Binding in hippocampal-entorhinal circuits enables compositionality in cognitive maps

Christopher J. Kymn[1*], **Sonia Mazelet**[1,2], **Anthony Thomas**[1,3], **Denis Kleyko**[4,5], **E. Paxon Frady**[6], **Friedrich T. Sommer**[1,6], **and Bruno A. Olshausen**[1,7]

[1]Redwood Center for Theoretical Neuroscience, Helen Wills Neuroscience Institute, UC Berkeley, Berkeley, USA
[2]Université Paris-Saclay, ENS Paris-Saclay, Gif-sur-Yvette, France
[3]Department of Electrical and Computer Engineering, UC Davis, Davis, USA
[4]Centre for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden
[5]Intelligent Systems Lab, Research Institutes of Sweden, Kista, Sweden
[6]Intel Labs, Santa Clara, USA
[7]Herbert Wertheim School of Optometry & Vision Science, UC Berkeley, Berkeley, USA
[*]cjkymn@berkeley.edu

Abstract

We propose a normative model for spatial representation in the hippocampal formation that combines optimality principles, such as maximizing coding range and spatial information per neuron, with an algebraic framework for computing in distributed representation. Spatial position is encoded in a residue number system, with individual residues represented by high-dimensional, complex-valued vectors. These are composed into a single vector representing position by a similarity-preserving, conjunctive vector-binding operation. Self-consistency between the representations of the overall position and of the individual residues is enforced by a modular attractor network whose modules correspond to the grid cell modules in entorhinal cortex. The vector binding operation can also associate different contexts to spatial representations, yielding a model for entorhinal cortex and hippocampus. We show that the model achieves normative desiderata including superlinear scaling of patterns with dimension, robust error correction, and hexagonal, carry-free encoding of spatial position. These properties in turn enable robust path integration and association with sensory inputs. More generally, the model formalizes how compositional computations could occur in the hippocampal formation and leads to testable experimental predictions.[1]

1 Introduction

The hippocampal formation (HF), consisting of hippocampus (HC) and the medial and lateral part of the neighboring entorhinal cortex, (MEC) and (LEC), is critical for forming memories and representing variables such as spatial position [1, 2]. Recent work has provided evidence of compositional structure in HF representations, enabling complex representations to be *composed* by simpler building blocks and their formation rules. Examples include novel recombinations of past experience occurring in replay [3], or the exponential expressivity of the grid cell code [4, 5]. In particular, compositional representations afford high expressivity with lower dimensional storage requirements [6], less complexity in latent state inference, and generalization to novel scenes with familiar parts.

To gain insight into the possible computational principles and neural mechanisms at play in the HF, we take a normative modeling approach. That is, we seek to construct a model built from a set of

[1]Code is available at `https://github.com/SoniaMaz8/Hippocampal_enthorinal_circuit`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

neural coding principles that effectively achieves the postulated function of the system. With this approach, we can then explain details about the neuroanatomical and neurophysiological structures in light of their particular contributions to an information processing objective. The resulting model can also lead to new predictions about the neural mechanisms that enable this function.

The postulated function of the HF —as a cognitive map and episodic memory— has a core computational requirement, to represent and navigate space. Here, space is either the actual physical environment or a more abstract conceptual space. We formulate multiple desiderata for an effective representation of space. We then show that a residue number system, incorporated into a compositional encoding scheme, fulfills these desiderata. It is achieved by a modular attractor network that factorizes encoded locations into components of a residue number system. This provides an algorithmic-level hypothesis of hippocampal-entorhinal interactions. A core mechanism of this algorithm is *binding*, which draws inspiration from work in neuroscience, cognitive science, and artificial intelligence.

2 A normative model for the hippocampal formation

2.1 Principles for representing space

Our first principle is that space is represented by a compositional code that has high spatial-resolution, is noise-robust, and in which algebraic operations on the components can be updated in parallel. Prior work [4, 5] has proposed the residue number system (RNS) [7] as a candidate for fulfilling these requirements. An RNS expresses an integer x in terms of its remainder relative to a set of co-prime moduli $\{m_i\}$. For example, relative to moduli $\{3, 5, 7\}$, $x = 40$ is encoded as $\{1, 0, 5\}$. The Chinese Remainder Theorem guarantees that all integers in the range $[0, M-1]$, where $M = \prod_i m_i$, are assigned a unique representation. An RNS provides high spatial resolution, carry-free arithmetic operations, and robust error correction [8]. Experimental observations in entorhinal cortex show a discrete multi-scale organization of spatial grid cells [9] that is compatible with the assumption of discrete RNS modules.

The second principle we adopt is that an individual residue value should be encoded by a neural population in a similarity-preserving fashion. In particular, we require that distinct integer values are represented with nearly orthogonal vectors. To achieve this principle, we use a method similar to random Fourier features [10]. Each modulus, with value m_i, is assigned a seed phasor vector, $\mathbf{g}_i \in \mathbb{C}^D$, whose elements $(\mathbf{g}_i)_j$ are drawn uniformly from the m_i-th roots of unity (i.e., $(\mathbf{g}_i)_j = e^{\sqrt{-1}\,\omega_{ij}}$, with $\omega_{ij} = \frac{2\pi}{m_i}\, k_j$, and k_j chosen randomly from $\{0, ..., m_i - 1\}$). The representation of a particular residue value $a_i \in \{0, \ldots, m_i - 1\}$ is then given by rotating the phases of the seed vector according to [11]:

$$\mathbf{g}_i(a_i) = (\mathbf{g}_i)^{a_i}, \tag{1}$$

where we abuse notation slightly to also think of $\mathbf{g}_i$ as a function that takes a_i as input and produces an embedding as described above. The complex-valued vectors can be mapped to interpretable population vectors via a randomized Fourier transform (Figures 6D and S2).

Our third principle concerns the manner in which a unique representation of a particular point in space is formed from the individual residue representations. This requires that we somehow combine the residue vectors for each modulus. Combining via concatenation, though straightforward, is not effective because codes that coincide in subsets of their residue representation would be similar, even when the encoded values are very different. Thus, the method of combining residue codes must be *conjunctive*. Conjunctive composition is often called *binding* and is of fundamental importance in neuroscience [12], cognitive science [13], and machine learning [14]. An early proposal for binding is the tensor product of vector representations [15], with the tensor order equal to the number of bound objects.

Here, we implement binding with component-wise vector multiplication, a dimensionality preserving operation that represents a lossy compression of the full tensor product [16, 17]. The resulting compositional vector representation of an integer $x \in \mathbb{Z}$ using an RNS representation with K moduli, $\{a_1, a_2, .., a_K\}$, is:

$$\mathbf{p}(x) = \bigodot_{i=1}^{K} \mathbf{g}_i(a_i). \tag{2}$$

We prove in Appendix A.1 that this coding scheme represents distinct integer states using nearly orthogonal vectors, and we show that it generalizes in a natural way to support representation of arbitrary real numbers in a similarity preserving fashion.

Eq. 2 represents individual points along a line. In general, however, a spatial representation involves points in 2D or 3D spaces. Conveniently, vector binding can be also used to compose representations of multidimensional lattices from vectors representing individual dimensions. As we will explain, there is still a choice in this composition that determines the resulting lattice structure. Following earlier proposals [18–20], our fourth normative principle is to choose the lattice structure so that spatial information is maximized, as described in Section 3.5.

The final principle we require is that for computations such as path integration, there should be a simple vector manipulation that results in addition of the encoded variables. Again, vector binding provides this functionality with our coding strategy, because of the following property:

$$\mathbf{g}(x) \odot \mathbf{g}(y) = \mathbf{g}(x + y). \tag{3}$$

2.2 Modular attractor network for spatial representation

A standard model of grid cell circuits is the line attractor, in which states that represent a consistent location lie on a low-energy manifold [4]. When initialized from a noisy location pattern, the circuit dynamics will generate a denoised location representation. Rather than forming a line attractor model for the entire representational space (Eq. 2), we propose a modular network architecture, so that the compositional structure of a residue number representation can scale towards a large range with fewer memory resources (Section 3.2), in a manner robust to noise (Section 3.3).

A starting point for our attractor network model is the Hopfield network, which acts as an associative memory by storing memory patterns as fixed-point attractors. The Rademacher-Hopfield network [21] is a dynamical system whose state is a vector $\mathbf{x} \in \{-1, +1\}^D$ that obeys the following dynamics:

$$\mathbf{x}(t + 1) = \text{sign}(\mathbf{X}\mathbf{X}^T\mathbf{x}(t)) \tag{4}$$

with $\mathbf{X}$ as the matrix of memorized patterns (column vectors of $\mathbf{X}$). The fixed-point attractor dynamics can be generalized to complex memory patterns $\mathbf{z} \in \mathbb{C}^D$:

$$\mathbf{z}(t + 1) = \sigma(\mathbf{Z}\mathbf{Z}^\dagger\mathbf{z}(t)), \tag{5}$$

where σ is a non-linearity normalizing the amplitude of each complex-valued component to one [22], and $\mathbf{Z}$ the corresponding matrix of memorized patterns. The model can also be discretized, such that each component is often quantized to a r-state phasor [23]. The Rademacher-Hopfield model is the special case where $r = 2$ and the phasors happen to be real-valued.

An r-state phasor network of the form of Eq. 5 is well-suited to serve as an attractor network for each of the residue vectors in an RNS representation of position, with $r = m_i$ for modulus i, and the matrix $\mathbf{Z}$ (which we shall denote $\mathbf{G}_i$) storing the $\mathbf{g}_i(a_i)$ for $a_i \in \{0, .., m_i - 1\}$. However, we desire a method for representing the whole coding range $M := \prod_i^K m_i$ without storing all M patterns in one large associative memory. For this purpose we show that a *resonator network*, a recently proposed recurrent network for *unbinding* conjunctive codes [24–26], lets us represent this range by storing only $n := \sum_i^K m_i \ll M$ patterns. Given a vector encoding of position, $\mathbf{p}(x)$, as formulated in Eq. (2), a resonator network will factorize it into its constituent RNS components by iteratively updating each residue vector estimate, $\hat{\mathbf{g}}_i$. This update is similar to the attractor dynamics of Eq. (5) but made to be consistent with $\mathbf{p}(x)$ given all other residue estimates $\hat{\mathbf{g}}_{j\neq i}$:

$$\hat{\mathbf{g}}_i(t + 1) = \sigma\Big(\mathbf{G}_i\mathbf{G}_i^\dagger\big(\mathbf{p} \bigodot_{j\neq i}^{K} \hat{\mathbf{g}}_j^*(t)\big)\Big) \quad \forall\, i \tag{6}$$

Let us now assume that the input $\mathbf{p}(x_t)$ encodes a spatial position x_t using Eq. (2). Given a velocity input $\mathbf{q}_i(v_t)$, estimated from self-motion input, path integration is performed by first running attractor dynamics, *then* updating attractor states by velocity.

$$\hat{\mathbf{g}}_i(t + 1) = \mathbf{q}_i(v_t) \odot \sigma(\mathbf{G}_i\mathbf{G}_i^\dagger\mathbf{p}(x_t) \bigodot_{i\neq j}^{K} \hat{\mathbf{g}}_j^*(t)) \tag{7}$$

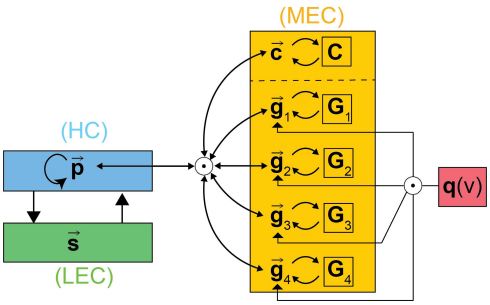

Figure 1: **Schematic of proposed attractor model.** In MEC, the $\mathbf{g}_i$ are residue representations in grid modules, and $\mathbf{c}$ encodes a context label. Input of velocity estimate $\mathbf{q}(\mathbf{v})$ can produce path integration in grid modules via binding, denoted by $\odot$. In HC, $\mathbf{p}$ represents contextualized place. Binding serves two roles in the MEC/HC interaction (symbolized by bidirectional arrows): *a)* factorizing $\mathbf{p}$ into $\mathbf{g}_i$'s, and *b)* generating an update of $\mathbf{p}$ from the $\mathbf{g}_i$'s, for example, after path integration. In LEC, $\mathbf{s}$ represents sensory input, interacting with $\mathbf{p}$ through a learned heteroassociative projection.

After velocity updates, one can update the input state $\mathbf{p}(x_t)$ with the conjunctive representation of the current factor estimates:

$$\mathbf{p}(x_{t+1}) = \bigodot_{i}^{K} \hat{\mathbf{g}}_i(t+1). \tag{8}$$

Further explanation and detail is provided in Appendix B.3.

2.3 Mapping the model to the HF

Although it is not obvious how the components of our normative model should map to the anatomical architecture of HF, we make one proposal as shown in Figure 1. The memory networks for residue representations $\hat{\mathbf{g}}_i$ correspond to grid modules in MEC. Similar to the grid modules, a module for context can be added to the architecture, such as a tag for the identity of a specific environment, with the recurrent synapses $\mathbf{C}$ storing tags of different environments.

The context neurons could correspond to the non-grid entorhinal cells, which can contain local, non-spatial information about the environment [27]. The vector $\mathbf{p}(x_t)$ can be linked to place cells in hippocampus. Internal HC circuitry can either buffer the input as in Eq. (6) or allow it to be updated dynamically according to the MEC input (Section 4.1). The mutual interactions between HC and MEC grid modules require projections between these structures. The binding operations that these interactions involve according to Eq. (6) are hypothesized to be implemented by nonlinear interactions between dendritic inputs in HC and MEC neurons.

The model also assumes the ability for sensory cues to provide the initialization signal of the cognitive map, represented by $\mathbf{s}$ in Figure 1. For completeness, we adopt the assumption of previous models (e.g., [28]) that heteroassociative memories are formed by the brain that link sensory cues to the hippocampal representations $\mathbf{p}$ (Section 4.2). This process would require the system to generate a new context vector $\mathbf{c}$ and initialize the cognitive map to a default location in order to learn about new environments. We show that through even a simple heteroassociative mechanism, our modular attractor network can robustly retrieve sensory memories and even protect its compositional structure.

3 Coding properties of the model

3.1 RNS representations have exponential coding range

The compositional RNS vector representation Eq. (2) can encode a coding range of M values using a total of n component patterns for representing the residue of individual modules. The scaling of the coding range is exponential in the number of moduli, K, since if each module has $\mathcal{O}(m)$ patterns, and the co-prime condition is satisfied, the scaling of the coding range is $\mathcal{O}(m^K)$. This recovers the expressivity argued by [4, 29].

More generally, it is also exponential in the number of component patterns, n. The optimal coding range is given by the best partition of n into a set of positive $\{m_i\}$. This optimization is identical to that of finding the maximum order of an element in the group of permutations S_n, because the maximum order can be found by finding the longest cycle. The scaling of this value in n is characterized by Landau's function $f(n)$, which is known to converge to $\exp(\sqrt{n \ln n})$ as $n \to \infty$ [30]. Figure 2A

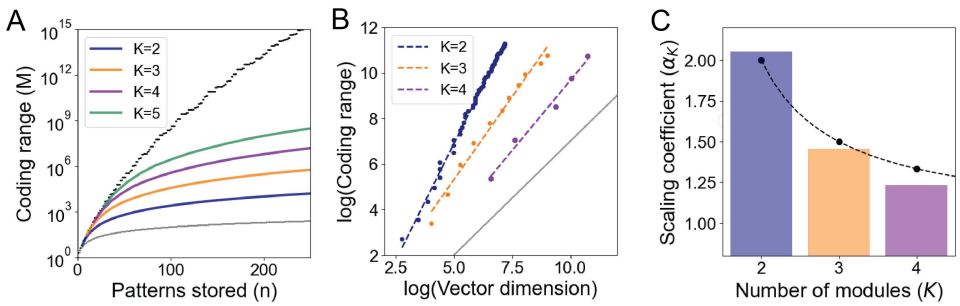

Figure 2: **Residue number systems, combined with a modular attractor network (resonator network), result in a new kind of attractor neural network with favorable scaling for a large combinatorial range. A)** Number of encoding states, M, grows rapidly in the number of modules, up to a maximum established by Landau's function (black dots). **B)** Coefficient of coding range, M, scales roughly as $\mathcal{O}(D^{\alpha_K})$, depending on the number of moduli, K, but with $\alpha_K > 1$. **C)** Estimation of scaling from slopes of linear regression (fit to log-log scale). Higher values of K require a higher dimension to achieve a particular coding range; empirical values are close to $\alpha_K = \frac{K}{K-1}$.

illustrates how Landau's function is the upper bound to what is achievable for any fixed number of moduli (K).

Though other kinds of representations can achieve an exponential coding range, the advantage of the compositional encoding of Eq. (2) comes from the fact that the binding operation implements carry-free vector addition (our fourth principle). This enables updates of the encoded value without requiring further transformations such as decoding, facilitating tasks such as path integration (Section 4.1, Appendix C.3). Binary representations, by contrast, have exponential coding range but require carry-over operations to implement.

3.2 The modular attractor network has superlinear coding range

The exponential scaling of the coding range of the RNS representation is a prerequisite to obtain a large coding range with the attractor network that has to perform computations on this representation, such as input denoising, working memory, and path integration. To estimate the scaling of the coding range in the proposed attractor network (Eq. 6), we study the critical dimension for which the grid modules converge with high probability. Specifically, we empirically estimate the minimum dimension required to retrieve an arbitrary RNS representation with high probability, given a maximum number of iterations (Figure 2B). Remarkably, we find that the number of component patterns n that can be stored is superlinear in the pattern dimension D; empirically $\mathcal{O}(D^{\alpha})$ for some $\alpha \geq 1$. For 2, 3, and 4 moduli, $\alpha \approx 2.05, 1.45$ and 1.23, respectively (Figure 2C).

These empirical scaling laws are consistent with a simple information-theoretic calculation (Appendix A.2). The minimal amount of bits to be stored for the entire RNS vector encoding scheme is of order $\mathcal{O}(M \log M)$, and the number of synapses in the attractor network is $\mathcal{O}(D \sqrt[K]{M})$. If one makes the cautious assumption of a capacity per synapse of $\mathcal{O}(1)$, the leading order for the coding range M is $\mathcal{O}(D^{\alpha})$, with $\alpha = \frac{K}{K-1}$.

While the coding range increases with the number of moduli (K) for the RNS representation, the superlinear scaling coefficient α_K decreases with K for the modular attractor network, reaching maximum superlinearity at the smallest value $K = 2$. This reversal is caused by the fact that increasing K decreases the number of synapses, i.e., the memory resource in the attractor network.

3.3 Robust error correction

In addition, we evaluate the robustness of our attractor model to noise. Because the RNS representations are composed of phasors, which are circular variables, we sample noise from a von Mises distribution with two parameters: mean ($\mu = 0$) and concentration pattern κ (Figure 3A). Higher κ values imply less noise; the distribution approximates a Gaussian with variance $1/\kappa$ for large κ. Further tests of model robustness to dropout, limited precision, and ablation are provided in Figure S6.

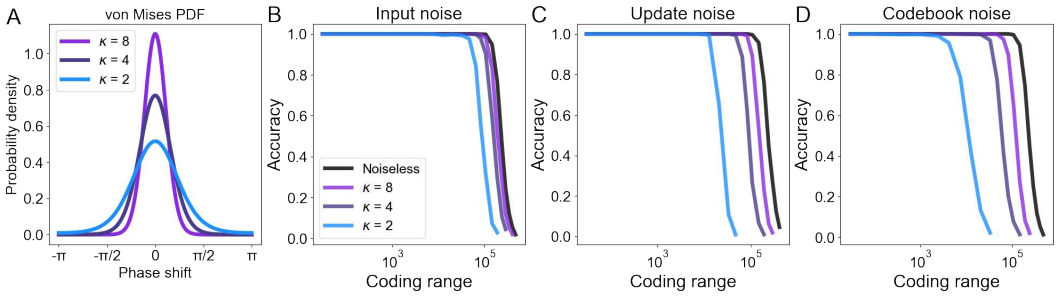

Figure 3: **Recovery of encoded positions is robust to multiple kinds of noise**. **A)** Visualization of the von Mises weight distribution. Note that the magnitude of the noise is inversely proportional to κ, and that the variance of the phase perturbation is much larger than the distance between the discrete states of phasors. **B-D)** Visualizations of accuracy as a function of coding range and κ for three separate cases: input noise (B), update noise (C), and codebook noise (D). Cases are shown in order of increasing difficulty. The resonator network maintains perfect accuracy up to a point, after which accuracy decays at an earlier point than the noiseless dynamics (black curve).

We consider three cases: noisy input patterns, noise added to each time step, and noisy weights corruptions of patterns in $\mathbf{G}_i$ (Appendix B.2). The empirical accuracy of recall varies depending on the type of corruption applied (Figure 3A). We find that for a given dimension D (in this case, 1024), increasing noise decreases the maximum coding range that can be decoded with high accuracy (Figure 3B-D). For a fixed noise level, the high-accuracy coding range is largest for input noise, followed by update noise and codebook noise. It is perhaps not surprising that codebook noise has the worst coding range, given that noise added to every stored pattern compounds across the dynamics. Fortunately, the demonstrated robustness to input noise enables sensory patterns to be denoised via heteroassociation (Section 4.2).

3.4 Interpolation between patterns enables continuous path integration

In general, there is a sharp difference between point and line attractors. In our attractor model, the RNS representations of integer values are stored as discrete fixed points. Nevertheless, the attractor network also converges to states that represent non-integer values that are not explicitly stored. In other words, the network smoothly interpolates to points on a manifold of states that represent integer and non-integer values encoded by (2). Figure 4A provides a visualization, showing that the kernel induced by inner product operations retains graded similarity for sub-integer shifts. This kernel enables the modular attractor network to settle to fixed points that correspond to interpolations between integers, and for sub-integer positions to be decoded.

The resolution of decoding is fundamentally limited by the signal to noise ratio. Even so, we find that, up to a fixed noise level, the accuracy regimes of integer decoding and sub-integer decoding coincide. This property enables sub-integer position shifts to be encoded within the states of the network, which, as we will show, results in stable, error-correcting path integration (Section 4.1). We quantify the gain in precision in terms of the bits of information that can on average be reconstructed from a vector (Figure 4D, Appendix B.2). Notably, even a moderate noise level of $\kappa = 8$ results in nearly the same information content as in the noiseless case.

3.5 Triangular frames in 2D maximize spatial information

In two-dimensional open field environments, grid cells have firing fields arranged in a hexagonal lattice [31]. Work in theoretical neuroscience shows the optimality of this lattice for 2D environments in terms of spatial information [18–20]. However, the presence of hexagonal firing fields raises a puzzle for residue number systems. Although a crucial property of a RNS is the carry-free property, most implementations of RNS will not perform carry-free updates within a module in non-Cartesian coordinate systems. This generally occurs because the updates of different coordinates must interact due to non-orthogonality.

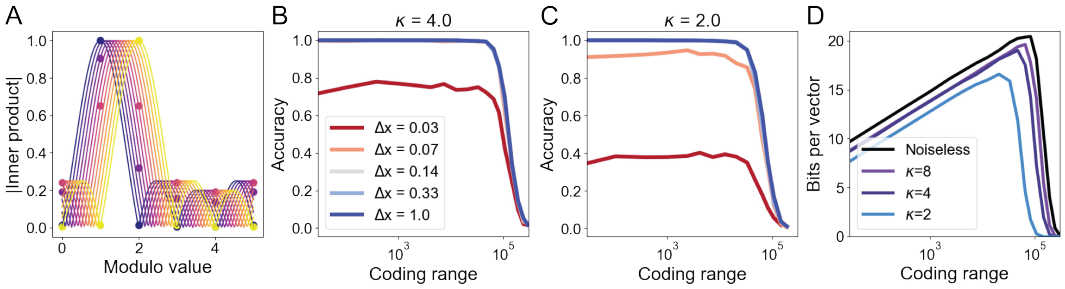

Figure 4: **Smooth interpolation between integer states enables encoding and decoding of sub-integer values. A)** Visualization of interpolation between two integer states. The position of the fractional value can be estimated by fitting a periodic sinc function (Appendix A.1) based on the inner products with integer codebooks (visualized in dots), then finding the location of the peak. **B, C)** Sub-integer states can be be decoded, up to a precision set by the noise level. Note that in both cases, sub-integer decoding can be just as accurate as integer decoding for the same range, even though the sub-integer decoding problem is strictly harder. Even $\kappa = 4$ is sufficient to achieve accuracy within a precision of $\Delta x = 0.07$, but for higher noise ($\kappa = 2$), the precision is worse. **D)** The best spatial precision (in bits) that can be decoded for a fixed noise level. Representations with less noise achieve both a higher coding range and higher information content per vector.

We resolve this issue by showing how to implement a version of vector binding of multiple coordinates in a triangular 'Mercedes-Benz' frame that enables carry-free hexagonal coding. Furthermore, we provide a combinatoric argument for the optimality of triangular *frames* for $\mathbb{R}^2$. (A frame is a spanning set for a vector space in which the basis vectors need not be linearly independent.) Our argument relies on the combinatorics of residue numbers, and so for the first time gives an explanation of why the coexistence of RNS and hexagonal codes is optimal.

To form a hexagonal tiling of 2D position requires two steps: first, projection into a 3-coordinate frame, and second, choosing phases such that simultaneous, equal movements along all three frames cancel out (Appendix A.3). The resulting Voronoi tessellation for different states is pictured in Figure 5A. This encoding enables higher spatial resolution in terms of the number of discrete states: $3m^2 - 3m + 1$ for triangular frames, versus m^2 for Cartesian frames. This increased expressivity results in a higher entropy) code for space (Figure 5B). It also results in both a periodic hexagonal kernel and the individual grid response fields being arranged in a hexagonal lattice (Figure 6C).

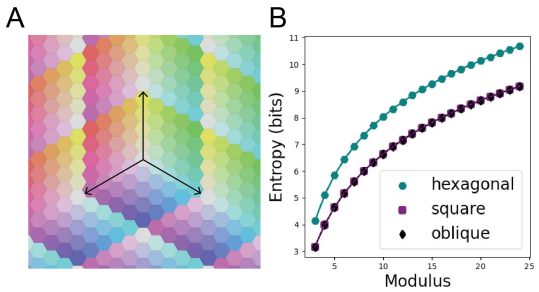

Figure 5: **Hexagonal coding improves spatial resolution. A)** Voronoi tessellation for $m = 5$. Each distinct color corresponds to a unique codeword in $\mathbb{C}^D$. Black arrows show the coordinate axes of the triangular 'Mercedes-Benz' frame in 2D. **B)** Hexagonal lattices have higher entropy than square lattices, allowing each state to carry higher resolution in its spatial output.

Prior models achieved hexagonal lattices either by pattern formation from circularly symmetric receptive fields (e.g., [32, 33]) arranged on a periodic rectangular sheet or by distorting a square lattice into an oblique one (e.g., [28, 34]). Importantly, oblique lattices have the combinatorial complexity as the square grid and, unlike the construction described above, they do not achieve the same level of spatial resolution (Figure 5B).

4 Testing functionalities of the model

4.1 Robust path integration

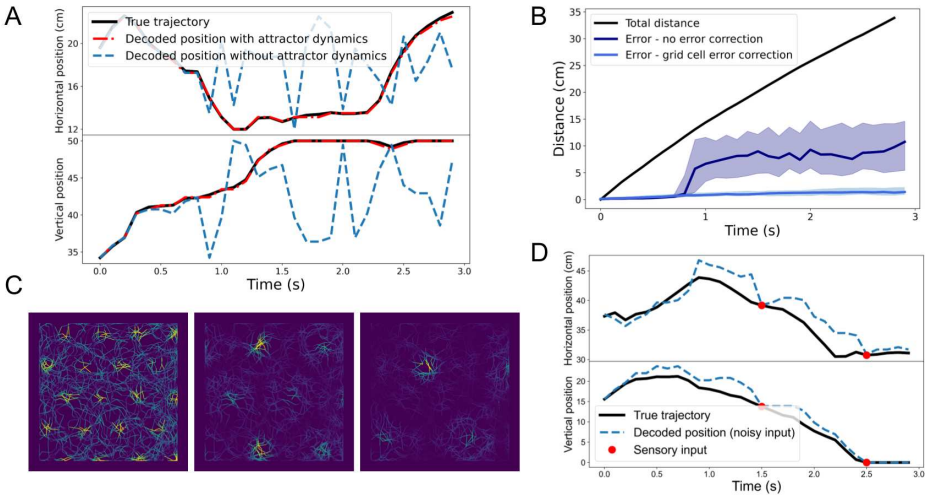

Figure 6: **Attractor dynamics facilitate robust path integration. A)** Example of path integration of a 2D trajectory in the case of intrinsic input noise on the place cell representation. The grid cell modules correct the noise that would otherwise induce drift after a short period of time. **B)** Path integration results averaged over multiple trajectories in the case of intrinsic input noise on the place cell representation. Grid cell modules limit noise accumulation along the trajectory. Solid lines report the median error over 100 trials, with shaded intervals reporting 25^{th} and 75^{th} percentile. **C)** Simulated trajectory, along which colors represent the similarity between the $\mathbf{g}_i$ of three different modules and vectors representing each position in the environment. We see hexagonal response fields, similar to those obtained from single unit recordings of MEC. **D)** Sensory patterns (symbolized by red dots), representing visual cues, are associated to positions in the environment. Presentation of visual cues helps correct drifted positions due to extrinsic noise.

Given the ability of the attractor model to update its representation of position from velocity inputs, along with its ability to represent continuous space, we evaluate its ability to perform path integration in the presence of noise. We simulate trajectories based on a statistical model for generating plausible rodent movements in an arena [35, 36], and we update grid cell and place cell state vectors according to Equations 7 and 8, respectively.

To evaluate the robustness of the model to error (Appendix B.3), we consider both sources of extrinsic noise (e.g., mis-representations of velocity information), and intrinsic noise (e.g., due to noise in weight updates). The robustness of our model to intrinsic noise is tested by comparing our results to the estimated trajectories obtained without the correction by the MEC modules (Figure 6A and B). We find that our model strongly limits noise accumulation along the trajectory and allows highly accurate integration for a longer period of time (Figure 6A). Consistent with our previous experiments on noise robustness (Figure 3), we find that the model has strong robustness to intrinsic noise, with extrinsic noise resulting in a drift of the estimated position.

We visualize the response fields in different modules and find hexagonal lattices with a module dependent scaling (Figure 6C, Appendix 4.1). In addition, we show that tethering to external cues (e.g., visual inputs), can significantly increase the accuracy of the attractor network. To study this, we associate visual cues to corresponding patches see Section 4.2) and observe that integration of information from sensory visual inputs succeeds in correcting drift due to extrinsic noise (Figure 6D).

4.2 Denoising sensory states via a heteroassociative memory

Finally, we describe a simple extension to our model, in which sensory patterns are fed from the lateral entorhinal cortex (LEC) to update the hippocampal state. This is consistent with theories of

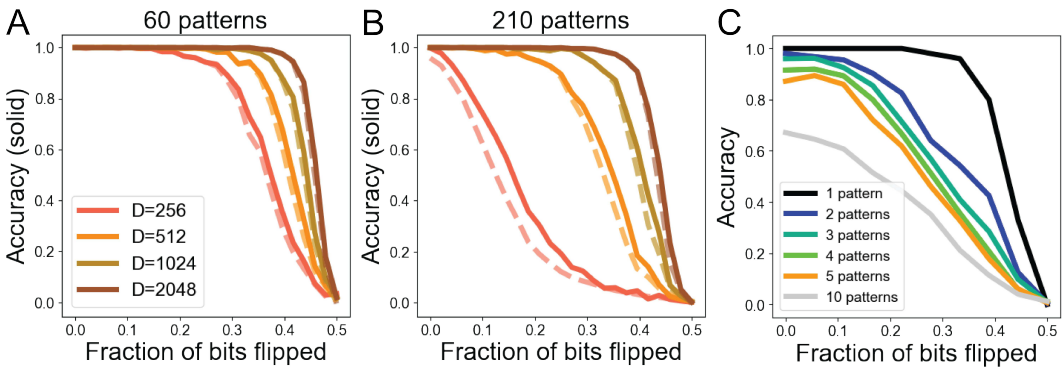

Figure 7: **Heteroassociation enables recovery of sensory patterns under corruption and superposition. A)** Accuracy for denoising 60 different random binary patterns for different vector dimension D. The dotted line is the average similarity between the decoded and ground truth patterns. **B)** Same experiment as in panel A, but with 210 different possible binary patterns. The accuracy is lower on average. **C)** Accuracy for denoising multiple patterns from a single input. This task is especially challenging, because sums of patterns combined in this way interfere with each other in retrieval (a phenomenon known as cross-talk noise). However, the compositional structure of our modular attractor network enables multiple patterns to be decoded with high probability.

memory suggesting that LEC provides the content of experiences to hippocampus [37], as well as with neuroanatomical evidence [38]. Although the structure of the representations of those sensory patterns is unknown, it is theorized that HF is critical to sensory pattern completion [39].

Consistent with this function, recent work [28, 40] has proposed that a heteroassociative scaffold connects sensory patterns to hippocampal activity, allowing robust denoising of sensory states. Though the main focus of our normative model is not sensory denoising, we show that a simple extension to our model (Appendix B.4) robustly retrieves noisy pattern even under high levels of corruption (Figures 7A and B). In Appendix C.3, we also discuss how this capacity for generalization can serve as a model for sequence retrieval and show some preliminary experiments.

In addition to robust denoising of single patterns, our model is also well-equipped to deal with compositions of sensory patterns. Two situations are worth emphasizing: first, we can often unmix multiple sensory states corresponding to a sum of patterns, because the compositional structure of binding between grid modules *protects* the items in summation (Figure 7C). This differentiates our model from other heteroassociative memories, in which sums of patterns would have multiple equally valid yet incompatible decodings. Second, the context vector module can separate the sensory information corresponding to different environments (Figure S3).

5 Discussion

There are by now numerous theories of the entorhinal cortex and hippocampus, including those that draw upon attractor dynamics and residue number systems. What this paper contributes to the existing body of work is a concrete set of design principles that can be brought together to build a functioning neural system capable of representing space and performing path integration, making the most use of limited neural resources and precision. In particular, a core design principle of this model is a compositional representation of space that achieves a superlinear coding range, which is achieved by a compact, multi-module attractor network. The compositional representation, in turn, is achieved via a vector binding operation, which enables binding multiple scales (moduli) and spatial dimensions, context, and spatial shifts for path integration. This binding mechanism builds on prior work in the field of hyperdimensional computing and vector symbolic architectures [11, 17, 26, 41–43] — and goes beyond it to develop a specific algorithmic hypothesis about structured operations in HF. Our analyses and experiments confirm that the model can achieve important functions of the hippocampal formation and they explain experimental observations, such as hexagonal grid cells, place cells, and remapping phenomena. The model thus contributes to, and greatly benefits from, existing work in theoretical neuroscience on residue number systems [4, 5], continuous attractor network models

of grid cells [4, 32, 44], models of compositionality in the hippocampal formation [45], and the optimality of hexagonal representations in 2D [18, 29].

That biology organized grid cells into multiple discrete modules, rather than pooling all resources into a single module attractor network, has posed a long-standing puzzle to theoreticians: What advantages are conferred by this organization? Our answer is that it provides exponential scaling in dynamic range by combining modules with limited dynamic range *multiplicatively*. Other recent work has focused on the problem of coordinating representations across multiple modules [28, 34, 46–48], and large scale recordings of the hippocampal formation [49] may provide new opportunities to evaluate their resulting predictions.

Our approach starts from principles of space encoding, in particular, the requirement of compositionality. This strategy is complementary to investigations of the emergence of place and grid cells in artificial neural networks (e.g., [36, 50–57]). These approaches can show the optimality of biological response features under the model assumptions, such as ANN properties, network architecture, training objective and protocol. Here, we emphasize the role of multiplicative binding, a computational primitive that is typically difficult to have emerge in an ANN setting. Early suggestions for realizing conjunctive binding already ventured outside the framework of ANNs [11, 15]. A simple extension of ANNs are sigma-pi neurons [58, 59] that can implement vector binding [60]. Recent work amplifies the view that full conjunctive binding would be a useful inductive bias to augment deep learning architectures [61], and various augmentations of ANNs with dedicated binding mechanisms have been proposed [14, 62, 63].

Our model has clear limitations. The attractor neural network for the cognitive map is still a high-level abstraction of spiking neural circuits in the hippocampal formation. In particular, the phasor states in the model are one linear transform removed from vectors that describe neural population activity. Thus, the mapping between model and neurobiological mechanisms requires an additional step. This disadvantage can be directly addressed by switching to other encoding schemes, such as sparse real- or complex-valued vectors, e.g., [64], for which conjunctive binding operations have been proposed [65, 66]. Although the model is more comprehensive than typical normative models, which usually focus on a single computation, it is far from covering the many other functional cell types observed in the hippocampal formation or contextual modulations observed during remapping. In addition, the current model includes learning only in the heteroassociative projection to LEC. Most observations regarding plasticity in HF are not captured, i.e., signals from reward, or eligibility traces. Finally, our assumptions about inputs to HF from the sensory pathway are simplifying and primarily intended as a proof of concept.

The purpose of the model, to express the fundamental principles of a compositional cognitive map, also leads to testable predictions: First, at the biophysical level, the model predicts multiplicative interactions between dendritic inputs providing the conjunctive binding operation. There are several biophysically realistic ways in which neurons can multiply their inputs [67, 68]. Contextual gating in dendritic branches of hippocampal neurons is consistent with our theory, hippocampal remapping, and neurophysiology of hippocampal dendrites [27]. Our attractor model predicts direct multiplicative interactions between MEC modules, which remains to be tested. Second, the model predicts relatively fixed attractor weights between place and grid cells, with a higher degree of plasticity for the weights between sensory encodings and hippocampal states. Third, we predict that causal perturbations of one grid module can affect the states of other grid modules without involvement of the hippocampus, in a direction that is self-consistent with the update of the attractor state.

Finally, we believe that the proposed modeling approach and the specific attractor model presented have broader applications in neuroscience. For example, the problem of factorization is critical to forming compositional representations of visual scenes, and a closely related attractor neural network can find efficient solutions to such problems [69]. In addition, there are promising ways to map complex-valued attractor neural networks to spiking neural networks [70, 71], which could connect the principles derived here to a concrete implementation on neuromorphic hardware. Such a neuromorphic implementation could yield further quantitative predictions for neuroscience and is an exciting direction for future work.

Acknowledgments

The work of CJK was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. The work of SM was carried out as part of the ARPE program of ENS Paris-Saclay and supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement HORIZON-INFRA-2022-SERV-B-01. The work of DK and BAO was supported in part by Intel's THWAI program. The work of CJK and BAO was supported by the Center for the Co-Design of Cognitive Systems (CoCoSys), one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, as well as NSF awards 2147640 and 2313149. DK has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 839179. FTS discloses support for the research of this work from NIH grant 1R01EB026955-0.

References

[1] H. Eichenbaum, "On the integration of space, time, and memory," *Neuron*, vol. 95, no. 5, pp. 1007–1018, 2017.

[2] E. I. Moser, M.-B. Moser, and B. L. McNaughton, "Spatial representation in the hippocampal formation: A history," *Nature Neuroscience*, vol. 20, no. 11, pp. 1448–1464, 2017.

[3] Z. Kurth-Nelson *et al.*, "Replay and compositional computation," *Neuron*, vol. 111, no. 4, pp. 454–469, 2023.

[4] I. R. Fiete, Y. Burak, and T. Brookings, "What grid cells convey about rat location," *Journal of Neuroscience*, vol. 28, no. 27, pp. 6858–6871, 2008.

[5] S. Sreenivasan and I. Fiete, "Grid cells generate an analog error-correcting code for singularly precise neural computation," *Nature Neuroscience*, vol. 14, no. 10, pp. 1330–1337, 2011.

[6] T. E. Behrens *et al.*, "What is a cognitive map? Organizing knowledge for flexible behavior," *Neuron*, vol. 100, no. 2, pp. 490–509, 2018.

[7] H. L. Garner, "The residue number system," in *Western Joint Computer Conference (WJCC)*, 1959, pp. 146–153.

[8] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," in *Annual ACM symposium on Theory of Computing (STOC)*, 1999, pp. 225–234.

[9] H. Stensola, T. Stensola, T. Solstad, K. Frøland, M.-B. Moser, and E. I. Moser, "The entorhinal grid map is discretized," *Nature*, vol. 492, no. 7427, pp. 72–78, 2012.

[10] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 20, 2007.

[11] T. A. Plate, "Holographic recurrent networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 5, 1992.

[12] C. von der Malsburg, "Am I thinking assemblies?" In *Brain Theory*, 1986, pp. 161–176.

[13] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[14] K. Greff, S. Van Steenkiste, and J. Schmidhuber, "On the binding problem in artificial neural networks," *arXiv:2012.05208*, 2020.

[15] P. Smolensky, "Tensor product variable binding and the representation of symbolic structures in connectionist systems," *Artificial Intelligence*, vol. 46, pp. 159–216, 1990.

[16] T. Plate *et al.*, "Holographic reduced representations: Convolution algebra for compositional distributed representations.," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1991, pp. 30–35.

[17] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, pp. 139–159, 2009.

[18] A. Mathis, M. B. Stemmler, and A. V. Herz, "Probable nature of higher-dimensional symmetries underlying mammalian grid-cell activity patterns," *Elife*, vol. 4, e05979, 2015.

[19] X.-X. Wei, J. Prentice, and V. Balasubramanian, "A principle of economy predicts the functional architecture of grid cells," *Elife*, vol. 4, e08362, 2015.

[20] F. Anselmi, M. M. Murray, and B. Franceschiello, "A computational model for grid maps in neural populations," *Journal of Computational Neuroscience*, vol. 48, pp. 149–159, 2020.

[21] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[22] A. Noest, "Phasor neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 1987, pp. 584–591.

[23] A. J. Noest, "Discrete-state phasor neural networks," *Physical Review A*, vol. 38, no. 4, p. 2196, 1988.

[24] E. P. Frady, S. J. Kent, B. A. Olshausen, and F. T. Sommer, "Resonator networks, 1: An efficient solution for factoring high-dimensional, distributed representations of data structures," *Neural Computation*, vol. 32, no. 12, pp. 2311–2331, 2020.

[25] S. J. Kent, E. P. Frady, F. T. Sommer, and B. A. Olshausen, "Resonator networks, 2: Factorization performance and capacity compared to optimization-based methods," *Neural Computation*, vol. 32, no. 12, pp. 2332–2388, 2020.

[26] C. J. Kymn *et al.*, "Computing with residue numbers in high-dimensional representation," *arXiv 2311.04872*, 2023.

[27] P. Latuske, O. Kornienko, L. Kohler, and K. Allen, "Hippocampal remapping and its entorhinal origin," *Frontiers in Behavioral Neuroscience*, vol. 11, p. 253, 2018.

[28] S. Chandra, S. Sharma, R. Chaudhuri, and I. Fiete, "High-capacity flexible hippocampal associative and episodic memory enabled by prestructured "spatial" representations," *bioRxiv*, 2023.

[29] A. Mathis, A. V. Herz, and M. B. Stemmler, "Resolution of nested neuronal representations can be exponential in the number of neurons," *Physical Review Letters*, vol. 109, no. 1, p. 018 103, 2012.

[30] E. Landau, "Über die maximalordnung der permutationen gegebenen grades," *Archiv der Mathematik und Physik*, vol. 3, pp. 92–103, 1903.

[31] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–806, 2005.

[32] M. C. Fuhs and D. S. Touretzky, "A spin glass model of path integration in rat medial entorhinal cortex," *Journal of Neuroscience*, vol. 26, no. 16, pp. 4266–4276, 2006.

[33] Y. Burak and I. R. Fiete, "Accurate path integration in continuous attractor network models of grid cells," *PLoS Computational Biology*, vol. 5, no. 2, e1000291, 2009.

[34] N. Mosheiff and Y. Burak, "Velocity coupling of grid cell modules enables stable embedding of a low dimensional variable in a high dimensional neural attractor," *Elife*, vol. 8, e48494, 2019.

[35] F. Raudies and M. E. Hasselmo, "Modeling boundary vector cell firing given optic flow as a cue," *PLoS Computational Biology*, vol. 8, no. 6, e1002553, 2012.

[36] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[37] J. R. Manns and H. Eichenbaum, "Evolution of declarative memory," *Hippocampus*, vol. 16, no. 9, pp. 795–808, 2006.

[38] J. J. Knierim, J. P. Neunuebel, and S. S. Deshmukh, "Functional correlates of the lateral and medial entorhinal cortex: Objects, path integration and local–global reference frames," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1635, p. 20 130 369, 2014.

[39] T. J. Teyler and P. DiScenna, "The hippocampal memory indexing theory," *Behavioral Neuroscience*, vol. 100, no. 2, pp. 147–154, 1986.

[40] S. Sharma, S. Chandra, and I. Fiete, "Content addressable memory without catastrophic forgetting by heteroassociation with a fixed scaffold," in *International Conference on Machine Learning (ICML)*, 2022, pp. 19 658–19 682.

[41] R. W. Gayler, "Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience," in *Joint International Conference on Cognitive Science (ICCS/ASCS)*, 2003, pp. 133–138.

[42] D. Kleyko, D. Rachkovskij, E. Osipov, and A. Rahimi, "A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–52, 2023.

[43] N. S.-Y. Dumont, J. Orchard, and C. Eliasmith, "A model of path integration that connects neural and symbolic representation," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 44, 2022.

[44] K. Zhang, "Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory," *Journal of Neuroscience*, vol. 16, no. 6, pp. 2112–2126, 1996.

[45] D. C. McNamee, K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman, "Flexible modulation of sequence generation in the entorhinal–hippocampal system," *Nature Neuroscience*, vol. 24, no. 6, pp. 851–862, 2021.

[46] N. Mosheiff, H. Agmon, A. Moriel, and Y. Burak, "An efficient coding theory for a dynamic trajectory predicts non-uniform allocation of entorhinal grid cells to modules," *PLoS Computational Biology*, vol. 13, no. 6, e1005597, 2017.

[47] L. Kang and V. Balasubramanian, "A geometric attractor mechanism for self-organization of entorhinal grid modules," *Elife*, vol. 8, e46687, 2019.

[48] H. Agmon and Y. Burak, "A theory of joint attractor dynamics in the hippocampus and the entorhinal cortex accounts for artificial remapping and grid cell field-to-field variability," *Elife*, vol. 9, e56894, 2020.

[49] T. Waaga *et al.*, "Grid-cell modules remain coordinated when neural activity is dissociated from external sensory cues," *Neuron*, vol. 110, no. 11, pp. 1843–1856, 2022.

[50] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization," in *International Conference on Learning Representations (ICLR)*, 2018, pp. 1–19.

[51] J. C. Whittington, W. Dorrell, S. Ganguli, and T. Behrens, "Disentanglement with biological constraints: A theory of functional cell types," in *International Conference on Learning Representations (ICLR)*, 2022.

[52] W. Dorrell, P. E. Latham, T. E. Behrens, and J. C. Whittington, "Actionable neural representations: Grid cells from minimal constraints," in *International Conference on Learning Representations (ICLR)*, 2023.

[53] B. Sorscher, G. C. Mel, S. A. Ocko, L. M. Giocomo, and S. Ganguli, "A unified theory for the computational and mechanistic origins of grid cells," *Neuron*, vol. 111, no. 1, pp. 121–137, 2023.

[54] R. Schaeffer, M. Khona, T. Ma, C. Eyzaguirre, S. Koyejo, and I. Fiete, "Self-supervised learning of representations for space generates multi-modular grid cells," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.

[55] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman, "The hippocampus as a predictive map," *Nature Neuroscience*, vol. 20, no. 11, pp. 1643–1653, 2017.

[56] J. C. Whittington *et al.*, "The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation," *Cell*, vol. 183, no. 5, pp. 1249–1263, 2020.

[57] Y. Chen, H. Zhang, M. Cameron, and T. Sejnowski, "Predictive sequence learning in the hippocampal formation," *bioRxiv*, 2022.

[58] J. A. Feldman and D. H. Ballard, "Connectionist models and their properties," *Cognitive Science*, vol. 6, no. 3, pp. 205–254, 1982.

[59] B. W. Mel and C. Koch, "Sigma-Pi learning: On radial basis functions and cortical associative learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 1989, pp. 474–481.

[60] T. A. Plate, "Randomly connected Sigma-Pi neurons can form associator networks," *Network: Computation in Neural Systems*, vol. 11, no. 4, p. 321, 2000.

[61] A. Goyal and Y. Bengio, "Inductive biases for deep learning of higher-level cognition," *Proceedings of the Royal Society A*, vol. 478, no. 2266, 2022.

[62] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves, "Associative long short-term memory," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1986–1994.

[63] A. Ganesan *et al.*, "Learning with holographic reduced representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 25 606–25 620.

[64] M. Laiho, J. H. Poikonen, P. Kanerva, and E. Lehtonen, "High-dimensional computing with sparse vectors," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, 2015, pp. 1–4.

[65] D. A. Rachkovskij and E. M. Kussul, "Binding and normalization of binary sparse distributed representations by context-dependent thinning," *Neural Computation*, vol. 13, no. 2, pp. 411–452, 2001.

[66] E. P. Frady, D. Kleyko, and F. T. Sommer, "Variable binding for sparse distributed representations: Theory and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2191–2204, 2023.

[67] C. Koch, *Biophysics of computation: information processing in single neurons*. Oxford university press, 2004.

[68] L. N. Groschner, J. G. Malis, B. Zuidinga, and A. Borst, "A biophysical account of multiplication by a single neuron," *Nature*, vol. 603, no. 7899, pp. 119–123, 2022.

[69] C. J. Kymn, S. Mazelet, A. Ng, D. Kleyko, and B. A. Olshausen, "Compositional factorization of visual scenes with convolutional sparse coding and resonator networks," in *2024 Neuro Inspired Computational Elements Conference (NICE)*, IEEE, 2024, pp. 1–9.

[70] E. P. Frady and F. T. Sommer, "Robust computation with rhythmic spike patterns," *Proceedings of the National Academy of Sciences*, vol. 116, no. 36, pp. 18 050–18 059, 2019.

[71] J. Orchard, P. M. Furlong, and K. Simone, "Efficient hyperdimensional computing with spiking phasors," *Neural Computation*, vol. 36, no. 9, pp. 1886–1911, 2024.

[72] T. A. Plate, *Holographic Reduced Representation: Distributed representation for cognitive structures*. CSLI Publications Stanford, 2003, vol. 150.

[73] A. Thomas, S. Dasgupta, and T. Rosing, "A theoretical perspective on hyperdimensional computing," *Journal of Artificial Intelligence Research*, vol. 72, pp. 215–249, 2021.

[74] E. P. Frady, D. Kleyko, C. J. Kymn, B. A. Olshausen, and F. T. Sommer, "Computing on functions using randomized vector representations (in brief)," in *Annual Neuro-Inspired Computational Elements Conference (NICE)*, 2022, pp. 115–122.

[75] K. L. Clarkson, S. Ubaru, and E. Yang, "Capacity analysis of vector symbolic architectures," *arXiv preprint arXiv:2301.10352*, 2023.

[76] J. O. Smith, *Spectral Audio Signal Processing*. 2011.

[77] B. Komer and C. Eliasmith, "Efficient navigation using a scalable, biologically inspired spatial representation," in *Annual Meeting of the Cognitive Science Society (CogSci)*, 2020, pp. 1532–1538.

[78] B. Komer, "Biologically inspired spatial representation," Ph.D. dissertation, University of Waterloo, 2020.

[79] E. P. Frady, D. Kleyko, and F. T. Sommer, "A theory of sequence indexing and working memory in recurrent neural networks," *Neural Computation*, vol. 30, no. 6, pp. 1449–1513, 2018.

[80] D. Kleyko *et al.*, "Efficient decoding of compositional structure in holistic representations," *Neural Computation*, vol. 35, no. 7, pp. 1159–1186, 2023.

[81] T. J. Wills, C. Lever, F. Cacucci, N. Burgess, and J. O'Keefe, "Attractor dynamics in the hippocampal representation of the local environment," *Science*, vol. 308, no. 5723, pp. 873–876, 2005.

[82] L. Thompson and P. Best, "Place cells and silent cells in the hippocampus of freely-behaving rats," *Journal of Neuroscience*, vol. 9, no. 7, pp. 2382–2390, 1989.

[83] A. O. Constantinescu, J. X. O'Reilly, and T. E. Behrens, "Organizing conceptual knowledge in humans with a gridlike code," *Science*, vol. 352, no. 6292, pp. 1464–1468, 2016.

[84] J. L. Bellmund, P. Gärdenfors, E. I. Moser, and C. F. Doeller, "Navigating cognition: Spatial codes for human thinking," *Science*, vol. 362, no. 6415, 2018.

[85] M. I. Schlesiger *et al.*, "The medial entorhinal cortex is necessary for temporal organization of hippocampal neuronal activity," *Nature Neuroscience*, vol. 18, no. 8, pp. 1123–1132, 2015.

[86] J. Yamamoto and S. Tonegawa, "Direct medial entorhinal cortex input to hippocampal CA1 is crucial for extended quiet awake replay," *Neuron*, vol. 96, no. 1, pp. 217–227, 2017.

[87] X. Li and P. Li, "Quantization algorithms for random fourier features," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6369–6380.

Supplemental material

A Mathematical derivations

A.1 Similarity-preserving properties of embeddings

In the following section, we examine the similarity-preserving properties of our coding scheme. Recall from Section 2.1 that our crucial desiderata are that: (1) distinct residue values are represented using vectors which are nearly orthogonal, and that (2) the inner product between representations of sub-integer values are reflective of a reasonable notion of similarity between the encoded values. There is a robust literature on this topic both within the Vector Symbolic Architectures community [72–75] and the broader ML community [10], who often study these techniques under the name "random features." The methods pursued here fall under these traditions.

To briefly recapitulate the construction of Equation 1: fix some positive integer m, and let $P(k)$ denote the uniform distribution over $\{0, ..., m-1\}$. Define an embedding $g : \mathbb{R} \to \mathbb{C}^D$ using the following procedure: draw $k_1, ..., k_D$ independently from $P(k)$, and set:

$$g(a)_j = \exp\left(i\omega k_j\right)^a / \sqrt{D},\ j = 1, ..., D,$$

where $\omega = 2\pi/m$, and $i = \sqrt{-1}$. To simplify analysis, we here assume that m is odd, in which case the above is equivalent to shifting the support of $P(k)$ to $\{-(m-1)/2, ..., (m-1)/2\}$, and defining the embedding $g : \mathbb{R} \to \mathbb{C}^D$ component-wise via:

$$g(a)_j = \exp\left(i\omega k_j a\right) / \sqrt{D},\ j = 1, ..., D.$$

The case that m is even is slightly different, but can be handled using similar techniques and the discrepancy does not affect any of our modeling goals.

Our basic claim is that in expectation with respect to randomness in the draw of $k_1, ..., k_D$, inner-products between the embeddings of two numbers a, a' recover the periodic sinc-function [76] of their difference. That is:

$$\mathbb{E}[\mathbf{g}(a)^\top \mathbf{g}(a')^*] = \frac{\sin(\pi(a-a'))}{m\sin(\pi(a-a')/m)} := \text{psinc}(a-a'),$$

This accomplishes goal (1) because, for t an integer which is not an integer multiple of m, psinc(t) $= 0$. Therefore, distinct integers are represented using vectors which are, in expectation, orthogonal. It also accomplishes goal (2), because psinc(t) ≈ 1 for $0 < |t| \ll 1$. The following theorem demonstrates this property more formally, and provides an approximation guarantee for a *specific* instantiation of $k_1, ..., k_D$.

Theorem 1. *Fix any $D > 0$ and $\delta \in (0, 1)$. For any pair $a, a' \in \mathbb{R}$ such that $a - a'$ is not an integer multiple of m, with probability at least $1 - \delta$ over randomness in the draw of $k_1, ..., k_D$:*

$$\left| \mathbf{g}(a)^\top \mathbf{g}(a')^* - \frac{\sin(\pi(a-a))}{m\sin(\pi(a-a')/m)} \right| \leq \sqrt{\frac{2}{D} \ln \frac{2}{\delta}}.$$

Proof. Fix any pair $a, a' \in \mathbb{R}$, and denote for concision $t = a - a'$. Taking an expectation with respect to randomness in $k_1, ..., k_D$ and using a well-known calculation from the signal processing

literature [76]:

$$\begin{aligned}
\mathbb{E}_{k_1,\ldots,k_d}\left[\mathbf{g}(a)^\top \mathbf{g}(a')^*\right] &= D\mathbb{E}_{k_1}[g(a)_1 g(a')_1^*] \\
&= \frac{1}{m}\sum_{k_1=-\frac{m-1}{2}}^{\frac{m-1}{2}} \exp\left(i\omega k_1(a-a')\right) \\
&= \frac{1}{m}\left(\frac{\exp\left(-\frac{i\omega t(m-1)}{2}\right) - \exp\left(\frac{i\omega t(m+1)}{2}\right)}{1-\exp(i\omega t)}\right) \\
&= \frac{\exp(i\omega t/2)}{m\exp(i\omega t/2)}\left(\frac{\exp(-\pi i t) - \exp(\pi i t)}{\exp(-\pi i t/m) - \exp(\pi i t/m)}\right) \\
&= \frac{\sin(-\pi t)}{m\sin(-\pi t/m)} \\
&= \frac{\sin(\pi(a-a'))}{m\sin(\pi(a-a')/m)},
\end{aligned}$$

The third equality follows from the second by noting that the latter is a sum of a geometric series with common ratio $r = \exp(\omega t)$. The fifth line follows from the fourth by recalling the identity $\sin(x) = (e^{ix} - e^{-ix})/2i$. In the limit of $t \to 0$, the expression evaluates to 1, consistent with the normalized inner product of a vector with itself.

To show concentration around this value, consider:

$$\mathbf{g}(a)^\top \mathbf{g}(a')^* = \frac{1}{D}\sum_{j=1}^{D} \exp(i\omega k_j(a-a')),$$

and note that since the complex part of the sum vanishes in expectation, we may consider, without loss of generality, the average of the real-valued quantities: $(\cos(\omega k_j(a-a')))_{j=1}^D$, which are bounded in the range ± 1. Therefore, by Hoeffding's inequality:

$$\Pr\left(\left|\mathbf{g}(a)^\top \mathbf{g}(a')^* - \mathbb{E}[\mathbf{g}(a)^\top \mathbf{g}(a')^*]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{D\epsilon^2}{2}\right),$$

whereupon we conclude that, with probability at least $1-\delta$ over randomness in the draw of $k_1, ..., k_D$:

$$\epsilon \leq \sqrt{\frac{2}{D}\ln\frac{2}{\delta}},$$

as claimed. □

This result can be readily extended to the binding of multiple residue number values. Let $\mathbf{g}(a) = \bigodot_{i=1}^K \mathbf{g}_i(a)$, where each $\mathbf{g}_i(a)$ is instantiated independently. Then, by independence, we observe that:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{g}(a)^\top \mathbf{g}(a')^*\right] &= \mathbb{E}\left[\prod_{i=1}^K \mathbf{g}_i(a)^\top \mathbf{g}_i(a')^*\right] \\
&= \prod_{i=1}^K \mathbb{E}\left[\mathbf{g}_i(a)^\top \mathbf{g}_i(a')^*\right]
\end{aligned}$$

The implication is that $\mathbb{E}[\mathbf{g}(a)^\top \mathbf{g}(a')^*] = 1$ if and only if all residue values agree, and zero otherwise. To show concentration around this value, we can again use Hoeffding's inequality, which recovers the same bound on the sufficient dimension.

A.2 Information-theoretic estimate of required pattern dimension

In this section, we describe an information-theoretic estimate on the dimension D necessary to retrieve n patterns within K modules. The main result we aim to show is that $D = \mathcal{O}(n^{(K-1)/K})$;

equivalently, the scaling of n for a given D is $\mathcal{O}(D^{K/(K-1)})$. This scaling roughly predicts our empirical results of finding the dimension required to achieve high accuracy, suggesting that the attractor network described here performs close to the theoretical bound.

The minimal total amount of information a network needs to store for denoising an RNS representation with coding range M is $\mathcal{O}(M \log(M))$. This results from the requirement of content addressability, i.e., for serving as a unique pointer to one of n patterns, each pattern must at least carry information of the order of $\mathcal{O}(\log(M))$. For simplicity, we now assume that each module is of size $\mathcal{O}(M^{1/K})$. The total capacity of the network is bounded by the number of synapses, which is $\mathcal{O}(D * K * M^{1/K}) = \mathcal{O}(D * M^{1/K})$ (assuming K is constant), times the capacity per synapse. Under the conservative assumption that the capacity per synapse is $\mathcal{O}(1)$, the dimension is of order $\mathcal{O}(e^{\frac{K-1}{K}\log(M)+\log(\log(M))})$. Thus, the leading order of how D depends on n is $\mathcal{O}(M^{(K-1)/K})$. If the capacity per synapse is assumed to be larger, $O(\log(M))$ bits, only the non-leading term cancels and the resulting order of D is still the same.

A.3 Construction of triangular frames

In order to convert a $2D$ coordinate $\mathbf{x}$ into a $3D$ frame $\mathbf{y}$, we first multiply it by a matrix, Ψ whose rows are the elements of a $3D$ equiangular frame:

$$\mathbf{y} = \begin{bmatrix} -1/\sqrt{3} & -1/3 \\ 1/\sqrt{3} & -1/3 \\ 0 & 2/3 \end{bmatrix} \mathbf{x} \tag{S1}$$

(This particular frame is commonly referred to as a 'Mercedes Benz' frame due to its resemblance to the iconic symbol.) A consequence of working with an overcomplete frame is that there may exist multiple values of $\mathbf{y}$ that correspond to the same $\mathbf{x}$. For this frame, the null space of Ψ^+ is the subspace spanned by $[1,1,1]^\intercal$ – grounding the intuition that equal movement in all equiangular directions "cancels out." It therefore might seem that triangular frames require extra operations to determine if two coordinates are equal, but here we show how to avoid this consequence.

The core strategy is to choose seed vectors $\mathbf{g}_{i,1}, \mathbf{g}_{i,2}, \mathbf{g}_{i,3}$ for each modulus m_i that implement this self-cancellation. For a modulus m_i, we draw the phasors of seed vectors from the m-th roots of unity. However, we further require that, for each vector component, the three selected phases sum to $0 \pmod{2\pi}$. We then form a hexagonal coordinate vector by binding the three seed vectors:

$$\mathbf{g}_i = \mathbf{g}_{i,1} \odot \mathbf{g}_{i,2} \odot \mathbf{g}_{i,3} \tag{S2}$$

By enforcing that the phases sum to $0 \pmod{2\pi}$, we ensure that positions that have an equivalent $\mathbf{x}$ coordinate are mapped to the same $\mathbf{g}_i$. Observe that Hadamard product binding of phasors is equivalent to summing their phases, and that binding e^{0i} corresponds to adding nothing. Hence, a pair of three-dimensional coordinates whose differences are a multiple of $[1,1,1]$ will be mapped to equivalent vector representations. Finally, we then form the residue number representation for different moduli by binding, as in Eq. 2. The presence of multiple modules and self-cancellation properties complement prior work on the efficiency of hexagonal kernels for spatial navigation tasks [77, 78].

The equivalence of certain 3D coordinates also helps us count the number of states. Clearly, the redundancy means that we have less than m^3 states, but it also shows us that every position in the hexagonal grid can be represented by a 3D coordinate which contains at least one coordinate equivalent to 0. There is one state where all coordinates are 0, $3(m-1)$ states where exactly two coordinates are 0, and $3(m-1)^2$ states where exactly one coordinate is zero. Thus, there are $3m^2 - 3m + 1$ states for the hexagonal lattice, compared to the m^2 states for the square lattice.

In the case of square lattices in $2D$, all states occupy an equal proportion of space; however, this is not the case for the hexagonal lattice (see Figure 5A). This is because states with more zero-valued coordinates occur slightly more frequently. To estimate the effect of unequal proportions on the entropy, we directly calculate the Shannon entropy of hexagonal lattices for finite size spatial grids of increasing radius l, as an approximation to the infinite lattice. We find that even for $l = 1000, m > 7$ the hexagonal code has 99 percent of the entropy of a system that divided all possibilities equally, and that this gap decreases as m grows larger. Thus asymptotically, as $m \to \infty$, the ratio of entropy for hexagonal vs. square grids tends towards $\log_2(3)$.

B Experimental details

All experiments were implemented in Python involving standard packages for scientific computing (including NumPy, SciPy, Matplotlib). We describe here the parameters and training setup of our experiments in further detail.

B.1 Scaling in dimension

For each number of moduli, K, we seek to find the smallest dimension D for which our attractor model factorizes its input, $\mathbf{p}$, into the correct grid states in a fixed time (50 iterations) with high probability (at least 99 percent empirically). In instances where the network states remain similar over time (at least 0.95 cosine similarity), we consider that it converged to a fixed point. If such convergence did not occur, we evaluate the accuracy at the last time step.

To evaluate scaling, we first choose our base moduli to be a set of K consecutive primes. We randomly select one of M random numbers to serve as the input and set the grid states to be random. We then evaluate a candidate dimension on the factorization task for a set number of trials (200) and check accuracy. We compare accuracy by considering whether the amplitude of the complex-valued inner products are highest for the true factor. If the accuracy is above our threshold, we then evaluate performance of a slightly higher dimension (dimensions evaluated are spaced apart on a logarithmic scale). Once a sufficiently high dimension achieves the accuracy threshold, we assume that the scaling is non-decreasing and use the last successful dimension as the first try.

Finally, we fit linear regression to all data points on a log-log scale to estimate the scaling between dimension and problem size. We report the slopes to estimate the scaling coefficients.

B.2 Error correction

General experimental setup. We fix in advance the vector dimension, noise level (determined by $1/\kappa$), and number of moduli. Given these parameters, we estimate the empirical accuracy of factorization on an arbitrary input known to correspond to one of the patterns. We use the same method for checking convergence as above, though we increase the maximum number of iterations to 100. For all experiments in this section, we average over $1{,}000$ trials.

In the case of input noise, the vector $\mathbf{p}$ is multiplied by a noise vector. In the case of update noise, after every time step, each module of the attractor network is corrupted by a von Mises noise update. In the case of codebook noise, all codebooks are corrupted before the start of any iterations.

Decoding values between integers. In order to test the ability of the modular attractor network to decode at sub-integer resolution, we fix a spatial resolution Δx to decode from. In our experiments, we test $\Delta x = \{1/3, 1/7, 1/15, 1/31\}$, and we also report $\Delta x = 1$ (integer decoding) as a control. Then, using as input a random integer and random multiple of Δx, we let the modules of the attractor network settle until convergence (as in other experiments). To evaluate accuracy, we test if the resulting output of the attractor network, $\odot_i \hat{\mathbf{g}}_{i=1}^{K}(t)$, is closer to the ground truth RNS representation than to any other value. We test this with a "coarse-to-fine" approach: first checking if it is within an integer, and then checking all fractional values within one of that integer. We regard the output as correct if both the integer and fraction match, and incorrect otherwise.

Estimation of information content from a vector. To measure the total resolution of our coding scheme in bits, we factor in both the number of states distinguished ($\tau = \frac{M}{\Delta x}$ and the empirical accuracy (ρ). To quantitatively estimate this, we report the information decoded in bits according to the following equation [79, 80]:

$$I(\tau, \rho) = a \log_2(\tau\rho) + (1-\rho)\log_2\left(\frac{\tau}{\tau - 1}(1-\rho)\right). \tag{S3}$$

A consequence of this equation is that the information decoded is 0 when the empirical accuracy is at chance ($1/\tau$).

B.3 Path integration

General experimental setup. We generate paths using a statistical model simulating rodent two-dimensional trajectories in a 50 cm^2 closed square environment [35, 36], with $\Delta t = 100$ ms. The

path integration method starts from the ground truth first position (x_0, y_0) which is converted to hexagonal coordinates (a_0, b_0, c_0) (see Section A.3) and encoded as an RNS representation $\mathbf{p}(0)$ of dimension $D = 3{,}000$ following the method in Section 2.1, for moduli $\{3, 5, 7\}$. We then factorize $\mathbf{p}(0)$ into $\{\hat{\mathbf{g}}_\mathbf{i}(0)\}_{i=1}^K$ to produce the estimated representation $\hat{\mathbf{p}}(0) = \bigodot_{i=1}^K \hat{\mathbf{g}}_\mathbf{i}(0)$.

At each time step $t \geq 0$, we estimate the position (x_{t+1}, y_{t+1}). We give the modular attractor network as input the previous position vector estimate $\hat{\mathbf{p}}(t)$. It is factorized into the residue components $\{\hat{\mathbf{g}}_\mathbf{j}(t)\}_{j=1}^K$ that are then shifted according to the velocity (da_t, db_t, dc_t) between (a_t, b_t, c_t) and $(a_{t+1}, b_{t+1}, c_{t+1})$. Namely, for each residue module, we build a velocity vector $\mathbf{q}_j(t) = \mathbf{g}_{j,1}(da(t)) \odot \mathbf{g}_{j,2}(db(t)) \odot \mathbf{g}_{j,3}(dc(t))$ that is binded to each residue component $\hat{\mathbf{g}}_\mathbf{j}(t)$. The estimated position vector is then the binding of the shifted estimated residue components: $\hat{\mathbf{p}}(t+1) = \bigodot_{j=1}^K \hat{\mathbf{g}}_\mathbf{j}(t) \odot \mathbf{q}_j(t)$. The estimated position $(\hat{x}_{t+1}, \hat{y}_{t+1})$ is chosen to be the position (x, y) in a grid of 30×30 positions mapping the entire environment, corresponding to the highest similarity between $\mathbf{p}(x, y)$ and $\hat{\mathbf{p}}(t+1)$.

We show the robustness of the path integration dynamics to two different sources of noise. In the case of extrinsic noise (Figure 6D), the hexagonal velocity is corrupted by additive Gaussian noise of variance 0.1. In the case of intrinsic noise (Figures 6A and B), the position vector $\tilde{p}_t$ is corrupted by binding with a vector sampled from a von Mises distribution with concentration parameter $\kappa = 2$.

Response field visualization. Given a moduli m_i and a vector $\mathbf{g}_i$, we visualize its response field by computing the similarity of the modular attractor output $\hat{\mathbf{g}}_i(t)$ and $\mathbf{g}_i$ along a trajectory. The periodicity in the distribution of random weights and the hexagonal coordinates produce periodic hexagonal receptive fields whose scale depends on m_i. Since the inner product between vector states induces a translation-invariant kernel, the response fields for a given moduli are translations of each other.

Connection to sensory cues. Sensory cues are random binary vectors of size $N_s = D$ that are associated with positions along the trajectory. When the true trajectory reaches a sensory cue, the hippocampal state $\hat{\mathbf{p}}_\mathbf{t}$ is updated using the heteroassociation method described in Appendix B.4

B.4 Heteroassociation

General experimental setup. We evaluate our model's performance for pattern denoising using a heteroassociative learning rule [28, 40]. We consider random binary patterns of size $N_s = D$. We corrupt the patterns by randomly flipping bits with probability $p_{\text{flip}} \in [0, 0.5]$ and associate them to place cell representations using heteroassociation with a pseudo-inverse learning rule. Let $\mathbf{S} \in \mathbb{R}^{N_s \times M}$ be the matrix of M patterns to hook to the scaffold and $\mathbf{H} \in \mathbb{C}^{M \times D}$ the matrix of M position vectors on which to hook the patterns. We associate a pattern $\mathbf{s}$ to a place cell representation $\mathbf{p} = \mathbf{H}\mathbf{S}^+\mathbf{s}$, where $\mathbf{S}^+$ is the pseudo-inverse of $\mathbf{S}$. The model returns a denoised place cell representation $\hat{\mathbf{p}}$ from which we can estimate a denoised pattern by inverting the heteroassociation projection $\hat{\mathbf{s}} = \text{sgn}\,(\mathbf{S}\mathbf{H}^+\hat{\mathbf{p}})$. Examples of corrupted inputs and reconstructed patterns are shown in Figure S3.

Scaling with dimension. We evaluate the impact that the dimension D has on the denoising performance in Figure 7, for a number of stored patterns $M = 60$ (in this case, $3 \times 4 \times 5$) and 210 (in this case, $5 \times 6 \times 7$). For each dimension $D \in \{256, 512, 1024, 2048\}$, we show the evolution of accuracy for different levels of corruption. For a given dimension D and noise level p_{flip}, we denoise a pattern and consider that the denoising is correct if the denoised pattern is closest to the ground truth pattern (in terms of cosine similarity). We repeat over 500 trials and report the accuracy as well as the average similarity (normalized inner product) between the denoised pattern and its noiseless version.

Superposition of patterns. We show that our model can denoise a superposition of n_p patterns one at a time, for $n_p \in \{1, 2, 3, 4, 5, 10\}$. We fix the dimension D to $2{,}000$ and for different values of bit flip probability $p_{\text{flip}} \in [0, ..., 0.5]$, we run the model on a superposition $\mathbf{s}$ of random binary patterns $\{\mathbf{s}_1, ..., \mathbf{s}_{n_p}\}$ of size $N_s = 2{,}000$: $\mathbf{s} = \mathbf{s}_1 + ... + \mathbf{s}_{n_p}$. We run the model n_p times and between each run the denoised pattern is explained away from the superposition [69]. Namely, for run $r \in \{1, ..., n_p - 1\}$ we denote $\hat{\mathbf{s}}(r)$ the denoised pattern. The input to run $r + 1$ is then $\mathbf{s}(r+1) = \mathbf{s}(r) - \hat{\mathbf{s}}(r)$. We find that the more patterns are superposed, the lower the overall denoising accuracy is. This is due to the fact that when a pattern is incorrectly denoised, explaining away

adds noise or spurious patterns to the representation of the superposition which makes the following denoising steps more difficult.

Comparison to structured patterns. We evaluate our model's ability to denoise structured patterns. We consider the FashionMNIST dataset, from which we select 105 images of size 28×28 that we binarize by setting pixel values to be -1 if below 127, and 1 elsewhere. We compare the denoising performance to the performance on random binary patterns of size $28 \times 28 = 784$ for fair comparison (Figure S4).

C Additional results

C.1 Further visualizations of grid cell modules

We further visualize the response fields for path integration by showing response fields from different units taken from the same grid module. We simulate a trajectory that traverses the entire environment and represent the activation of different position vectors along the trajectory. For each modulus $m_i \in \{3, 5, 7\}$, we show the similarity between 4 different vectors $\mathbf{g}_i$ from module m_i and the position vectors along the trajectory. We show in Figure S1 that the different receptive fields of a given module are translations of one another.

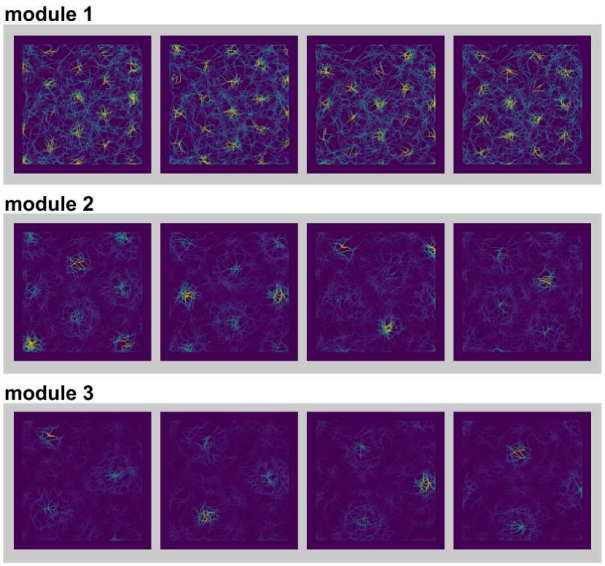

Figure S1: **Response field visualization of 4 different $\mathbf{g}_i$ in 3 different modules $m_i = 3, 5$ and 7.** For a fixed module, the response fields appear as translated versions of one another.

C.2 Remapping via modulation of context

We demonstrate that the context vector can serve as a model of *global remapping* in hippocampal place fields, which occurs when different environments are encoded with different populations of cells [27]. The simplest instance of this is when a place field occurs in context A but not context B, consistent with the observed sparsity of hippocampal activity [82]. To model this kind of remapping phenomenon, we consider an instance where there is a gradation of contexts with some phase transition between them; such an instance was studied experimentally [81]. Towards this end, we model linear combinations of these contexts, where the weights each context is given are $\text{sigmoid}(x), 1 - \text{sigmoid}(x)$, with x varying from -5 to 5 in 8 equally spaced increments, and with $\text{sigmoid}(x) = 1/(1 + \exp(-x))$. To model hippocampal units, we generate units that prefer one of the two contexts and have a random place field location, using its weight vector, or address, as $\mathbf{c} \bigodot_{i=1}^{K} \mathbf{g}_i$, and compare its output to that of the context/grid system at each location and context. It is worth noting that the original experiment of [81] also exhibited instances of rate remapping for

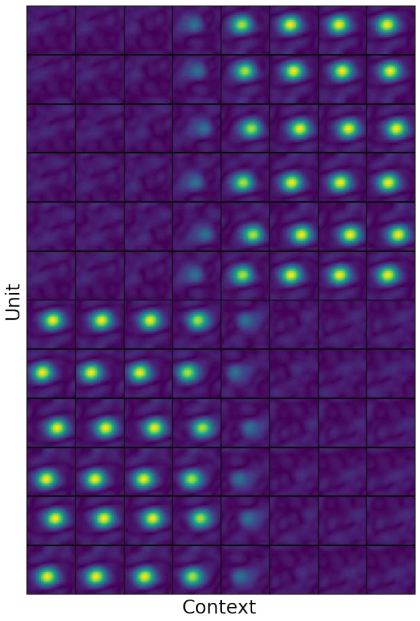

Figure S2: **Remapping of place cells depending on context.** The global remapping observed in the model response fields is similar to the findings of an experimental study of attractor network dynamics in hippocampus [81].

some units, and so there is certainly additional complexity underlying remapping that is not captured by our simple model.

C.3 Storing and retracing sequences

We demonstrate that our model can recover sequences by heteroassociation of patterns to positions and path integration in a conceptual space (Figure S5A). This is consistent with the postulated role of the hippocampal formation in performing navigation in conceptual spaces [83, 84], and the role of entorhinal cortex in generating sequences of neural firing in hippocampus [85, 86]. To evaluate our attractor model's fidelity at sequence memorization and retrieval, we simulate trajectories to form sequences of random binary patterns and recall the sequence using the path integration mechanism following the method in Section 4.1, for $D = 10{,}000$ and moduli $\{3, 5\}$. We add extrinsic noise to the velocity input, which accumulates along the trajectory and induces a drift. This implies that patterns at the end of sequences are less well recovered than ones at the beginning (Figures S5B and C).

C.4 Further tests of model robustness

We find that the proposed modular attractor network is also robust to other sources of noise. In particular, we evaluate robustness to synaptic noise, or dropout, decaying synaptic precision, and weight lesions (Figures S6A-C, respectively).

D Broader impacts

The results presented here are primarily addressing fundamental research questions, suggesting computational mechanisms in the brain. These results could lead to experimental design that improves our understanding of circuits in the hippocampal formation, or to artificial intelligence models capable of incorporating compositional structure in navigation tasks. Such impacts are typical of computational neuroscience research. On the other hand, we point out that the explicit compositionality of our modeling approach provides transparency into its operations, which would reduce the risk of unforeseen consequences.

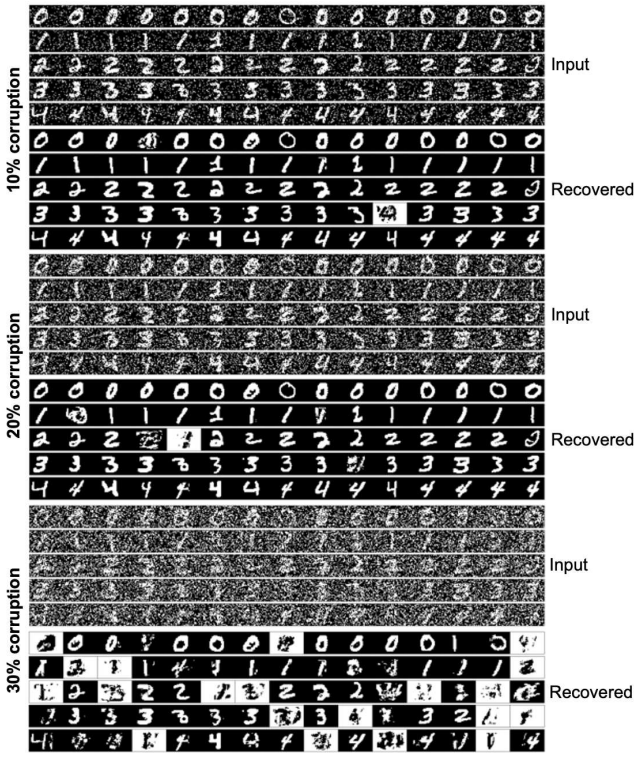

Figure S3: **Examples of sensory denoising with a heteroassociative memory on a binarized version of the MNIST Dataset.** Here, different contexts are used to index particular digit patterns. The degree of corruption (shown as "Input") influences the success of denoising (shown as "Recovered."

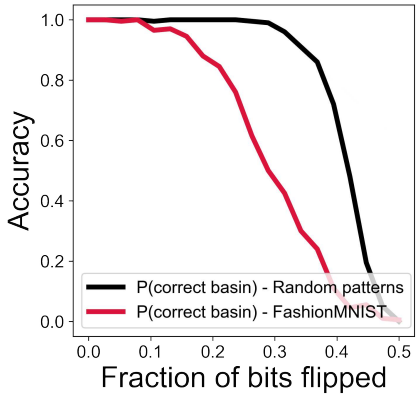

Figure S4: **A performance comparison for the heteroassociative memory on random patterns versus a binarized version of the FashionMNIST dataset.** For different levels of corruption, we denoise flattened binarized FashionMNIST images as well and random binary vectors of the same size. The overall denoising accuracy is lower for FashionMNIST, reflecting the difficulty in storing correlated patterns.

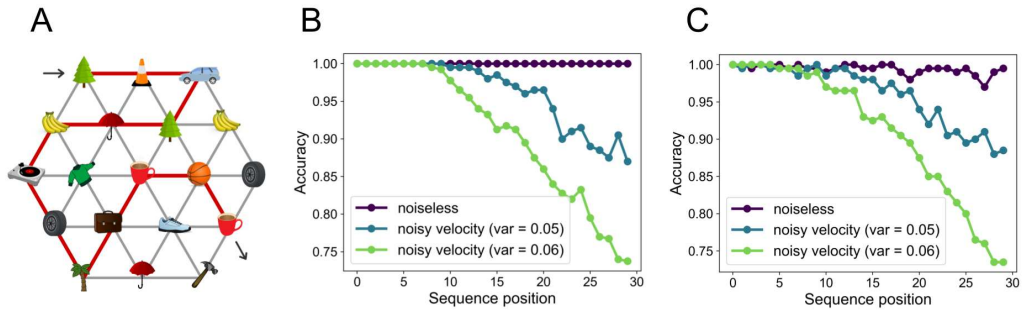

Figure S5: **Flexible sequence retrieval via path integration in a conceptual space. A)** An example of a hexagonal lattice with sensory observations associated with different states. Having knowledge of the underlying graph enables generalization to new trajectories in the space [6, 56]. **B)** Accuracy of random binary pattern retrieval as a function of position in the sequence for a fixed error rate and one context tag. The noiseless case achieves perfect accuracy, but errors accumulate after incorrect sequence predictions. **C)** Same as B), but with the additional task of inferring the context tag.

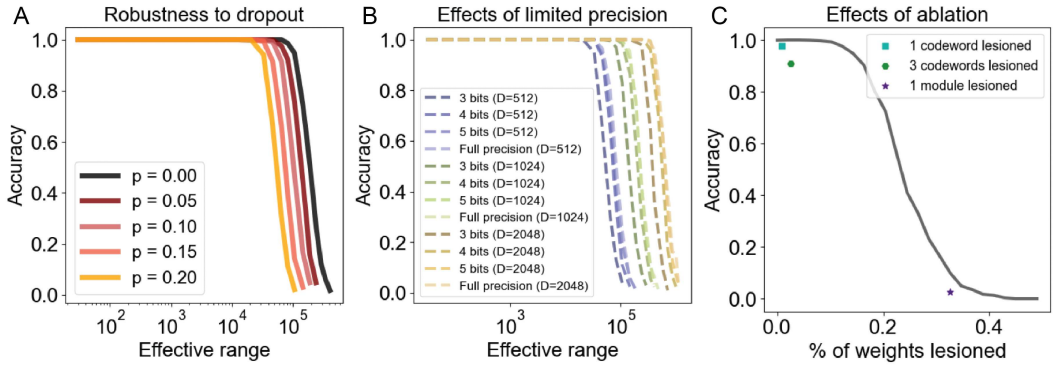

Figure S6: **Robustness of attractor network to additional sources of noise. A)** Robustness of the modular attractor network to synaptic failure (dropout). At each time step in the dynamics, each entry (weight value) in the matrices $\mathbf{G}$ and $\mathbf{G}^\dagger$ has an independent probability, p, of being set to 0. In spite of this synaptic noise, the model empirically converges to the correct solution up to a slightly smaller coding range. **B)** Robustness of the modular attractor network to limited precision. We use stochastic quantization, a technique studied in random feature models in machine learning [87], to round our model down to b bits of precision. We find empirically that 5 bits of precision (in this regime) performed nearly identically to the full precision vectors, indicating diminishing returns for higher precision. On the other hand, increasing vector dimension, which also requires more memory, results in increased capacity without facing the same diminishing returns. **C)** The effects of lesions (setting weights to 0) on network performance. The curves and data points reflect averages over 1000 trials. The network uses vectors of dimension $D = 1024$, and it has a dynamic range of $M = 65231$ ($= 37 \times 41 \times 43$). The gray curve shows the effect of lesioning random weights, each with independent probability p. Accuracy remains high up to a small percentage of lesioned weights (less than 10 percent). The teal square shows performance after lesioning one random column of one column of a random module's weights $\mathbf{G}_i$; the green hexagon shows effects of lesioning one random column for all three modules, and the purple star shows lesions to all codebooks in one module representing a non-zero residue. In each case, the performance is worse than random lesions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the abstract and introduction stay within the bounds of what we introduce in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed explicitly in the Discussion (Section 5), as well as implicitly throughout the rest of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The three technical sections, provided in Appendix A.1, Appendix A.2 and Appendix A.3, respectively, provide full proofs or calculations and cite any required background lemmas.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We explicitly wrote Appendix B to disclose any extra pieces of information required to reproduce experimental results. We have also provided implementations in code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Open access to the data and code is available at `https://github.com/SoniaMaz8/Hippocampal_enthorinal_circuit`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, these details are presented in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results are accompanied by error bars and confidence intervals when appropriate for our figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were performed on CPU with local resources. We do not have precise estimates of the amount of compute operations required, but each individual simulation took less than 3 days of total compute time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that it conforms to all standards outlined.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Appendix D.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The model does not involve any of the examples listed, and does not require additional safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are owned and created by the authors unless explicitly stated otherwise.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper involves neither crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper involves neither crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.