional data for visualization (Figure 4) was
 829 achieved using Isomap [89]. To reduce computation Isomap was fitted to 10% randomly se-
 830 lected data points and then used to embed the entire dataset for visualization. For subsequent
 831 Topological Data Analysis (TDA) point cloud data was subjected to PCA dimensionality re-
 832 duction to generate a 200×2500 data matrix and Isomap was then used to further reduce
 833 dimensionality to 10 (Isomap fitted to 10% of the data) [3].

834 **Topological data analysis**

835 To perform persistent homology analysis the Julia packages Ripserer.jl and PersistenceDiagrams.jl [90] were used. To reduce computation, the TDA filtration was computed using
836 a subset of randomly selected data points (20% of the entire dataset) to obtain the persis-
837 tence diagrams shown in Figure 4 A. Only intervals with a lifespan > 7 were kept to remove
838 features due to noise and the number of persistent intervals of each dimension (up to two
839 dimensional cavities) were counted to obtain Betti numbers, which were then compared with
840 those expected for manifolds of the given topology.
841

842 **Visualizing neural tuning curves**

843 To visualize neural activation turning curves over \mathcal{N} in Figure 4, we used PCA and ISOMAP
844 to reduce the dimensionality of neural activity to three dimensions. We thus obtained 2500
845 low dimensional points which we colored according to the activity of one selected neuron in
846 the corresponding neural state. To visualize activity over the neural lattice \mathcal{P} , we started by
847 selecting one random neural state from the 2500 simulations. Then, we uniformly sampled \mathcal{P}
848 and for each location $\theta_i \in \mathcal{P}$ we identified the closest neuron in the CAN (by coordinates).
849 We then colored each point in \mathcal{P} according to the activation of the closest neuron.

850 **Intrinsic manifold dimensionality analysis**

851 To estimate the manifold's intrinsic dimensionality all data points in the n -dimensional state
852 space were utilized. Pairwise Euclidean distance between each data point was computed to
853 obtain each data point's k nearest neighbors (using the NearestNeighbors.jl package). While
854 Euclidean distance in state space does not necessary match on-manifold geodesic distance
855 on \mathcal{N} in general, on a sufficiently small scale a manifold's Euclidean structure makes this
856 approximation acceptable. Next, 250 random data points (10% of the total) were selected
857 for estimation of local dimensionality in their neighborhood. For each, the k closest points
858 were selected and PCA fitted to the data. The number d of principal components required to
859 explain at least 75% of the data was used as estimate of local dimensionality and the manifold's
860 intrinsic dimensionality was taken to be the average across repeats. Thus, the dimensionality
861 estimation procedure depended on two hyperparameters: k and the percentage of variance
862 explained. Preliminary tests on artificially generated data with known dimensionality and
863 variable Gaussian noise were used to select the parameters used here, and we've found the
864 estimated intrinsic dimensionality to be robust across a wide range of parameters values (data
865 not shown). For the analyses shown here we used $k = 500$ throughout. Our preliminary tests
866 showed that much smaller values of k resulted in noisy estimates (especially in the face of noise)
867 and very large values of k led to an overestimation of the manifold intrinsic dimensionality
868 (likely due to the higher global embedding dimensionality).

869 **Attractor dynamics analysis**

870 To explicitly quantify attractor dynamics, a torus network was constructed as described above
871 and simulated without external stimuli for a simulation time of 250ms (given 100 random
872 initializations). Next, the network's state was perturbed by addition with a random vector v
873 of the same dimensionality as the network activity. For each of 100 simulations the vector was
874 chosen to have a random orientation but fixed magnitude. The magnitude was computed to be
875 50% of the average distance between states on the torus manifold and the origin. Following the
876 stimulus, the simulation was continued for 750ms more for a total simulation time of 1000ms.
877 Data for each simulation was collected for the analysis of on- vs off-manifold dynamics. For

878 each repeat the state just prior to stimulus application was used as seed for local PCA using
879 k-nearest points from the point cloud data used for previous estimation of manifold topology,
880 as described above (i.e. the steady states from previous simulations without the inputs were
881 used to estimate the tangent plane to the manifold). The top two principal components were
882 retained as approximation of the manifold's tangent plane. The neural trajectory was then
883 decomposed into on-manifold and off-manifold components by projection onto the tangent
884 plane and remaining $N - 2$ dimensions. The euclidean distance in each subspace from the
885 initial condition over time was then computed to asses drift following stimulus application
886 and averaged across repeats.

887 QAN construction

888 To construct quasi-periodic attractor networks for integration, first a choice of variable (\mathcal{M})
889 and neural lattice (\mathcal{P}) manifolds was made, ensuring that an identity or periodic mapping
890 existed between the two (unless explicitly stated otherwise). Next, a map $\pi : \mathcal{M} \rightarrow \mathcal{P}$ and
891 its inverse π^{-1} were defined (e.g. mapping each point on the plane, \mathcal{P} , to a corresponding
892 point on the torus, \mathcal{N}). To compute connection weights in each QAN, points on \mathcal{P} were
893 selected as before and the same distance metrics and kernel functions were used applying
894 an offset to the neurons' coordinates during distance computation. For the Line and Ring
895 attractor, two QANs were constructed using the offset vectors $\delta_{\pm m} = \pm 0.15$. For the Plane,
896 Cylinder, Torus, Möbius attractors four QANs were constructed using $\delta_{\pm 1} = [\pm 0.25, 0]$ and
897 $\delta_{\pm 1} = [0, \pm 0.25]$ as offset vectors. For the sphere attractor, six QANs were constructed using
898 as offset vectors $\delta_{1\pm} = \pm[0, -z, y]$, $\delta_{2\pm} = \pm[z, 0, -x]$ and $\delta_{3\pm} = \pm[-y, x, 0]$ where $[x, y, z]$
899 represents coordinates on the unit sphere embedded in three dimensional euclidean space.
900 For the sphere, therefore, the offset vector magnitude varied as a function of position on
901 the sphere to ensure that Killing vector fields were used (which are constant for the other
902 manifolds used). The same vectors were used to compute the velocity-dependent stimulus
903 $\omega_{\pm m}(\dot{x})$ to each QAN. For some simulations, non-Killing vector fields where used. In the
904 Torus, the offset vectors $\hat{\delta}_{\pm 1} = \pm[\cos(\theta), \sin(\theta)]$, where $\theta = [x, y]$, and $\hat{\delta}_{\pm 2} = R\hat{\delta}_{\pm 2}$ where
905 used (where R is the rotation matrix $R = [[0, 1], [-1, 0]]$). For the sphere, the same vectors as
906 above were used, except they were normalized to be of unit length everywhere on the sphere
907 (except where they vanished).

908 QAN dynamics simulation

Similarly to CANs, network dynamics were simulated using forward Euler integration. For each QAN in a network performing integration the discrete time dynamics were:

$$\mathbf{s}_{\pm m}(t + \Delta t) = \frac{f[W_{\pm m}\mathbf{s}(t) + \omega_{\pm m}(\dot{x}(t)) + b] - \mathbf{s}(t)}{\tau}$$

909 where $\mathbf{s}(t) = \sum_{\pm m} \mathbf{s}_{\pm m}$ and $\omega_{\pm m}(\dot{x}(t)) = Jx(t) \cdot \delta_{\pm m}$ where J is the Jacobian of the map
910 π evaluated at a point $\hat{x} \in \mathcal{M}$ decoded from the neural state \mathbf{s} and $\delta_{\pm m}$ is the offset vector
911 at a point $\theta \in \mathcal{P}$ corresponding to the location of the neuron with highest activation in the
912 network. The velocity input $\dot{x}(t)$ was computed by simulating the random walk of a point
913 particle in the variable manifold \mathcal{M} .

914 To assess integration accuracy we generated 50 random trajectory (each corresponding to
915 1 second of simulated time) and simulated integration with the QANs. For each simulation,
916 a trajectory $\hat{\gamma} \in \mathcal{M}$ was decoded from neural activity and compared to the input trajectory
917 $\gamma \in \mathcal{M}$. The simulation error was computed as a fraction of the trajectory length and was
918 given by:

$$\ell = \frac{\sum_t d(\gamma(t), \hat{\gamma}(t))}{L_\gamma}$$

where d is a metric function for \mathcal{M} as described above and L_γ the trajectory length of γ on \mathcal{M} . More precisely, for decoding we used $\hat{\gamma}_t = \theta_t^*$ where $\theta_t^* = \theta_{\text{argmax}(\mathbf{s}(t))} + \omega$. Here ω is a correction factor used only when \mathcal{M} and \mathcal{N} had different topologies such that \mathcal{M} had non-periodic dimensions (plane) and \mathcal{N} had periodic dimensions (torus). At each decoding step, we added or subtracted 2π to ω when necessary to account for the neural state "wrapping around" the boundary dimension(s). Here ω is a d -dimensional vector and each value is set to 0 for non-periodic dimensions in \mathcal{N} and is $k2\pi$ for some integer k otherwise.

To generate the trajectories, we first defined a set of $2d$ vector fields Φ_i over \mathcal{M} , each corresponding to a vector field Ψ_i over \mathcal{P} . Then, we generated smoothly varying vectors A_t^i such that at each time t the velocity vector \dot{x}_t was given by $\sum A_t^i \Phi_i$. These weights vectors were given by the sum of two sine waves with random periods and scaled to have amplitude < 0.1 . We then computed \mathbf{X} , the trajectory over \mathcal{M} by, at each time step, computing \dot{x}_t and $x_t = x_{t-1} + \dot{x}_t/dt$ (with dt the simulation time step, 0.5). Finally, we computed \mathbf{V} the set of inputs to the QANs. For each time point t , the input v_j to the j^{th} QAN was given by $\pi_*(\dot{x}_t) \cdot \Psi_j$ where π_* was the push forward of the map $\pi : \mathcal{M} \rightarrow \mathcal{N}$.

In some conditions we artificially injected Poisson noise in the QAN neural dynamics as described previously to assess the effect of noise on path integration.

Neural tuning curves on \mathcal{M}

To visualize neural tuning curves with respect to \mathcal{M} in Figure 6, we generated a single trajectory densely sampling from \mathcal{M} (5-10 seconds of simulated time). After simulating path integration, we selected one random neuron to visualize its tuning curve. The visualization method varied based on the manifold topology. For one dimensional manifolds we simply plotted decoded value, x , against the neuron's activity. For most two dimensional manifolds, with the exception of the sphere, we generated a heatmap by binning x and quantifying the average neuron's activity for samples from each bin. A small amount of noise was added to x before binning to improve visualization. For the sphere, we first sampled 2000 points uniformly distributed on \mathcal{M} . Then, for each point we looked at the closest decoded value. We then colored each point on the sphere according to the neural activity value at the corresponding sample.

Non Killing fields and non-periodic manifold mapping

To demonstrate that path integration depended on the weight offset vector fields Ψ being Killing fields we generated two variants of the torus and sphere QANs. For the torus, we kept the magnitude and relative orientation of the offset vector fields constant, but gradually rotated their position by an angle $\cos(\theta_1)$ (i.e. only as a function of position along one manifold dimension). This ensured that vector fields at the boundary conditions were identical, as expected. For the sphere, we started with the Killing vector fields we had, and simply normalized each vector such that all vectors had constant length. We then ran 50 simulations using random trajectories as described previously.

To assess path integration when no trivial or periodic mapping between \mathcal{M} and \mathcal{N} existed, we performed path integration simulations with \mathcal{M} as a cylinder and \mathcal{N} as a sphere. We used the same procedure described above to generate 50 random trajectories over the cylinder and computing the corresponding velocity vectors over \mathcal{P} .

961 Supplementary Information

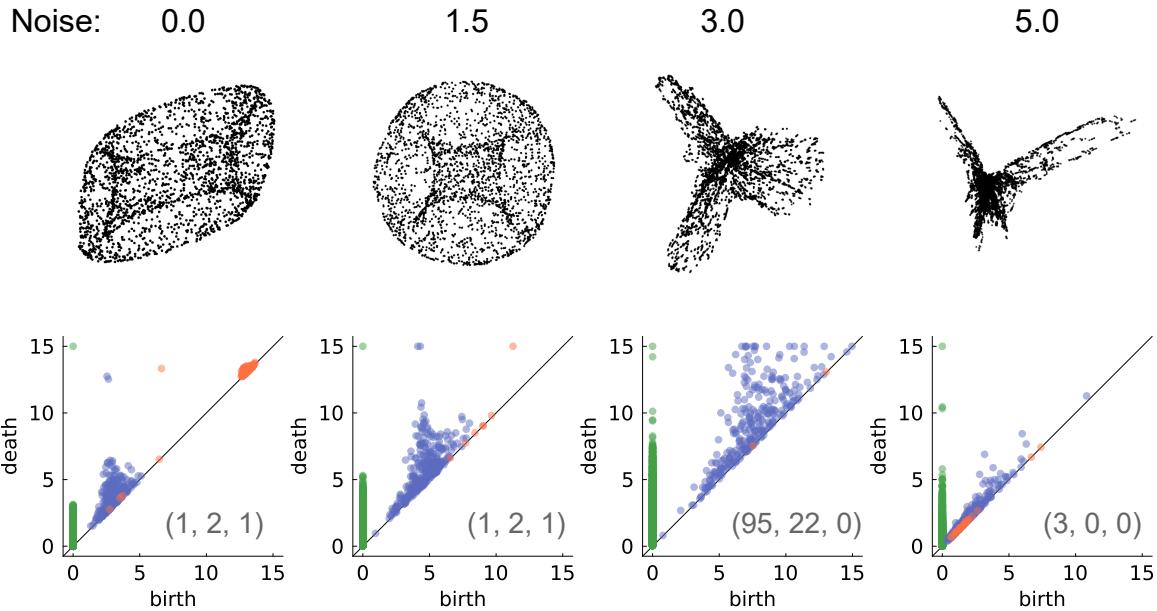


Figure S1: Torus CAN activity manifold (top) and persistence diagram (bottom) for varying noise intensity levels (columns).

962 1 Kernels constructed through distance metrics produce sin- 963 gle bump states

964 Here we estimate the conditions on the interactions that lead to the formation of bump states
965 on the lattice of neurons, \mathcal{P} .

966 As earlier, consider an interaction weight matrix $W(\theta, \theta') = k(d(\theta, \theta'))$. We rewrite the
967 kernel k as $k(d) = -k_0 + k_1(d)$, where $k_1(d) \rightarrow 0$ as $d \rightarrow \infty$ and $k_1(0) = k_0 > 0$; and
968 correspondingly write $W(d(\theta, \theta')) = -W_0 + W_1(d(\theta, \theta'))$. We assume that the kernel k has a
969 length scale σ , i.e., $k_1(d) \approx 0$ for $d \geq \sigma$.

970 Since σ is the only spatial scale being introduced in the dynamics, we qualitatively expect
971 the localized bump states will have a scale of $\mathcal{O}(\sigma)$. If σ is much smaller than the distances over
972 which the manifold \mathcal{P} has curvature, \mathcal{P} will be approximately flat within a ball V_σ centered
973 on any $x \in \mathcal{P}$. In this approximation, the conditions for the formation of a stable bump state
974 are the same as those for the formation on a bump state on a globally flat manifold.

975 To examine the conditions for the existence of a bump state, we will first calculate the
976 homogeneous steady state supported by Eq. 1. Next, we note that since W is symmetric
977 in this case, thus Eq. 1 can be described through an energy function[58], and thus a stable
978 steady state must exist. If the homogeneous state is unstable, there must then exist a stable
979 symmetry broken state of the system. If this symmetry broken state is localized, we refer to
980 it as the bump state.

The homogeneous steady state $s(x) = s_0$ must satisfy

$$\begin{aligned} s_0/\tau &= s_0 \int W(\theta - \theta') d\theta' + b \\ &= s_0 k_0 [\bar{k} - V] + b, \end{aligned}$$

981 where V is the volume of the manifold, $\int d\theta$, and $\bar{k} = \int k_1 d\theta$. Rearranging, we obtain

$$s_0 = \frac{b}{1/\tau + k_0[V - \bar{k}]} \quad (11)$$

982 Since the kernel k_1 is supported on a small volume of the entire manifold, $V > \bar{k}$, and thus
983 the right-hand side of Eq. 11 is positive, consistent with the assumed rectifying nonlinearity
984 f of Eq. 1.

985 To examine the stability of this homogeneous state, consider a small perturbation, $s(x, \theta) =$
986 $s_0 + \exp(\alpha(\omega)t + i\omega \cdot \theta)$ to Eq. 1. Following the analysis in Ref. [59], we obtain

$$\alpha(\omega) = \mathcal{F}[W](\omega) - 1/\tau, \quad (12)$$

where $\mathcal{F}[W]$ is the Fourier transform of the interaction W .

$$\begin{aligned} \mathcal{F}[W](\omega) &= \int W(\theta) \exp(i\omega\theta) d\theta \\ &= \int W_1(\theta) \exp(i\omega\theta) d\theta - 2\pi W_0 \delta(\omega), \\ &= \mathcal{F}[W_1] - 2\pi W_0 \delta(\omega) \end{aligned}$$

987 where $\delta(\omega)$ is the Dirac delta function, obtained from the Fourier transform of a constant.
988 Thus, the homogeneous steady state will be unstable if $\mathcal{F}[W](\omega) > 1/\tau$ for some ω . Since $\alpha(\omega)$
989 denotes the rate of exponential growth, the maxima of Eq. 12 will determine the dominant
990 growing mode. If $\mathcal{F}[W]$ were maximized at $|\omega| > 0$, then the growing perturbation would have
991 a periodic component, and would thus likely not form a localized mode. Instead, if $\mathcal{F}[W_1](\omega)$
992 were maximized at $\omega = 0$, then $\mathcal{F}[W](\omega)$ will be maximized at $\omega \rightarrow 0$ ($\mathcal{F}[W](\omega)$ cannot be
993 maximized strictly at $\omega = 0$ itself due to the $-2\pi W_0 \delta(\omega)$ contribution to $\mathcal{F}[W](\omega)$). In this
994 case, the growing perturbation will be unimodal, likely leading to the formation of a localized
995 state.

996 Thus, for the formation of a stable bump state on a general manifold, we obtain two
997 requirements: First, the Fourier transform of the kernel $k_1(d)$ must be maximized at $\omega = 0$;
998 and second, this maximum must be larger than $1/\tau$. If we are solely interested in interaction
999 shapes that lead to bump formation, we assume we have freedom to rescale the interactions.
1000 Thus, if a positive maximum is attained at $\omega = 0$ a rescaling can always make this maximum
1001 larger than $1/\tau$. Thus, we primarily focus on the first requirement.

While we do not provide an exhaustive classification of interaction kernels k_1 whose Fourier
transforms are maximized at zero, we provide a broad sufficient condition — if $k_1(d) \geq 0$ for
all d , then its Fourier transform will be maximized at zero. This can be proved as:

$$\mathcal{F}[k](\omega) = \int k(d) \exp(i\omega d) dd \leq \int k(d) |\exp(i\omega d)| dd = \int k(d) dd = \mathcal{F}[k](0).$$

1002 Thus, we finally conclude that, up to a rescaling of the strength of the interaction, an
1003 interaction $W(d(\theta, \theta'))$ will lead to the formation of a bump state if it can be rewritten as
1004 $W(d(\theta, \theta')) = k_1(d(\theta, \theta')) - k_0$ for: $k_0 \geq 0$; a kernel k_1 that satisfies $k_1(d) \geq 0$ and $k_1(d) \rightarrow 0$
1005 for $d \geq \sigma$; and sufficiently small σ over which the manifold \mathcal{P} is approximately flat.

1006 2 Manifold of single bump states \mathcal{N} is isometric to manifold 1007 of neural lattice \mathcal{P}

1008 Here we will show that the manifold \mathcal{N} of neural activity, formed through single bump states at
1009 each point of the neural lattice \mathcal{P} , is isometric to \mathcal{P} . Specifically, we provide a distance metric

1010 d_N on the manifold \mathcal{N} , such that (\mathcal{N}, d_N) is isometric to (\mathcal{P}, d_P) , where d_P represents the
 1011 geodesic distance considered as the distance metric during the MADE construction described
 1012 in the main text.

1013 While we will not prove this in complete generality for any \mathcal{P} , we will assume that if \mathcal{P}
 1014 has a sufficiently large separation of lengthscales (as assumed in the previous section), it will
 1015 suffice to show this result for \mathcal{P} given as the flat Euclidean manifold \mathbb{R}^n (and correspondingly,
 1016 d_P being the usual L_2 metric).

1017 To prove the existence of an isometry, we first argue that \mathcal{N} and \mathcal{P} are diffeomorphic. In
 1018 Apx. 1, we argued that the prescribed connectivity kernel leads to the formation of activity
 1019 bump states centered at any $x \in \mathcal{P}$. Define the function f from \mathcal{P} to \mathcal{N} to characterize
 1020 the shape of the activity bump, i.e., for any $x_0 \in \mathcal{P}$, we let $f_{x_0}(x) = f(x - x_0)$ be the
 1021 shape of the activity bump centered at x_0 . Since these activity bump states are generated
 1022 through radially symmetric kernel interaction functions, the bump states $f(x - x_0)$ must also
 1023 be radially symmetric, i.e., $f(x - x_0) = F(|x - x_0|)$. In this case, we can see that $\Phi : x \rightarrow f_x$
 1024 is now a diffeomorphism, since it is a smooth function and has a smooth inverse (the inverse
 1025 map is simply computing the center of the radially symmetric activity bump).

1026 Next, we examine candidate metrics on \mathcal{N} that may lead to an isometry with (\mathcal{P}, L_2) .
 1027 Note that a direct L_2 norm on \mathcal{N} does not suffice, since for sufficiently distant x_0 and x_1 ,
 1028 the distance between f_{x_0} and f_{x_1} given by $\sqrt{\int |f(x - x_0) - f(x - x_1)|^2 dx}$ is approximately
 1029 $\sqrt{\int 2|f(x)|^2 dx}$. Thus the distance between f_{x_0} and f_{x_1} is bounded, whereas the distance
 1030 between x_0 and x_1 is not, indicating that there cannot exist a direct isometry.

Instead, we construct here a metric of intrinsic length induced by the Riemannian metric
 on the tangents of \mathcal{N} . For any two vectors $u(x)$ and $v(x)$ in $T_s \mathcal{N}$, the tangent space of \mathcal{N} at
 s . Define the Riemannian metric as $g(u, v) = \langle u, v \rangle = \int uv dx$. Then, for any path $\gamma(t) \in \mathcal{N}$,
 we can define the length of the path $L[\gamma(t)]$ as

$$L[\gamma(t)] = \int |\gamma'(t)| dt,$$

1031 where the norm of a tangent vector γ' is defined as $\sqrt{g(\gamma', \gamma')}$. This can now be used to define
 1032 the geodesic metric between f_{x_0} and f_{x_1} on \mathcal{N} given as the infimum of the lengths of all paths
 1033 between f_{x_0} and f_{x_1} . Here we will show that under this geodesic metric, the spaces (\mathcal{N}, d_N)
 1034 and (\mathcal{P}, d_P) are isometric. Specifically, we will show that the metric tensor (the Riemannian
 1035 metric computed for coordinate basis vectors) is proportional to identity, the metric tensor
 1036 for flat Euclidean space.

Assume that \mathcal{N} is an n dimensional manifold. Let (x^1, \dots, x^n) be a coordinate chart in
 the neighborhood of a bump state $f_{x_0} = f(x - x_0)$. A basis for the tangent space is then
 given by the differentials $\{\partial/\partial x^1, \dots, \partial/\partial x^n\}$. Note that since $f(x)$ is radially symmetric,
 $f(x) = F(|x|)$, the basis vectors can be simplified as

$$\frac{\partial f(x)}{\partial x^i} = F'(|x|) \frac{\partial |x|}{\partial x^i} = F'(|x|) \frac{x^i}{|x|}.$$

We can now compute the metric tensor $g_{ij} = g(\partial/\partial x^i, \partial/\partial x^j)$

$$\begin{aligned} g_{ij} &= \int_{\mathbb{R}^n} \frac{\partial f(x - x_0)}{\partial x^i} \frac{\partial f(x - x_0)}{\partial x^j} dx \\ &= \int_{\mathbb{R}^n} \frac{\partial f(y)}{\partial y^i} \frac{\partial f(y)}{\partial y^j} dy \end{aligned} \tag{13}$$

$$= \int_{\mathbb{R}^n} F'(|y|)^2 \frac{y^i y^j}{|y|^2} dy, \tag{14}$$

1037 where Eq. 13 is obtained by performing the change of variables $y = x - x_0$. From Eq. 14 we
 1038 can make two crucial observations: first, since the integrand is odd in y^i and y^j , thus $g_{ij} = 0$
 1039 for $i \neq j$; second, g_{ii} is independent of x_0 , and by symmetry is also independent of i — it is
 1040 entirely determined by the shape of the bump state $F(|x|)$. Thus, the metric tensor g_{ij} has
 1041 constant entries on the diagonal, and zero on the off-diagonal elements, i.e., g is proportional
 1042 to the identity matrix. We denote this proportionality constant as α .

1043 The length of an infinitesimal line element ds is then given as $ds^2 = \sum_{i,j} g_{ij} dx^i dx^j =$
 1044 $\alpha \sum_i (dx^i)^2 = |dx|^2$. The length of a path γ from f_{x_0} to f_{x_1} is then simply $\int |\gamma'(t)| dt =$
 1045 $\sqrt{\alpha} \int |dx|$, which is the Euclidean path length from x_0 to x_1 scaled by $\sqrt{\alpha}$. Thus, the geodesic
 1046 metric from f_{x_0} to f_{x_1} is the infimum of Euclidean path lengths, i.e., the Euclidean straight-
 1047 line distance $\sqrt{\alpha} |x_0 - x_1|$. We can additionally redefine a new metric \tilde{g} on the tangent space
 1048 as g/α , leading to the new geodesic distance to be exactly the Euclidean distance $|x_0 - x_1|$.

1049 Thus, under the approximation of \mathcal{P} being treated as a flat space without curvature at
 1050 scales smaller than σ , the metric space (\mathcal{N}, d_N) is thus isometric to the metric space (\mathcal{P}, d_P) .

1051 3 External velocities ignorant about network structure and 1052 state require shifted-kernel networks to control bump flow

1053 In this section, for analytical simplicity, we will ignore the neural transfer function nonlinearity
 1054 f .

1055 The fixed points resulting from symmetric kernels in Eq. 1 satisfy:

$$s(\theta) = \int W(\theta - \theta') s(\theta') d\theta' + b, \quad (15)$$

where $s(\theta)$ denotes an activity bump centered at any point in \mathcal{P} . Consider two such activity bump states: $s_0(\theta)$ centered at θ_0 , and a nearby state s_ϵ centered at $\theta_0 - \epsilon$, i.e., $s_\epsilon(\theta) = s_0(\theta + \epsilon)$. For a neural state $s(\theta)$ to move from s_0 to s_ϵ in time Δt , the time derivative $\partial s / \partial t$ must equal

$$\begin{aligned} \frac{\partial s(\theta, t)}{\partial t} &= \frac{s(t + \Delta t) - s(t)}{\Delta t} \\ &= \frac{s_\epsilon(\theta) - s_0(\theta)}{\Delta t} \\ &= \frac{s_0(\theta + \epsilon) - s_0(\theta)}{\Delta t} \\ &\approx \frac{\epsilon}{\Delta t} \frac{\partial s_0(\theta)}{\partial \theta} \end{aligned} \quad (16)$$

We can use Eq. 15 to evaluate this space derivative as

$$\begin{aligned} \frac{\partial s_0(\theta)}{\partial \theta} &= \int \frac{\partial W(\theta - \theta')}{\partial \theta} s(\theta') d\theta', \\ &\approx \int \frac{[W(\theta - \theta' + \delta) - W(\theta - \theta')]}{\delta} s(\theta') d\theta', \\ &= \frac{1}{\delta} \int [W_\delta(\theta - \theta') - W(\theta - \theta')] s(\theta') d\theta', \\ &= \frac{1}{\delta} \left[\int W_\delta(\theta - \theta') s(\theta') d\theta - \int W(\theta - \theta') s(\theta') d\theta' \right], \\ &= \frac{1}{\delta} \left[\int W_\delta(\theta - \theta') s(\theta') d\theta' - (s_0(\theta) - b) \right]. \end{aligned}$$

1056 where W_δ represents a kernel with a small offset δ , i.e., $W_\delta = W(\theta - \theta' - \delta)$. We can insert
 1057 this in Eq. 16 to obtain

$$\frac{\partial s(\theta, t)}{\partial t} = \frac{\epsilon}{\delta \Delta t} \left[\int W_\delta(\theta - \theta') s(\theta') d\theta' - s_0(\theta) + b \right]$$

$$\frac{\delta \Delta t}{\epsilon} \frac{\partial s(\theta, t)}{\partial t} + s(\theta) = \int W_\delta(\theta - \theta') s(\theta') d\theta' + b.$$

1058 Comparing the above equation with Eq. 1, we find that the neural time constant $\tau = \delta \Delta t / \epsilon$.
 1059 Since the speed of the activity bump is $\epsilon / \Delta t$, we obtain a speed of

$$v = \delta / \tau. \quad (17)$$

1060 Thus, a network built with a kernel with offset δ in particular direction leads to activity
 1061 flow along that direction. Coupling multiple copies of such networks with opposing directions
 1062 of kernel offsets leads to an equilibrium, with the bump state at a fixed position. This can
 1063 be intuitively seen by noting that $W_\delta s + W_{-\delta} s \approx (W + \delta \partial_\theta W)s + (W - \delta \partial_\theta W)s = 2Ws$, and
 1064 thus opposing offset kernels acting on the same state are equivalent to the state being acted
 1065 on by a kernel with no offset.

To control the flow the bump in arbitrary directions, we will next demonstrate that the magnitude of the feed-forward input b in a particular subnetwork can bias the motion of the bump. To see this, we first consider Eq. 1 scaled by a factor α ,

$$\tau \frac{d\alpha s(\theta)}{dt} + \alpha s(\theta) = f \left[\int_{-\infty}^{\infty} W(\theta, \theta') \alpha s(\theta') d\theta' + \alpha b \right].$$

1066 Thus, scaling b by a factor α (i.e., $b \rightarrow \alpha b$) results in an equivalent solution of the dynamical
 1067 equation with the states s also scaled by the same factor α (i.e., $s(\theta) \rightarrow \alpha s(\theta)$).

Consider two such coupled networks with opposing offsets, with feedforward inputs scaled by $\alpha_1 = (1 + \alpha)/2$ and $\alpha_2 = (1 - \alpha)/2$. As noted above the neural firing rates can be assumed to be scaled by the same factors. Heuristically, we will assume that the firing rates of the coupled network can be approximated through individually scaled firing rates of independent offset networks. This leads to the effective interaction through the offset kernels as

$$\begin{aligned} W_\delta \alpha_1 s + W_{-\delta} \alpha_2 s &\approx \alpha_1 (W + \delta \partial_\theta W)s + \alpha_2 (W - \delta \partial_\theta W)s \\ &= [W + \delta(\alpha_1 - \alpha_2) \partial_\theta W]s \\ &\approx W_{\delta(\alpha_1 - \alpha_2)} s = W_{\delta\alpha} s. \end{aligned}$$

1068 Thus, the effective interaction is similar to that obtained by a kernel with an offset of $\delta\alpha$,
 1069 leading to a bump speed of $\delta\alpha / \tau$.

Finally, we note that while the above argument has been constructed for offsets along a single dimension, it readily generalizes to higher dimensions: For continuous and differentiable W , a directional derivative can be written as a linear combination of partial derivatives along coordinate axes, i.e.,

$$\alpha \frac{\partial W}{\partial \hat{e}_i} + \beta \frac{\partial W}{\partial \hat{e}_j} = \frac{\partial W}{\partial (\alpha \hat{e}_i + \beta \hat{e}_j)}.$$

1070 Thus, subnetworks with differently scaled feedforward inputs lead to differently scaled firing
 1071 rates s which leads to an interaction kernel that has an effective offset in the vector direction
 1072 determined by the scaling coefficients. This effective offset in a particular direction causes the
 1073 activity bump to flow along the manifold along that direction, leading to controllable flow of
 1074 the activity bump through differential feed-forward inputs to the coupled network.