High-capacity flexible hippocampal associative and episodic memory enabled by prestructured "spatial" representations

Sarthak Chandra[1†], Sugandha Sharma[1†], Rishdev Chaudhuri[2], and Ila Fiete[1*]

[1]Department of Brain and Cognitive Sciences & McGovern Institute, MIT, Cambridge MA
[2]Center for Neuroscience, Department of Neurobiology, Physiology, and Behavior & Department of Mathematics, University of Californa Davis, CA
[†]these authors contributed equally to this work
[*]fiete@mit.edu

ABSTRACT

Hippocampal circuits in the brain enable two distinct cognitive functions: construction of spatial maps for navigation and storage of sequential episodic memories. This dual role of the hippocampus remains an enduring enigma. While there have been advances in modeling the spatial representation properties of the hippocampus, we lack good models of its role in episodic memory. Here we present a neocortical-entorhinal-hippocampal network model that exhibits high-capacity general associative memory, spatial memory, and episodic memory without the memory cliff of existing neural memory models. Instead, the circuit (which we call Vector-HaSH, Vector Hippocampal Scaffolded Heteroassociative Memory) exhibits a graceful tradeoff between number of stored items and detail, achieved by factorizing content storage from the dynamics of generating error-correcting stable states. The exponentially large space avoids catastrophic forgetting. Next, we show that pre-structured representations are an essential feature for constructing episodic memory: unlike existing episodic memory models, they enable high-capacity memorization of sequences by abstracting the chaining problem into one of learning transitions within a rigid low-dimensional grid cell scaffold. Finally, we show that previously learned spatial sequences in the form of location-landmark associations can themselves be re-usably leveraged as robust scaffolds and associated with neocortical inputs for a high-fidelity one-shot memory, providing the first circuit model of the "memory palaces" used in the striking feats of memory athletes.

Introduction

As we navigate through life, the hippocampus weaves threads of experience into a fabric of episodic memory. Cross-linked by various contexts, this fabric allows us to revisit scenes and events from only a few cues, like Proust's famous madeleine[1]. Such cue-driven recall makes memories available in ways relevant to make inferences in the present and to plan for the future. The hippocampal complex is responsible for this functionality[2–5], but it is unclear exactly how the architecture and representations of the hippocampal formation and the adjoining entorhinal cortex and other cortical regions enable it.

The representations and dynamics in substructures of the hippocampal complex have been studied extensively[6–26], and experimental findings combined with models and model testing have resulted in striking progress in our understanding of local circuit mechanisms[18,27–55]. These works put us in an excellent position to now build our understanding of the combined system, on how the substructures work together to subserve robust, efficient, and high-capacity associative memory storage and recall. A particularly intriguing question centers on the dual role of this structure: the hippocampus underlies both general episodic memory and spatial memory. Why are these two forms of memory co-localized? The storage of new autobiographical experiences, or episodic memory, is famously compromised with damage to the hippocampal complex[56–58]. Spatial memory refers to our ability to navigate and remember the layout of our physical environment. The hippocampus is populated with place cells that fire at a particular location in a particular environment and context[59,60]. In the entorhinal cortex, grid cells play a complementary role: they generate an invariant spatial representation across environments[36, 61–64], in the form of triangular grid-like firing patterns[61]. Thus, entorhinal grid cells are hypothesized to generate a spatial coordinate system, while hippocampal cells encode specific locations. Both types of memory (episodic and spatial) can be accumulated and accessed over a lifetime without major interference, despite the small size of the hippocampus relative to cortex.

The dual spatial and episodic memory functions of the hippocampus might be understood by three distinct (but non-exclusive) hypotheses. The first is that spatial information content is the most critical type of memory for survival – remembering details about the locations where we found food or dangers[65]. In this view, the circuit is optimized for spatial memory, while episodic memory is a small augmentation of that system. The second is that the circuit is focused on and optimized for episodic memory, but that spatial coordinates (not necessarily detailed spatial information content) are merely among the most stable and useful indices into that memory[66]. The third hypothesis is that the highly structured memory architecture's dynamics are equally optimized for episodic memories that may or may not contain spatial information, as well as for spatial memory. In essence, the third hypothesis is that the *abstract* low-dimensional representational architectures that might be interpreted as spatial are equally critical scaffolds for linking together (potentially entirely non-spatial) elements of an episodic memory[66–73].

In this work, we build a new neocortical-entorhinal-hippocampal circuit model for content-addressable associative memory and extensively characterize its properties numerically and theoretically. The two critical features of this model are: 1) A factorization of memory into a structured scaffold for fixed-point dynamics powered by a completely invariant grid cell circuit (in accord with our knowledge of that circuit[36, 61–64]) interacting in a fixed and random way with hippocampus, and a separate stage for (hetero)association with input data, and 2) forcing transition dynamics in episodic memory to be mediated by a shift operator acting on the low-dimensional latent grid states.

We find that this circuit excels at three kinds of memory: for individual inputs (item memory), for spatial mapping (spatial memory), and for sequences (episodic memory). One of the most interesting properties of the model is that the seemingly spatial representations of the grid cell circuit, specifically the low-dimensional and vectorial nature of the code, play a critical and distinct role even for completely non-spatial episodic memory. In other words, our model supports the third hypothesis about the co-localization of spatial and episodic memory.

Computationally, for each input the model creates a hash code given by the scaffold state, uses pre-structured recurrent connectivity in the scaffold to convert these hash states into fixed points for error-correction, and then uses a "strongly full-rank" property of the scaffold to enable (partial) decompression or reconstruction of the input patterns. The model also exploits the fact that grid coding states are ordered and lie in a low-dimensional space to enable efficient sequence memorization through vector transitions. For these reasons, we call our model Vector-HaSH: Vector Hippocampal Scaffolded Heteroassociative Memory. As we will see, critical to the success of Vector-HaSH in its properties of massive capacity, graceful tradeoff of content richness with number of memories without a memory cliff, non-interference between memories, lack of catastrophic forgetting, and resemblance to biological memory, is the factorization of the problems of creating dynamical fixed points (for pattern completion and stability) from content storage. Critical to its success in sequential episodic memory is the abstract low-dimensional vector updating property of the grid cell circuit, which in the spatial context is called velocity integration.

We will see that the highly constrained architecture, neural activations (invariant low-dimensional representation in grid cells), synaptic weights, and biologically plausible learning rules of Vector-HaSH enable memory without the full erasure (memory cliff) seen in existing neural memory models when adding inputs beyond a fixed low capacity, Fig. 1a. All code for running the model will be made freely available (upon publication) for others to make and test predictions for future experiments.

Results

HaSH architecture for hippocampal associative memory: Factorization of dynamics and content

Our model is based on known and inferred recurrent connectivity between entorhinal cortex and hippocampus[77–80] and among grid cells in the entorhinal cortex[33]. Processed extrahippocampal inputs enter the hippocampus (Fig. 1b, purple) via direct and non-grid entorhinal inputs (Fig. 1b, green); these inputs carry sensory information from the

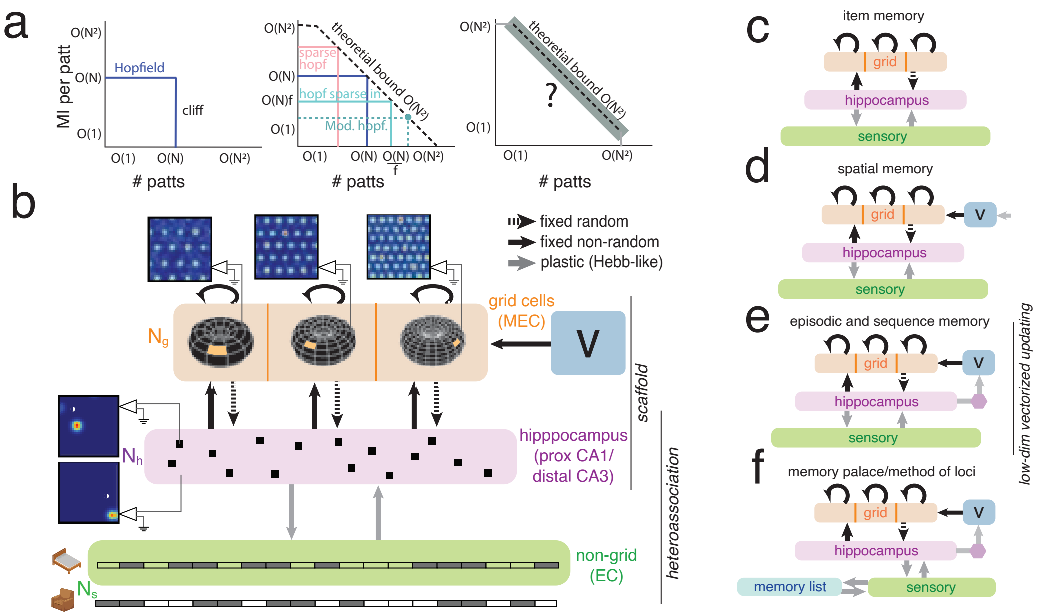

Figure 1. The challenge of biological memory and a biologically informed architecture for general episodic and spatial memory.: (a) *Left*: Hopfield networks exhibit a memory cliff where inputs are perfectly remembered before approximately N patterns but all memory of prior patterns is precipitously lost after approximately N input patterns. *Center*: Networks with N^2 plastic synapses have a theoretical bound of N^2 total memory bits[74,75]. All variations on Hopfield networks exhibit a memory cliff, albeit at different locations[74–76], approaching the bound at only a point. Here, $f = p\ln(p)$ where p is the sparseness of patterns in the Hopfield network with sparse inputs. *Right*: An ideal content-addressable memory would demonstrate continued information storage at or near the theoretical bound, regardless of the number of stored patterns. (b) Processed sensory inputs project from cortical and non-grid entorhinal neurons (green) into the hippocampus. The hippocampus (purple) also receives grid cell inputs via a fixed (non-plastic) random projection. Hippocampal projections back to grid cells are set once by associative learning (e.g. over development) and then held fixed, thus the grid-hippocampal circuit forms a prestructured and invariant "scaffold". Non-grid cell-hippocampal connections are bidirectionally plastic and modifiable with associative plasticity rules. The grid circuit possesses a low-dimensional "shift mechanism" (marked v in a nod to its role in velocity-based updating in the spatial context), which shifts the grid states (phases) along each module's two-dimensional grid representational space. (c) Circuit architecture for updating grid cell phases for high-capacity content-addressable (hetero)associative memory. (d) Circuit including shift mechanism linked to self-movement inputs for spatial memory (e) Circuit with hippocampal states driving shift mechanism to efficiently drive transitions for high-capacity episodic memory. (f) Circuit model of the memory palace mnemonic strategy, in which a previously learned spatial memory is can be repurposed as a scaffold for high-fidelity one-shot memory.

world, but also internally generated cognitive inputs from other brain regions[66]. The hippocampus also receives inputs from entorhinal grid cells (Fig. 1b, orange). It connects back out to both grid and non-grid cells.

The grid cell circuit consists of multiple grid modules[81], comprising disjoint groups of cells. Each grid module expresses an *invariant* set of low-dimensional states regardless of task. This invariance is established in an extensive set of studies of the population states and cell-cell relationships of co-modular grid cells across behavioral conditions and states, including navigation in familiar and novel environments, across different spatial dimensions, and across sleep and wakefulness[35,36,62,64]. In spatial contexts, we can describe grid cell modules as coding position as a phase modulo their spatially periodic responses[61,82,83]. In non-spatial contexts, the states of a grid cell module remain the same but can be conceptualized as abstract representations constrained to lie on a 2-dimensional torus.

Connections from grid cells to hippocampus are set as random and fixed. Connections from hippocampus to grid cells are set once (e.g. over development) through associative learning, and are then held fixed. As we will see, the fixed internal grid connectivity and random fixed projections from grid to hippocampal cells are critical for many important properties of the circuit. Connections between hippocampus and non-grid inputs remain bidirectionally plastic and set by associative learning.

Because the grid cell states are fixed and the grid-hippocampal weights are bidirectionally fixed, we refer to the grid-hippocampal circuit as the *scaffold* of the memory network. This architecture, involving a set of fixed low-dimensional states (grid cell circuit) that is recurrently coupled through fixed high-rank projections to the hippocampus, creates a large bank of well-behaved fixed points, as we will see next. Separately, we refer to the hippocampal-non-grid cortical feedback loop as the *heteroassociative* part of the circuit. In this circuit, a separate set of connections than those generating fixed points heteroassociatively attach sensory data to the scaffold. Unlike standard associative memory models like the Hopfield network[84] in which the recurrent weights stabilize and associate content directly, here the two are separated: Vector-HaSH *factorizes* recurrent dynamics from content.

We next explore the theoretical and empirical properties of this circuit architecture and its extensions for content-addressable memory in various settings, from spatial to non-spatial memory to sequential episodic memory, Fig. 1c-f.

Generation of vast library of robust fixed points in an invariant scaffold

The grid cell circuit consists of a few (M) putatively independent grid cell modules: the population states of the neurons in each module are constrained to lie on a 2-dimensional torus. Each grid module can express just one state on the torus at a time, independent of the other modules. The i^{th} module can take one of K_i states, thus together they express $\prod_i K_i \sim \langle K \rangle^M$ many, or exponentially many, distinct states (assuming the K_i's are coprime). Coupling co-active grid cells from all modules to a hippocampal cell through learned bi-directional connections could turn that grid-hippocampal state into a fixed point and enable error-correction[83,85]. However, the hippocampus does not possess enough cells to convert each grid state into an attractor in this way.

In the scaffold hypothesis, grid cells project with fixed random weights – a high-rank random projection – to hippocampal cells, which threshold and rectify their inputs. The return projection is learned once through simple Hebb-like learning to reinforce the input grid cell state, then held fixed (Methods).

Random fixed scaffold converts exponentially many grid states into exponentially many stable fixed points

Remarkably, the random grid to hippocampal projections combined with associatively learned return projections in the scaffold converts *all* the exponentially many grid states (as a function of number of grid modules) into stable fixed points or attractors of the entorhinal-hippocampal circuit, Fig. 2b, for a sufficiently (but not very) large hippocampal network, Fig. 2c. Adding noise to a hippocampal state derived from any grid state, then running the dynamics of the circuit, exactly restores the correct (denoised) hippocampal state for the original grid state.

The required number of hippocampal cells is much smaller than the exponential set of grid states (Fig. 2d, left; SI Fig. S1): it scales only linearly with the number of modules, and therefore logarithmically with the number of grid states. It is also nearly independent of the scale (periodicity) of the grid cells for a given number of modules (Fig. 2d, right, SI Fig. S2; analytic proof in SI Sec. C.1). In sum, the number of stable states generated by the scaffold is exponential in the combined number of scaffold neurons (grid and hippocampal cells).

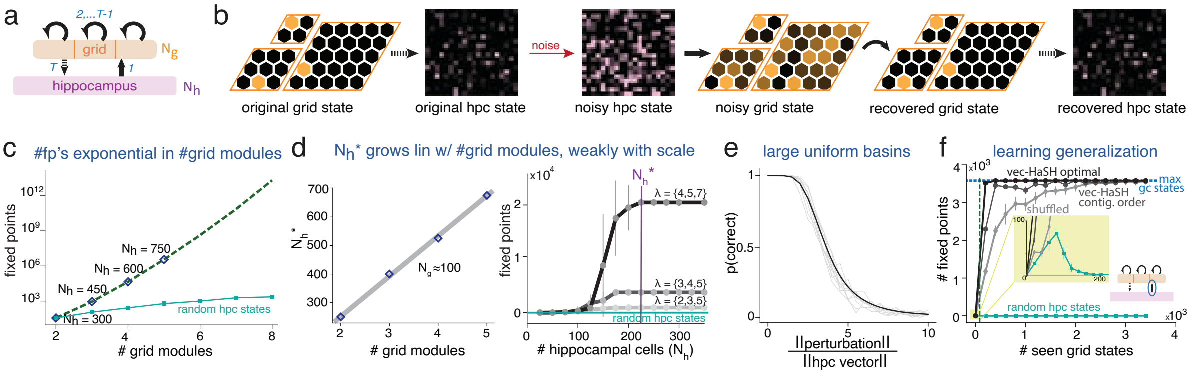

Figure 2. Scaffold generates exponentially many fixed points with large equal-sized basins. (a) The grid cell-hippocampal scaffold circuit, with pre-structured grid cell states, random fixed grid projections to hippocampus, and fixed return projections. Numbers indicate flow of dynamics (order of updating) in the circuit. (b) Illustration of states and dynamics in circuit. A noisy version of a hippocampal state is cleaned up in one round-trip pass through the scaffold. (c) The exponentially many coding states (as a function of grid module number)[82,83] are converted into exponentially many stable fixed points by the scaffold architecture for a sufficiently large number of hippocampal cells. States are defined as stable fixed points if they correct noise of magnitude 25% of the typical hippocampal state magnitude. (Grid periods for k modules are set as the first k prime numbers, number of hippocampal cells set to $150 \times k$). Numerical results (diamonds) exactly coincide with analytical theory (forest green dashed curve, derivation in SI C.1); zero variance across different random initializations. Light green: Learned bi-directional grid-hippocampal connectivity (with hippocampal states selected as shuffled versions of the states set by random grid cell projections) destroys the capacity of the scaffold (also see SI Fig. S3). (d) Left: Required hippocampus size (N_h^*) grows only linearly with grid modules, and thus only logarithmically with the number of fixed points. Thus, the number of scaffold fixed points is exponential in the network size. Right: Number of stable fixed points as a function of number of hippocampal cells (fixed points defined as in (c)). At a fixed number of modules, N_h^* is nearly independent of grid periods (gray curves), even though the number of fixed points grows with period (SI Fig. S2 and C.1 for theoretical derivation). Light green: as in (c). (e) Basin size, convexity, and uniformity: Fraction of states that return to the correct noiseless state after iteration as a function of the noise magnitude (normalized by the magnitude of the hippocampal vector) computed over 100 random noise realizations. Grey curves: five randomly selected fixed points. Black: mean over all (3600) fixed points. All basins are convex and uniformly sized. (f) Number of grid states stabilized as fixed points as a function of number of states over which the hippocampus-to-grid weights are learned: Learning from a vanishing ratio of seen states to all possible grid states ($\sim MK_{max}/e^M$) turns all possible grid states (horizontal dashed blue) into fixed points – termed strong generalization. Black: Vector-HaSH with optimal sequence of seen grid phases. Dark gray: contiguous sequence of grid phases. Light gray: random sequence of grid phases. Light green: as in (c). Vertical dashed forest green: theoretical minimum number of seen grid phases for strong generalization (Fig. S4 and SI Sec. C.4 for analytical derivation).

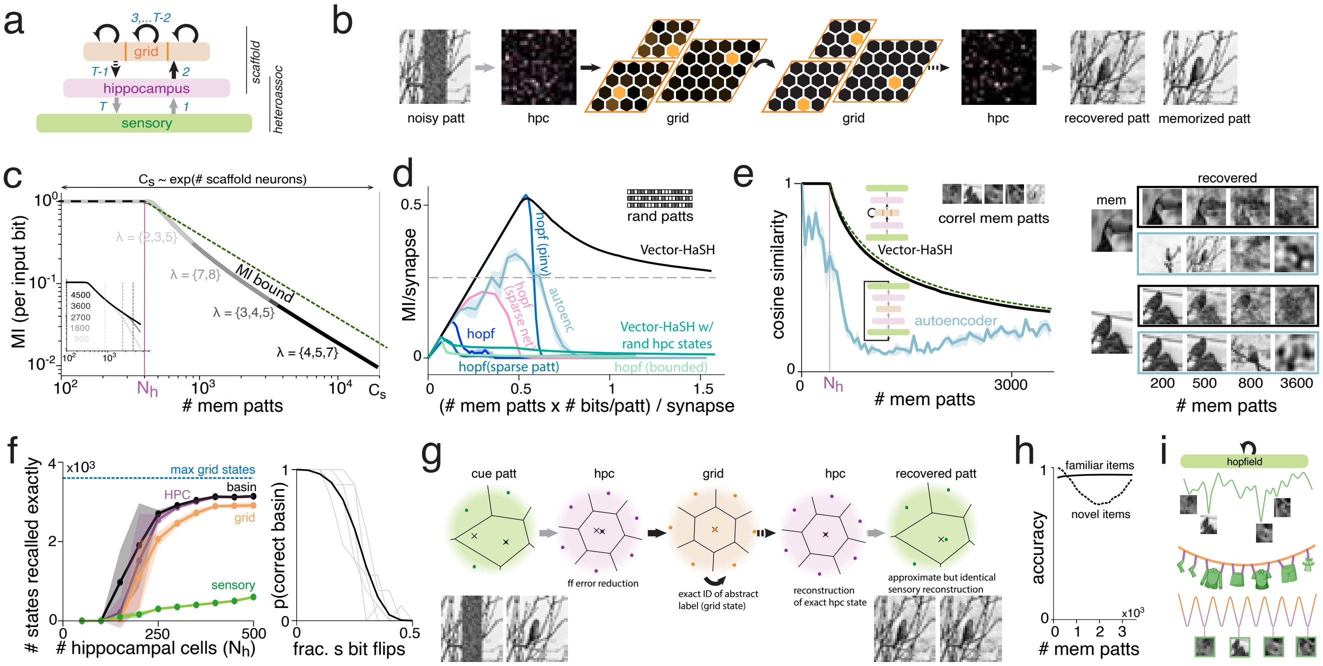

Figure 3. High-capacity content-addressable item memory via heteroassociation with scaffold. (a) Network architecture. Numbers indicate flow of dynamics (order of updating) in the circuit. (b) Associative content-addressable recovery: A corrupted or partial input is completed by the circuit. (c) Mutual information per input bit (MI divided by total # of input bits = MI/# patterns/bits per pattern) between memorized and recovered patterns as the number of memorized patterns is varied from 0 to the exponentially many scaffold fixed points; beyond the knee (corresponding to the location of the "memory cliff" in Hopfield networks), MI per input bit scales inversely with the number of input bits: this is the "memory continuum" of Vector-HaSH. Black dashed line: the theoretical upper bound for information storage (=# synapses/(2×# patterns))[74,75]. Inset: Effect of varying the size of the sensory area (N_s; indicated numbers), see also SI Fig. S7. (d) MI (per synapse) as a function of total number of input bits per synapse for various memory models. Most models supply zero MI per synapse after a threshold number (memory cliff). Vector-HaSH asymptotically approaches a constant MI per synapse (dashed gray line), scaling as the theoretical optimum. (e) Performance comparison of Vector-HaSH (visualized with unfolded architecture; black arrows and boxes designate predefined weights and activations) and a tail-biting autoencoder[86] with the same number of layers and neurons, trained end-to-end with supervised learning and unconstrained activations and weights; black arrow designates an identity weight matrix. Dashed forest green: analytical capacity result for Vector-HaSH. Right: reconstructed pattern after storage of a variable number of patterns (black outline: Vector-HaSH; blue: autoencoder). (f) Left: Number of patterns that can be memorized with perfect recovery (0-error) for grid, hippocampal, or sensory recall, respectively, when cued with sensory inputs with 2.5% noise. Most grid and hippocampal states are exactly recalled with big enough N_h. Sensory states are recalled partially, depending on position along the memory continuum, but fall in the correct basin (black curve). Right: Correct basin for recovered sensory state even for large noise in the input (shown for 500 memorized patterns; see SI Fig. S8 for the dependence on this basin structure as a function of the number of stored patterns). (g) Schematic of recovery in state space, left to right: Initial cues (bottom: masked and noiseless pattern) and where they fall in sensory space (top). The projection to hippocampus reduces noise; subsequent mapping to grid states completes a nearest-neighbor computation to the nearest grid state; this state drives the exact associated hippocampal state. The reconstructed sensory state is some distance from the memorized pattern (distance grows with number of patterns relative to N_h). Though inexact, recovery is reliable: all cued patterns in the same basin converge to the same recovered state (bottom). (h) Simple decoder based on the mean hippocampal population firing rate permits discrimination of familiar patterns (rate distributions for familiar vs. novel patterns in SI Fig. S13). (i) An energy landscape perspective: In Hopfield networks, the width, depth, and positions of the minima depend on pattern content. In Vector-HaSH, the scaffold generates a well-behaved landscape of large equally-sized minima. Arbitrary content can be "hooked" onto the minima, in analogy with a clothesline on which any clothing may be hung. In (c-f) errorbars (shaded) are SD over 5 runs.

We find theoretically that there are no spurious stable states, thus the entire hippocampal state space is devoted to forming large basins for the grid cell states. We also find theoretically that the basins are all convex and essentially identical in size across fixed points, SI Sec. C.2, Sec. C.3. Thus, the basins of each stable state should be uniformly and maximally large, which we confirm numerically in Fig. 2e.

Though remarkable, this result does not violate fundamental capacity bounds for Hopfield-like recurrent networks, according to which a network of $\mathcal{O}(M)$ neurons and $\mathcal{O}(M^2)$ synapses can support at most $\mathcal{O}(M)$ user-defined points as stable states of the dynamics, or $\mathcal{O}(M^2)$ bits of information[74,75], since the fixed points are pre-determined content-free states and not user-defined. The specific structure of the pre-determined grid states, randomly projected to the hippocampus, creates well-spaced robust grid-hippocampal attractors with large even-sized basins.

The specific structure of the fixed weights and activations in the scaffold is critical for its function: if instead of being determined by random grid projections, the hippocampal states are set to be randomly shuffled per-state versions of these states, preserving their statistics, and the grid-to-hippocampal and return weights are bidirectionally learned to make the grid-hippocampal pattern pairs self-consistent, the scaffold capacity collapses (Fig. 2d; theory in SI C.1).

Strong generalization property of scaffold The scaffold network possesses another remarkable property dependent on grid coding, which we call strong generalization. The hippocampal-to-grid weights are set by visiting the grid states, determining the hippocampal states via the grid-to-hippocampal random projection (with thresholding), then applying Hebb-like associative learning. We find that all $\mathcal{O}(K^M)$ grid states become stable fixed points of the iterated dynamics after learning the weights for only $\mathcal{O}(MK_{max})$ of the states, where K_{max} is the number of states in the largest module. When M and K are large, this is a miniscule fraction of all the states that become automatically stabilized, a very strong type of generalization (Fig. 2f). We derive a theoretical proof of this strong generalization property in SI Sec. C.4. For strong generalization, the grid states visited during learning could simply be generated by traversing a small contiguous set of locations (see SI Fig. S5 for an illustration of the minimal spatial region required for learning, compared to the full extent of the grid coding space). Learning from a random subset of grid states requires seeing many more patterns before all grid states become stable fixed points – there is some generalization, but it is substantially weaker (Fig. 2f). (Certain special sets of non-contiguous locations can lead to strong generalization SI C.4, SI Fig. S5.) When grid states are replaced by fixed patterns of otherwise identical sparsity (e.g., obtained by shuffling each grid coding states), as in MESH[87], there is almost no generalization: stabilizing exponentially many states requires learning from exponentially many patterns, close to a fraction of one. Theoretically, the metric or ordered (and thus indirectly, spatial) structure of the grid phase code, with learning of hippocampal-to-grid projections learned while visiting states in a metric order leads to strong generalization (proof in SI Sec. C.4).

The property of strong generalization is computationally useful and biologically critical: it means that the scaffold weights can be learned once, for instance from early spatial exploration within a small environment, then held fixed for the rest of the animal's life. This early and spatially restricted learning is then sufficient to provide a massive library of stable fixed points for future spatial and non-spatial memory function over the rest of the animal's life, as we will see below.

Content-addressable item memory through heteroassociation of inputs onto scaffold

A content-addressable memory must enable the storage and recall of arbitrary (user-defined) input patterns based on partial or corrupted patterns. Scaffold states are not themselves memory states because they are not user-defined.

Consider external inputs to the hippocampus, which arrive directly from neocortex[88–90] or via non-grid entorhinal cells, Fig. 3a (green). We will call these sensory inputs for short. An incoming sensory input is 'assigned' to a randomly chosen scaffold fixed point via Hebb-like one-shot learning between the input and the hippocampal state by a biologically plausible online Hebb-like implementation of the pseudoinverse rule[91,92] [1]. The goal of these weight updates is self-consistency: the drive from hippocampus back to the sensory states should attempt to generate

---

[1] We show in SI Sec. D.5 that one could simply use Hebbian learning instead of an iterative or standard pseudoinverse learning, while maintaining the same asymptotic capacity, with a smaller constant pre-factor, Fig. S6

the same sensory pattern that activated the hippocampal state. As additional sensory inputs are received, they are assigned to other scaffold states, and the weights between sensory inputs and the hippocampus are correspondingly updated. Inputs for item memory can be associated to any scaffold state, in any order (scaffold states need not be selected in a metric sequence for e.g. item memory).

Memories are reconstructed from partial sensory cues as follows: these inputs drive hippocampal activity, which drives the scaffold toward a valid fixed point; sensory information is then reconstructed via the heteroassociatively learned hippocampal-to-sensory weights. Vector-HaSH thus behaves as a content addressable memory (CAM) network, Fig. 3b. Reconstructions can be done bidirectionally: If the network is cued only by sensory inputs (the initial grid state is unknown), the network reconstructs the grid state. Conversely, given only the grid state (no sensory cues), the network reconstructs the sensory data.

Continued high-information associative memory via graceful item number-information tradeoff Memory recall is perfect up to N_h memorized patterns (N_s correctly recovered bits per pattern and a mutual information of 1 per input bit, where N_s is the size of the sensory input). Beyond N_h stored input patterns, the recovered information per pattern scales as the reciprocal of the number of patterns: there is a graceful tradeoff between number of stored patterns and recall richness. The mutual information (MI) per input bit between recalled patterns and memorized patterns saturates or nearly saturates the theoretical upper bound (given by the square of the number of synapses divided by twice the number of patterns), at all numbers of memorized patterns up to the number given by the exponentially many scaffold fixed points, Fig. 3c. It asymptotically scales the same way as the MI upper bound (Fig. 3c, black dashed line). Pattern memorization can occur online (patterns presented once, in sequence) or offline, with no difference in recall performance.

Comparison with existing memory models We can compare the performance of Vector-HaSH with Hopfield network models of various varieties: the classical Hopfield network, those with sparse weights, and those with sparse patterns. In all these networks, when the number of memorized patterns remains below a threshold (linearly proportional to network size), they are perfectly recovered. Memorizing patterns beyond this critical number leads to loss of all patterns, including those previously memorized: the mutual information between recalled and memorized patterns drops to zero, — a memory "cliff" (Fig. 3d). Other memory models exhibit a similar cliff[86, 93–98]. Others can store only a specific number of patterns for a fixed network architecture[76].

In Vector-HaSH, the network continues to store finite information per input bit, until the number of stored patterns equals the exponentially large scaffold number capacity. The recovered information per synapse approaches a constant value regardless of the number of stored patterns up to the scaffold number capacity, Fig. 3d – there is no memory cliff [2], and the total information in the network scales as the optimal value of N^2 regardless of number of patterns, Fig. 3c (dashed line). The information recoverable per network synapse tends asymptotically to a constant value rather than zero (Fig. 3d (dashed line)). We call the constant retrievable information in Vector-HaSH and the smooth tradeoff of information per pattern with number of patterns a memory "continuum", in contrast to the memory cliff of other CAM models[87].

Comparison with end-to-end trained deep networks Vector-HaSH can be unfolded for interpretation as an autoencoder[86], but a highly constrained one: the encoding in the bottleneck layer is fixed, with fixed recurrent dynamics within the layer. The weights from the bottleneck to decoder layers are fixed, and all weights are learned through biologically plausible associative rules, Fig. 3e (left). For comparison, we train an unconstrained autoencoder of the same dimensions with a tail-biting connection (identity weights from the output of the autoencoder to its inputs) to enable iterative reconstruction[86] (Fig. 3e (left)) via supervised learning (backprop). Strikingly, Vector-HaSH substantially outperforms this autoencoder, despite the latter's much larger potential flexibility, Fig. 3e (right),

---

[2] Since input sensory patterns are stored via association to scaffold states, once all scaffold states have been associated with an input there are three possibilities. The first is that the memory is saturated and no further sensory patterns can be stored. The second involves selecting an existing scaffold state randomly or based on sensory overlap and rewriting the previous memory's heteroassociative weights to accommodate the present sensory input. The third involves a gradual decay of the heteroassociative weights between sensory inputs and the scaffold, so that older memories are lost and those scaffold states are identified for reuse.

mirroring the significant advantage of MESH over the same autoencoder[87]. SI Fig. S9 shows that Vector-HaSH also outperforms both the tail-biting autoencoder and the non-iterated autoencoder when cued with noisy sensory cues.

In sum, the fixed attractor states in the scaffold (grid cells in Vector-HaSH) appear to provide a key inductive bias for robust high-capacity memory that learning with backpropagation on an unconstrained architecture cannot find or achieve.

Mechanisms of continued high-information associative memory When a partial or corrupted sensory state is presented to Vector-HaSH, it retrieves an item from memory. In different parts of the network, the precision of retrieval is different (Fig. 3f, left): grid and hippocampal states are recalled exactly almost always (with a threshold number of hippocampal cells). The sensory state is recalled only approximately, depending on the number of stored patterns relative to the size of the hippocampal area (the memory continuum; Fig. 3f, left, green). Though sensory reconstruction is approximate, the retrieved state is in the same Voronoi region as the memorized sensory pattern (Fig. 3f, left, black). This is true even when the fraction of error bits in the cue is large (Fig. 3f, right).

Unpacking the dynamics underlying these results: The projection of a noisy input onto the scaffold via the sensory-to-hippocampal weights already reduces some errors (Fig. 3b,g: input states move toward learned states, in the transition from green to pink basins). Next, the grid recurrent dynamics finds an exact grid coding state closest to the hippocampal inputs (Fig. 3g, transition from pink to orange basins). Further, this mapping is to the grid state in the correct basin (cf. Fig. 3f) even deep in the memory continuum, which we show analytically in SI Sec. D.1. Next, this grid state reconstructs an exact previously memorized hippocampal state (Fig. 3g, transition from orange to pink basin). Finally, the hippocampal to sensory states reconstruct an approximation to the sensory pattern corresponding to the hippocampal state. The *precision* of this reconstruction systematically decreases with the number of stored patterns because of interference in the reconstruction weights across patterns – this accounts for the memory continuum – but the reconstruction is *reliable*: regardless of the cue (which might be noiseless or corrupted), the reconstructed pattern is the same (Fig. 3g, transition from pink to green basin).

We prove that high-information associative memory is possible when the set of grid-driven hippocampal states is *strongly full rank* (SI Sec. D.2), a property satisfied if the hippocampal scaffold states: 1) are determined by random projection from grid cells and 2) involve some nonlinear transformation of the grid inputs (almost every nonlinear transformation in the space of all functions is sufficient, without fine-tuning the functional form). For instance, simple rectification with threshold is sufficient, for a wide range of activation thresholds (SI Figs. S10,S11; in contrast, a linear hippocampal layer does *not* result in strongly full rank hippocampal states, SI Fig. S12). The strong full rank property of the scaffold permits sensory inputs to be stored via association with scaffold fixed points in any order and at any location in the scaffold. We prove in SI Sec. D.2 that it guarantees perfect recall of the first N_h stored states, Fig. 3f and continued high-information associative memory beyond.

High-capacity recognition memory

The observation that the circuit recollects the correct basin for a previously seen input, even as the detail of recall declines, suggests that the circuit might also be exploited as a vast recognition memory system. We explored whether a simple statistic of hippocampal activity might differentiate between responses to familiar and novel inputs. We found that the activity distributions in the hippocampus are different for familiar versus novel sensory inputs[99]. When a large (small) number of patterns have been stored in the sensory to hippocampal layer weights, the mean hippocampal activity for novel sensory patterns is larger (smaller) than the mean activity for familiar sensory patterns (Fig. S13). Thus, we classify a pattern as familiar if the activity evoked in hippocampus lies within a narrow band around the typical average activation for familiar patterns. Outside this band, we classify it as novel. We found that this simple two-threshold classifier on hippocampal activity successfully classified a large fraction of inputs, with only a few errors, Fig. 3h.

Conceptually, we may understand the strong performance of Vector-HaSH compared to conventional (e.g. Hopfield) autoassociative memory networks as follows: the latter perform poorly because their fixed point landscape is governed by the content of the patterns, leading to highly uneven and small basins sizes, with many spurious minima. In Vector-HaSH, the landscape is set by the scaffold, which has large and well-spaced basins, and content is simply "hooked" onto these prestructured states. The analogy is with a clothesline (the scaffold), to which any clothes (sensory patterns) can be attached (via heteroassociation), Fig. 2h.

Spatial memory and inference

We now consider how this circuit performs *spatial* memory. Here, the metric or *vector* ordering of grid states, which was not necessary for pattern memory, becomes critical. When self-motion signals during spatial navigation are allowed to drive transitions between the metrically ordered grid states, Fig. 4a, we find below that the architecture and dynamics of Vector-HaSH support high-capacity life-long spatial memory without catastrophic forgetting and zero-shot spatial inference along novel paths.

In a novel room, we randomly initialize grid module phases, and velocity inputs to each module then update the grid phases through path integration[101]. Vector-HaSH learns associations between these updating grid states and spatially sparse sensory cues in the room, through the hippocampus, building up a map of the space which allows for bi-directional recall of grid states from sensory cues and vice-versa (Fig. 4a).

After even very sparse exploration in the room (Fig. 4c, left), Vector-HaSH is able to predict expected sensory observations when taking an entirely novel route through the room, on paths and path segments not previously taken (Fig. 4c, right). This zero-shot spatial inference ability arises from velocity integration[102–104]: the initial grid phases are updated based on velocity to generate accurate phases at locations even along novel paths, which can then drive reconstruction or prediction of the sensory cues associated with those phases from the past.

Next, we consider the spatial maps constructed by the circuit, and its ability to sequentially learn a series of different rooms, Fig. 4d. Grid cells and hippocampal cells exhibit realistic spatial tuning curves, including the spatially localized and typically single-bump tuning of place cells (Fig. 4b). For each room, Vector-HaSH learns a repeatably reliable spatial map, as assessed by testing after it has learned the room (Fig. 4e: compare the first two columns for each cell).

To assess sequential memory, and the extent of interference and overwriting of memories for different rooms acquired one after another, we assessed recall in all prior rooms after learning each of the 11 rooms sequentially (Figs. 4e, third column and 4f). The hippocampal and grid cell activations (maps) remain unchanged despite the subsequent acquisition of up to 10 new rooms. Notably, there is a complete absence of catastrophic forgetting without requiring replay or consolidative associative learning during recall assessments to refresh prior memories. This lack of catastrophic forgetting is due to the specific architecture and exponential scaling capacity of Vector-HaSH, in which random grid phase initializations result in maps that are well-separated in the coding space, Fig. 4g.

These properties correspond to observed responses in the entorhinal-hippocampal circuit, including the similarity of responses for repeated visits to the same room and orthogonal representation of different rooms, Fig. 4h-i. Additional properties of the hippocampal response, including the distribution of probabilities that a hippocampal cell has a field in multiple rooms, matches experimental data, Fig. 4j.

High-capacity sequence scaffold via vector updating of grid states

Sequence memory is typically modeled with asymmetric Hopfield networks[105–107], resulting in similar capacity limitations as standard Hopfield networks[74, 86, 108]. Remarkably, it is possible to construct a massive sequence memory in Vector-HaSH in a similar way as item memory: by factorizing the problem into a high-capacity abstract scaffold sequence then affixing content via heteroassociation. We first explore how to construct these high-capacity sequence scaffolds.

Memory networks perform poorly when user-defined patterns determine the attractor states (Fig. 2e-f). The equivalent problem in sequence memory is when user-defined patterns determine and drive the next user-defined pattern: a recurrent Hopfield-like auto-associative hippocampal network with asymmetric weights quickly (within ~ 50 steps) results in failure to accurately reconstruct the next step, Fig. 5a.

We reasoned that coupling hippocampus recurrently with the grid cell modules in the same way as the scaffold network might support high-capacity sequence reproduction, by denoising and pattern-completing otherwise inaccurate next-step patterns. Doing so roughly doubled the sequence capacity of the circuit (to ~ 100 steps), but did not fundamentally alter the scaling of capacity with network size, Fig. 2b.

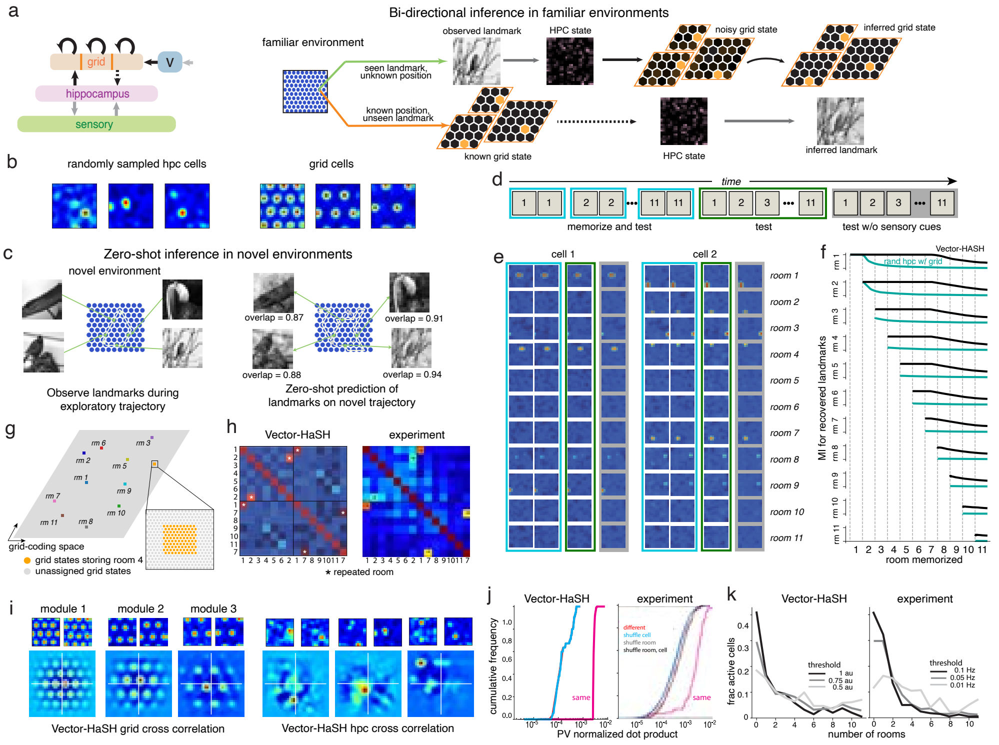

Figure 4. High-capacity spatial memory, spatial inference, and lifelong learning of environments without catastrophic forgetting. (a) *Left:* Vector-HaSH with a path-integration mechanism for velocity-driven shifts in grid phase. *Right*: Vector-HaSH performs bidirectional spatial inference of position (internal grid states) from sensory cues, and predicts sensory landmarks given a grid state, after simple associatve-like learning in an environment. (b) Randomly sampled hippocampal and grid cell tuning in one environment. (c) *Left:* First-time traversal of an environment along only the shown trajectory, accompanied by associative learning over the trajectory. *Right:* Vector-HaSH accurately predicts all landmarks during traversal of an entirely novel path from one of the landmarks, a form of zero-shot inference. For realism, we show the model 596 other patterns (mimicking memorization of different environments) before it sees these 4 landmarks. This results in approximate sensory recovery as the model is in the memory continuum regime. (d) Sequential learning protocol for 11 rooms: Vector-HaSH is steered along a random trajectory in room 1, then tested on a different random trajectory in the same room. During memorization, there is a landmark at each location of every room. Rooms 1 through 11 are memorized in sequence without revisiting prior rooms. During testing, only the first landmark in a room is visible, the rest must be predicted. At the end, the 11 rooms are tested again, then re-tested in the "dark" in each of the 11 rooms. (e) Spatial tuning of two hippocampal cells, across rooms and across learning and all three testing conditions. (f) Sequential recall curves for map i after learning maps $1 \cdots i-1$: there is no interference with or catastrophic forgetting of previously learned maps in Vector-HaSH and a baseline model consisting of random sparse hippocampal states, similar to the baseline considered in Figs. 2 and 3. (g) Random assignment of the starting grid states in across the vast grid coding space results (gray diamond represents the set of all possible grid states across modules – the side-length of the diamond is the exponentially large unique coding range per dimension of the grid code, across modules) in non-overlapping grid representations and the lack of catastrophic forgetting. (h) Hippocampal population activity similarity matrix across rooms (including repeated exposures to some rooms) from Vector-HaSH (left) and experiments[100] (right). (i) *Left*: Top: spatial maps of a grid cell from each of three modules across a pair of rooms (rooms 8,9), and bottom: cross-correlation of that cell's response across rooms. The modules exhibit differential shifts in phase. *Right*: Across-room cross-correlations of three hippocampal cells. (j) Hippocampal representation similarity for same versus different rooms and shuffle controls for Vector-HaSH and experiments: representations of different rooms are as orthogonal as the shuffles for both. (k) Distribution of number of hippocampal cells active in R rooms as a function of R, for Vector-HaSH and experiments. In both, most cells were active in only a few rooms.

Next, we reasoned that learning an abstract sequence of scaffold states rather than user-defined states in the full spirit of a scaffold network might be the solution. We tested the performance of learning transitions from one abstract grid state to the next, using the projections from hippocampus to grid cells to cue the next grid state, with the full benefit of the scaffold architecture. The sequence capacity remained qualitatively similar to Hopfield networks, Fig. 5b, with failure within ~ 30 steps. This is possibly because even abstract grid states are specific large activity patterns, which the previous state must sufficiently specify to reconstruct. This failure and hypothesized reason gave us the critical insight that learning the *input to the velocity shift mechanism*, which requires specifying merely a 2-dimensional vector to specify the next grid state given the current one, would minimize the information, would potentially alleviate the capacity limitation for sequence reconstruction.

We therefore used the previous grid state to cue the next grid state, but via the drastic dimensionality and complexity reduction of the velocity-shift mechanism: the previous grid state was used to specify a 2-dimensional velocity that signals where to move in the grid coding space to generate the next grid state. We built these associations via a simple feedforward network (MLP), Fig. 5a (top) that associated the previous grid state, via the hippocampus, with a 2-dimensional velocity vector. This architecture resulted in the accurate reconstruction of scaffold sequences of 1.4×10^4 grid states, using the same (small) number of cells in the scaffold network as before, Fig. 5a (left).

In other words, recalling a long abstract sequence of grid states can be achieved by solving the much simpler task of recalling a sequence of simple abstract two-dimensional vectors, each of which points from one state to the next in the grid coding space. This enables much longer sequence reconstruction because the information the network must recall for each step in the sequence is a mere two-dimensional vector, not the much larger grid pattern state, Fig. 6g. Consistent with this, when we parametrically vary the amount of information the network must recall at each step to arrive at the next, by increasing the range of possible 2D vector lengths to be recalled, the fraction of recalled sequence length decreases gradually (Fig. 5e; theoretically expected scaling, in which sequence length decreases inversely with the number of bits required to specify the next step, is shown in blue). Thus, constraining the sequence recall dynamics to a low-dimensional manifold where only low-dimensional velocity tangent vectors rather than the manifold states themselves must be reconstructed, results in vast increases in sequence length.

Quantitatively, we can take a statistical approach and assess how well the circuit can recall random velocity (shift) vectors assigned to each grid state, Fig. 5c. The circuit memorizes and perfectly recalls sequences of length $\sim 1.5 \times 10^5$ with a scaffold consisting of $N_h = 500$ and $N_g = 275$ grid cells of periods $5, 9$ and 13 (and hence a total of $\sim 3.4 \times 10^5$ total scaffold states); longer sequences are reproduced with modest decreases in recall performance. The dependence on the number of hippocampal cells is again smaller than logarithmic, similar to scalings for item memory, Fig. 5d (left), and the dependence on the number of units needed to learn the dimension-reducing mapping from state to velocity vectors is linear, Fig. 5d (right).

Given that the velocity transitions are 2-dimensional, we can plot the grid and hippocampal states as a function of these 2-dimensional transitions (which may or may not correspond to physical 2-dimensional space), and as expected given the invariance of grid modules, we obtain periodic grid responses in this space, Fig. 5f; hippocampal cells exhibit more-sparse and more-localized tuning relative to grid cells, Fig. 5g.

In sum, the velocity shift mechanism of the grid cell integrator networks enables memorization of exponentially large cumulative-length sequences of abstract grid states, relative to network size. Remarkably, therefore, the path integrability of the grid cell code can support not only highly efficient spatial inference and mapping but also sequence memory, even if the sequences do not involve physical navigation in real spaces. This defines a high-capacity abstract sequence scaffold.

Episodic memory

Just as high-capacity item memory was supported by the factorization of dynamics and content into a fixed-point scaffold and heteroassociation, Vector-HaSH supports high-capacity episodic memory by factorization into a sequence scaffold (above) and heteroassociation. As before, abstract grid cell and hippocampal scaffold states are heteroassociatively linked to sensory inputs during a temporally unfolding event.

This temporal unfolding through scaffold states can occur through next-step velocity recall based on hippocampal states (as seen in Fig. 5); however in this case the total sequence capacity is limited by the number of MLP units

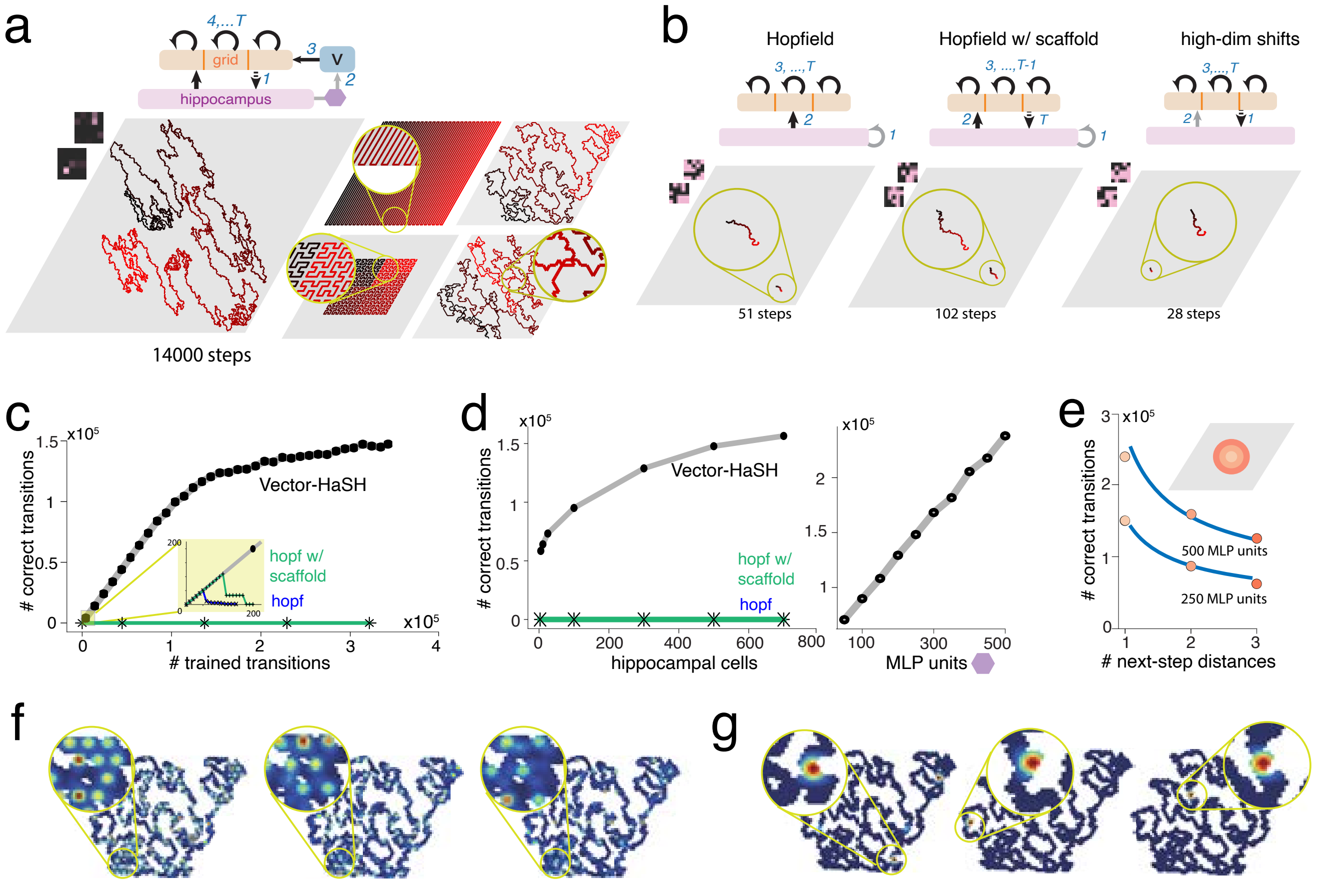

Figure 5. High-capacity scaffold for sequences via low-dimensional velocity update mechanism in grid cells (a) Architecture and performance of the Vector-HaSH sequence scaffold, where transitions of grid states are driven by hippocampus via the two-dimensional grid cell velocity shift mechanism[33, 109]. Hippocampal states are converted to velocity states by a small feedforward network (MLP, N_M=250), violet. Left to right: Recall of a self-avoiding random walk (Lévy flight), a hairpin curve sequence that spans the entire grid coding space, a Hilbert space-filling curve that spans every point in a subset of the grid coding space, and a uniform self-avoiding random walk (in the first, parameters are as in (b); in the last three, $\lambda = [3,4,5]$ with a 3600 sized state space). (b) Architectures and recall performance of hippocampus as an asymmetric Hopfield network (left), as an asymmetric Hopfield network assisted by the scaffold (center), and of the whole scaffold network with hippocampal-grid cell connections driving transitions from one grid cell state to the next (right). In all cases, the network readout is visualized by the grid state (a 585×585 state space, $\lambda = [5,9,13]$) and the asymmetric Hopfield network is constructed on $N_h = 500$ nodes. (c) Left: Each point in grid coding space is assigned a random shift to link it to the next step. Vector-HaSH successfully recalls large sequences while the Hopfield baselines fail to do so. (d) Left: the recalled sequence length increases exponentially with the number of hippocampal cells, when trained on all 585×585 random shifts. Right: The recalled sequence length increases exponentially with the size of the MLP that maps hippocampal states to a 2-dimensional velocity vector, when trained on all 585×585 random shifts. (e) Increasing the information load of next-state generation by increasing the range of potential shift step-sizes results in reduced performance. Blue curves: performance declines as the log of the number of distinct possible shift vectors. (f-g) Sample grid and hippocampal cell tuning in the abstract 2D space defined by the grid velocities. This space is abstract: it can, but need not, correspond to physical space.

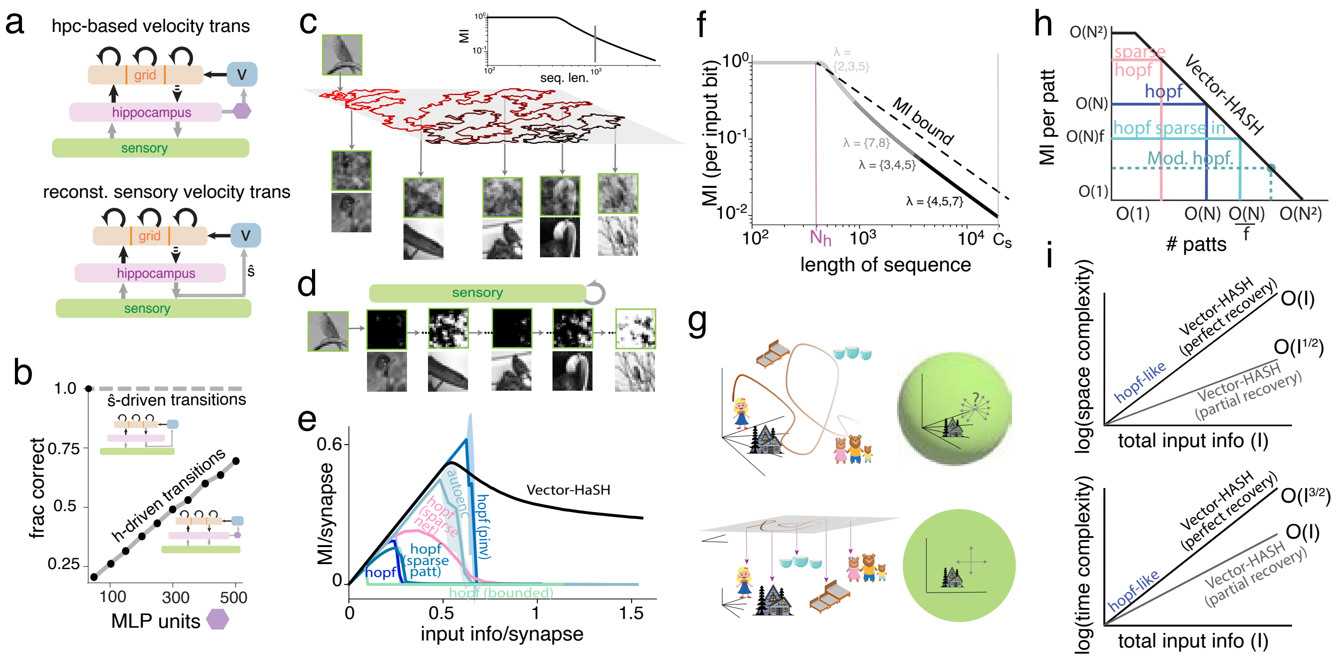

Figure 6. High-capacity episodic (sequence) memory via heteroassociation with sequence scaffold. (a)Architecture of Vector-HaSH for episodic memory. *Top*: Hippocampal states determine a 2-dimensional shift vector $\vec{v}$ for grid cells, via a decoding network (3-layered MLP, purple hexagon). *Bottom*: Alternatively, the reconstructed sensory state ($\hat{s}$) determines the 2-dimensional shift vector $\vec{v}$ (this resembles route learning, SI Fig. S14). (b) The accuracy of hippocampus-driven shifts depends on the size of the decoding network (MLP); the accuracy of sensory recall-based shifts can be consistently high because of the size of the sensory network. Because shifts are learned from the recalled sensory states rather than true sensory states, and sensory recall is reliable even when inaccurate (Fig. 3g), the shifts remain accurate deep into the memory continuum. The fraction of correct shifts is computed over learning a mapping from *all* scaffold states to random shifts, as in Fig. 5d. (c-d) Example recalled trajectory and sensory states in Vector-HaSH (c) and an asymmetric Hopfield network with the same number of synapses (d) for `bw-mini-imagenet` sensory inputs (see Methods; periods [3,4,5], N_g=50,N_h=400,N_s=3600 for Vector-HaSH, and N_s=3600 for Hopfield). (e) MI (per input bit) between the stored and recalled next-step sensory states as a function of episode (sequence) length when storing episodic memories, across models. (Sensory states are random binary patterns.) Vector-HaSH asymptotically approaches a constant amount of information per input bit. (f) MI between stored and recalled sensory states in Vector-HaSH as a function of sequence length (total number of stored memories), similar to the item memory curve of Fig. 3c. (g) In conventional sequence memory models, the recalled content of one item must be used to reconstruct the entire content of the next item. Vector-HaSH instead requires one item to specify or reconstruct a mere 2-dimensional vector, a task requiring far less information to be stored in the recurrent synapses. (h) As seen in (f) and Fig. 3c, Vector-HaSH thus achieves the memory continuum desired in Fig. 1a. (i) The space-complexity of storage and time-complexity of recall: *Top:* When perfectly recovering all stored patterns (when # patts $\leq N_h = N_s$), the storage space complexity of Vector-HaSH scales the same way as Hopfield models. When storing a larger number of patterns (when # patts = $N_s = C_s \gg N_h$), the space complexity of Vector-HaSH scales much more slowly. *Bottom:* The time-complexity of recall in Vector-HaSH scales similarly to Hopfield models for perfect recovery of a small number of patterns, but the scaling of time-complexity is much lower for partial recovery of a large number of patterns.

(Figs. 5d, 6b). We hypothesize an additional architecture wherein the next-step velocity recall is based on the recalled sensory inputs (Fig. 6a bottom, SI Fig. S14). In this case, all next-step transitions can be recalled due to the large number of sensory cells (Fig. 6b, proof in SI Sec. D.6). Note that, as earlier, the synaptic weights storing sensory information continues to be completely independent of the weights storing scaffold states

The heteroassociation part of the sequence scaffold is the same as heteroassociation for individual item memory, Fig. 6a, including one-shot learning via biologically plausible rules, robust recall of arbitrary inputs, and graceful memory degradation with increasingly many stored inputs. As in Fig. 3g, internal hippocampal states are computed with high fidelity throughout the memory continuum, while heteroassociative sensory recall fidelity degrades with sequence length Fig. 6f. Because sequence recall at each step is based on the (accurately reconstructed) previous hippocampal state, there is no degradation over time in the identity of the sensory state at each point in the sequence, Fig. 6c. The cumulative amount of sensory information recovered by the network during sequence recall asymptotically approaches a constant, Fig. 6e, as in item memory, Fig. 3f. For episodic memories with content that is not explicitly spatial, the sequential trajectory in the grid-hippocampal scaffold can be arbitrarily chosen — in our numerical simulations examining the maximal extent of sequential memory in Vector-HaSH (Figs. 6e-f), we choose a space-filling “hairpin” trajectory in scaffold space.

Thus, Vector-HaSH with heteroassociation onto the sequence scaffold generates a high-capacity episodic memory, Fig. 6f. This is in sharp contrast to the poor performance of Hopfield networks and tail-biting autoencoders attempting to memorize the same sequence of sensory inputs[86], Fig. 6d-e.

In sum, the hippocampal-entorhinal circuit in Vector-HaSH is able to store episodic memory of arbitrary input sequences with high capacity by exploiting the grid-hippocampal scaffold and vector updating of grid cell states, even in the absence of any spatial content in the memory.

Space- and time-complexity of memory in Vector-HaSH

The number of synapses in Vector-HaSH scales as $N_s N_h = \mathcal{O}(N_s)$ for large K and fixed M (Fig. 2d *right*, SI Fig. S2). The number of nodes $N_s + N_h + N_g = \mathcal{O}(N_s)$ scales the same way, meaning that Vector-HaSH is a highly sparse network. The number of stored patterns ranges from $N_h = \mathcal{O}(1)$ patterns relative to N_s (with information stored per pattern given by $\mathcal{O}(N_s) = \mathcal{O}(\#\text{ synapses})$ bits), all the way up to N_s patterns (with information stored per pattern being a nonzero amount), Fig. 6h.

For storage and perfect recovery of N_h patterns of length N_s, comprising $I = N_h N_s$ total bits, Vector-HaSH has a space complexity of $\mathcal{O}(I)$ (SI Sec. D.4). This is the same as Hopfield-like networks, which also require $\mathcal{O}(I)$ space to store I total bits, Fig. 6i (top). The time complexity for perfect recovery in Vector-HaSH scales as $\mathcal{O}(I^{3/2}\sqrt{N_h/N_s}) \leq \mathcal{O}(I^{3/2})$. When $N_h \ll N_s$, this scaling is faster than Hopfield-like models, which have a time complexity of $\mathcal{O}(I^{3/2})$, Fig. 6i (bottom).

When the number of stored patterns is much larger, scaling as cN_s for $0 < c \leq 1$, Vector-HaSH partially recovers the stored information (Fig. 3). In this regime, Vector-HaSH has further improved time and space complexity relative to the number of synapses: the required space complexity scales only as $\mathcal{O}(IN_h/(cN_s))$, and time complexity scales only as $\mathcal{O}(I^{3/2}N_h/(N_s\sqrt{c}))$ (Fig. 6i; see SI Sec. D.4 for a derivation of these scalings).

Vector-HaSH reproduces multiple additional aspects of hippocampal phenomenology

Vector-HaSH reproduces the results of memory consolidation experiments, as well as hippocampal place cell remapping (cf. Fig. 4) and the phenomenology of splitter cells[110–113].

The multiple traces theory of memory hypothesizes that episodic memory remains hippocampally dependent, and thus hippocampal damage should result in recall degradation but repeated presentation or recall of a memory should selectively reinforce it and make it more resistant to damage[114, 115]. We exposed Vector-HaSH to a set of inputs, out of which a fraction were presented or recalled multiple times. Each presentation or recall results in an further increment of the sensory-hippocampal weights, with the same learning rule. We found that memories reinforced in this way are remembered with richer detail relative to the rest, even after removal of a fraction of the hippocampal cells, Fig. 7a-b, SI Fig. S15. An alternative potential mechanism for consolidation could be that each repeated presentation or recall event activates a new scaffold state, adding associations between the same sensory input and multiple scaffold states. Our numerical results in Vector-HaSH did not support this mechanism, as associating the same input with two different scaffold states resulted in the activation of a third scaffold state when the partial input was presented for recall.

To model splitter cells (a fraction of hippocampal neurons whose firing in an unchanging environment varies based on context, e.g. varying with start and target locations, or for random foraging versus directed search)[110–113], we assumed that a sufficiently distinct context triggers re-initialization (remapping) across grid modules. A randomly selected set of initial grid phases is assigned to each context, and sensory-hippocampal-grid associations are built while traversing the environment in this context. When the agent returns to this context, the stored associations are recalled.

Under these assumptions, we simulate experiments in which splitter cells have been observed. Simulated grid and place fields on the shared stem of a Y maze are distinct based on the context of a right or a left run; the same is true for tree-shaped mazes, left- versus right- runs on a one-dimensional track, and clockwise versus counterclockwise runs on a closed path, Fig. 7c-f, SI Figs. S16,S17,S18. In the simulations, the ratio of splitter or route- or context-dependent cells to non-splitter cells was similar to the ratios seen in experiments, Fig. 7g, as were path and 'directionality indices' that quantified the degree with which context like the direction of the trajectory modified field locations[112,113], Fig. 7h-j. Our results indicate that we would expect similar directionality-dependence in the spatial tuning curves of grid cells, SI Fig. S19

Mechanism for the method of loci (memory palace) technique

An intriguing memory technique known for millennia, the memory palace or method of loci[116–119], is widely exploited by memory athletes in mnemonic competitions[120,121]. Given a list of typically non-spatial items to memorize, such as a list of names or a sequence of playing cards, memory athletes imagine a walk through a highly familiar and richly remembered space, such as one's childhood home or school. They attach the sequence of items to be memorized by mentally organizing and storing them near different locations and objects in a mental reimagining of the walk. Through unknown neural circuit mechanisms, this association enables highly accurate one-shot memorization and recall[122,123].

A simple extension of Vector-HaSH provides the first model for how memory palaces might work, helping to explain their power. Vector-HaSH is initialized to the start of a spatial trajectory in a highly familiar space. We assume that the neocortical representations of the sequence of items to be remembered is associated to the *recalled sensory states* in the familiar spatial trajectory. Crucially, even when these recalled trajectory sensory states are mere approximations of the actual sensory inputs (typical in the memory continuum), the approximate states are reliably the same: sensory reconstruction in Vector-HaSH is reliable even when approximate, Fig. 3b. Association with this reconstructed state allows the new patterns to themselves be highly accurately reconstructed even in the memory continuum of Vector-HaSH. The critical advantage of heteroassociation of new neocortical sequences with sensory recall states rather than hippocampal states as in episodic memory is that even in the memory continuum, the recall of new associated inputs is precise and detailed.

In sum, Vector-HaSH can explain the power and mechanisms of the memory palace technique: Associating arbitrary new input sequences with existing memory sequences (even when these are themselves imperfectly recalled) leads to one-shot high-capacity and high-precision memory for the new input sequences.

Discussion

Related models: Vector-HaSH is related to recent memory models[87,99], in that it shares the idea of a pre-structured exponential-capacity fixed-point scaffold for memory. However, Ref. 99 does not memorize externally specified patterns. Like Ref. 87, Vector-HaSH has a tripartite architecture in which external cues are heteroassociated with the scaffold. However, Ref. 87 does not include grid cell-like representations; as a consequence, the properties of strong generalization and high-capacity sequence memory from Vector-HaSH are absent in 87.

Though dense associative memories are capable of strong large numbers of patterns, their implementation is abstract in the form an gradient descent on energy landscapes; when implemented in a neural circuit with

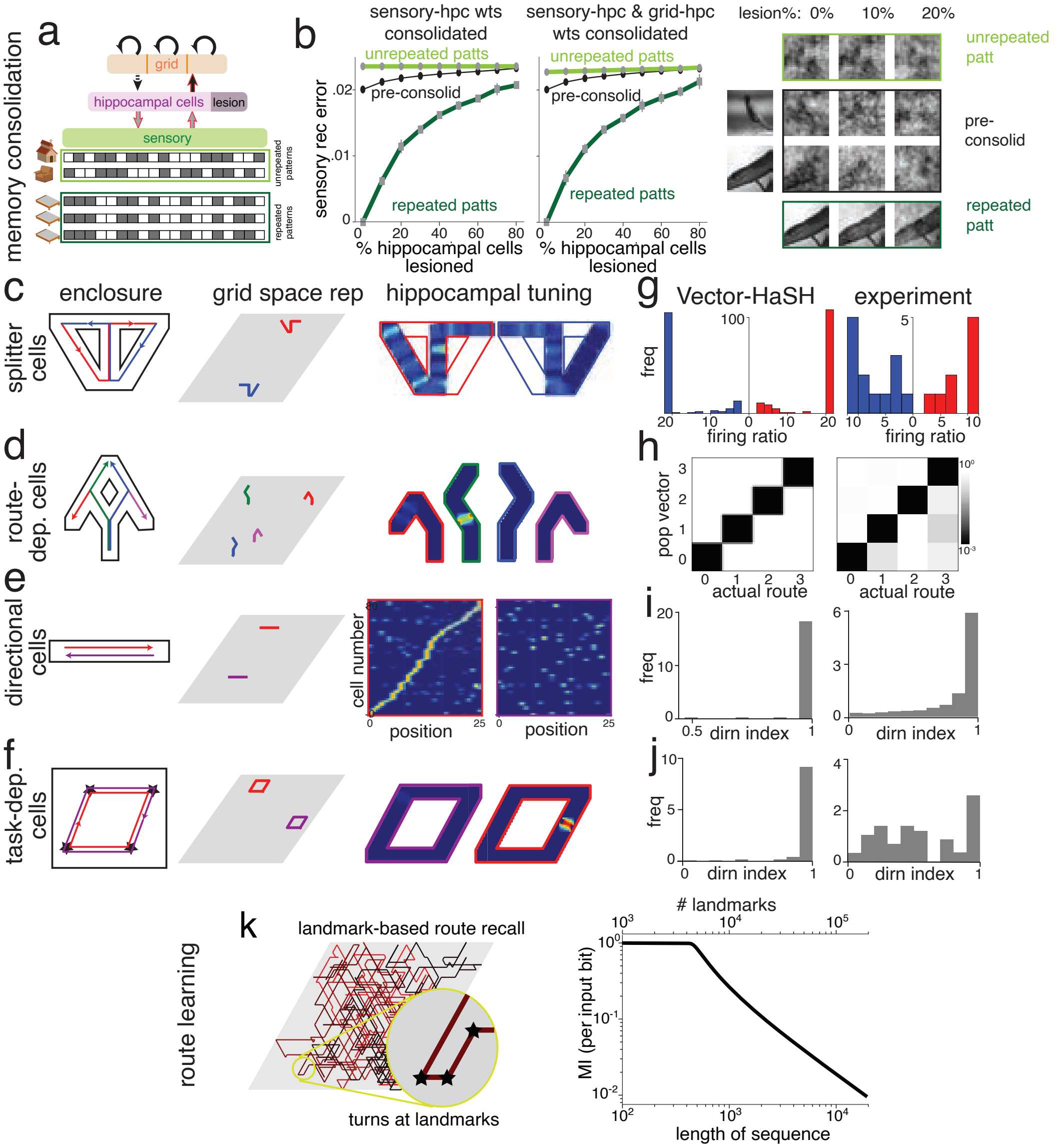

Figure 7. Vector-HaSH reproduces multiple aspects of hippocampal phenomenology. (a-b) Memory consolidation simulated in Vector-HaSH via the formation of multiple traces: repetition or recall of some input patterns drives an increment in the strength of the corresponding learned weights (red outlined arrows in (a)). (b) Repeated patterns are recovered, after hippocampal lesion, with smaller error than unrepeated patterns and than any of the patterns pre-consolidation. Recalled states are shown in SI Fig. S15. (c-f) Splitter cells[110], route-dependent spatial tuning[111], directional tuning[112], and task-dependent cells in an open arena[113] can all be modeled by Vector-HaSH if different trajectories traverse different regions of the grid coding space (grid cell remapping). (g-j) Quantification of directional/contextual/route selectivity of responses from the hippocampus in Vector-HaSH (left column) and experiments (right column), corresponding to (c-f). (g) Firing ratio distribution shows significantly different left- versus right-turn hippocampal tuning in the model and experiment[110] (see also SI Fig. S18). (h) Ensemble decoding of individual trajectories based on route population vectors, with color indicating the p-value of correct matches made by chance. (SI Sec.3.3 and SI Fig. S16)[111]). (i-j) The directionality index (a normalized metric for the difference in neural activity for different run directions[111, 112]) shows that a majority of hippocampal cells have directional fields[112] (Qualitatively similar results hold for a radial maze environment[113], SI Fig. S17.)

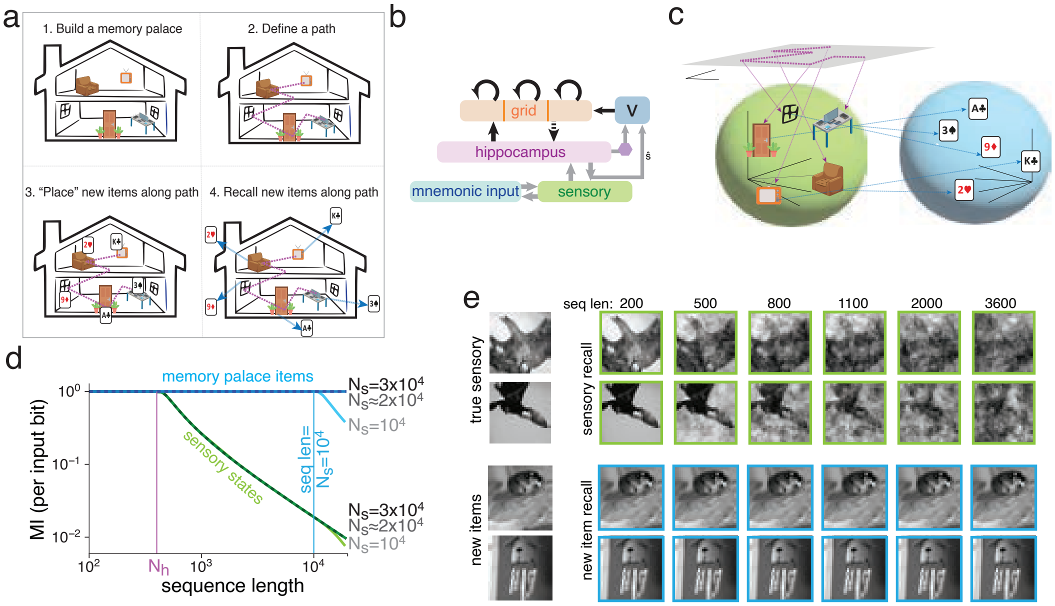

Figure 8. **Method of loci: Accurate recall of arbitrary inputs by heteroassociation onto landmarks in a "memory palace" formed by Vector-HaSH.** (a) The memory palace or Method of Loci technique for memory (figure adapted from[124]). (b) Model of the Method of loci: the full sequential Vector-HaSH circuit serves as a memory scaffold for new inputs, via heteroassociation with the sensory network. The path through the memory palace is learned via sequence learning, as in Fig. 6. The recollected sensory landmarks along the path are now (hetero)associated with the new items to be stored. (c) The large sensory area encoding spatial landmarks now forms the scaffold basis for heteroassociation of arbitrary sensory inputs in neocortex (cf. the smaller hippocampal basis for heteroassociation of sensory inputs within Vector-HaSH). (d-e) Scaffold states for neocortical input in the memory palace model are the (potentially imprecisely) recalled spatial landmarks. Because landmark recall is reliable even when imprecise (Fig. 3g, new items represented in neocortex can be recalled perfectly deep into the memory continuum even where landmark recall is substantially degraded relative to the true landmark, (d). Error bars (too small to be visible): standard deviation over 5 runs. (e) Two examples of the actual spatial landmarks, the recalled sensory states, and the recalled new neocortical inputs in the memory palace model, at different depths into the memory continuum. In (d-e), $\lambda = [3,4,5]$, $N_h = 400$.

pairwise synapses, they require exponentially many neurons (rather than the linearly many of Vector-HaSH) to store exponentially many patterns[76,125,126].

Vector-HaSH is similar to the hippocampal models of Refs. 127, 128, which also use fixed grid cell representations. In 127, hippocampal cells drive a low-dimensional update mechanism via grid cells, like our vector updating model. However, in both these models the scaffold is not invariant though the grid representations are fixed because the grid-hippocampal weights are learned. Therefore, they lack the properties of high capacity, large basins, and strong generalization. These models also do not memorize external inputs and are thus not memory models. A related memory model of the hippocampal complex is given in Ref. 72. This model memorizes external cues but like 127, 128 it lacks a fixed scaffold both because there is no fixed grid representation (recurrent interactions within the proto-grid network are (re)learned for each environment) and the grid-hippocampal weights are learned. Therefore, it lacks the high capacity and avoidance of catastrophic forgetting properties made possible by a fixed scaffold. It also lacks the low-dimensional shift mechanism learned from hippocampal or sensory states, and therefore the ability to store sequential memory at any level approaching that needed for episodic memory.

Given that the hippocampus is orders of magnitude smaller than the cortical states that represent events memorized by hippocampus, it is clear that any model of the hippocampus must involve state compression. In Vector-HaSH, the hippocampal representation is compact because the grid-hippocampal circuit functions as a content-independent *pointer* or hash mechanism for content in the cortex. A contrasting way to compress information is via content-based compression, as done by the bottleneck layer of an autoencoder (Fig. 3 and[129]). As we have seen here, direct content compression through learning is not highly performant: these models lack the capacity, resistance to catastrophic forgetting, and sequence memory properties of Vector-HaSH.

Nevertheless, the commonalities among many of these models point toward a converging view of the hippocampal complex. The highly performant features of Vector-HaSH suggest a first-level computational understanding of the circuit mechanisms of hippocampus as a highly performant memory system. Vector-HaSH is the first model we know of, besides MESH[130], that is capable of storing an exponentially large set of input patterns in an associative content-addressable memory.

Future extensions and directions: Vector-HaSH accounts for a range of phenomena in entorhinal cortex and hippocampus[100,110–115]. At the same time, there are numerous avenues for future research and extensions. These include incorporating different subregions of the hippocampus; relating the phenomena of different hippocampal LFP frequency peaks and the dominance of different inputs to the structure with the loop of dynamical updating and information flow in Vector-HaSH; investigating how the hippocampus and Vector-HaSH deal with conflicts between internal states and external cues and with changes in primary versus contextual inputs; enabling hierarchical and similarity-respecting representations for distinct but similar memories; exploring the dynamics of fragmentation of space and events into submaps and discrete episodes (e.g. via surprisal[102]) and map mergers; and understanding the contribution of different specific cell types and their roles in episodic memory within this circuit.

Relationship to anatomy: Vector-HaSH is based on the structure of hippocampal-entorhinal circuitry. However, in some respects it varies from the classic view: While hippocampal outputs to and inputs from entorhinal cortex are believed to be separated between deep versus superficial layers of entorhinal cortex, the scaffold involves a tight loop in which grid cells drive hippocampus and receive direct mirrored input back in a way that reinforces the input grid patterns. This structure is a prediction that the deep-to-superficial entorhinal projection closes a fully self-consistent loop, which can potentially be tested connectomically. Although much is known about entorhinal-hippocampal circuitry, new discoveries can still surprise: recent reports show that deep layers of EC, which receive hippocampal inputs and were believed to primarily send outputs to neocortex in fact send a copy of their outputs back to the hippocampus[131].

Random grid-to-hippocampal weights were key for several properties of Vector-HaSH, but are they a unique solution? Our theory and simulations show the sufficiency of random weights and the insufficiency of several types of non-random or learned weights. However, they do not eliminate the possibility of non-random solutions. The situation is similar to the construction of expander graphs for high-capacity error-correcting codes[132]: though non-random solutions can exist in principle, it has been hard to find them, while random connections are sufficient. Because Vector-HaSH is a full dynamical neural circuit that can be easily and directly queried for experimental predictions about representation, dynamics, and learning under a large variety of conditions and perturbations.

Summary: In sum, we have proposed a model that unifies the spatial and episodic memory roles of the hippocampal complex by showing that nominally spatial representations and architectures are critical for a well-behaved episodic memory, even if the memories are devoid of spatial content. Unlike unstructured recurrent memory models[76,84,86,93–98,125,126,133], the model factorizes the problem of associative memory into one of the creation of an abstract fixed point scaffold (for robust autoassociative recall of a massive number of abstract states and state sequences), with separate feedforward heteroassociation (abstract "pointers") to attach content to these abstract states. Mathematically, the model creates a hash code for each input pattern, which is given by the scaffold state, uses the pre-structured recurrent connectivity of the scaffold to be able to recreate those hash states as fixed points of the dynamics, and then uses the strongly full-rank property of the scaffold to enable (partial) decompression or reconstruction of the input patterns. In many senses, the model is consistent with past hints of the existence of abstract sequences that later become associated with newly explored spaces[66,134]. As a result, Vector-HaSH exhibits biologically realistic features of memory, such as avoiding the memory cliff of existing memory models. It maintains near-maximal use of its memory potential across a huge variation in numbers of stored memories, by gracefully trading off memory detail and number of items in memory. For sequence memory, the vectorized and metric nature of the grid cell states becomes critical: a very large combined sequence length capacity becomes possible when the only information required to specify the next state is a two-dimensional vector specifying direction and distance, rather than the full pattern of the next state.

Vector-HaSH provides a computational hypothesis for the mechanisms of the memory palace technique, based on understanding the advantages of co-localizing spatial and non-spatial memory. The model explains why impressive memory performance does not require exceptional intellectual ability or structural brain differences, but can be leveraged by anybody trained to appropriately engage the hippocampus[120,122,135]. From a neuro-AI perspective, the specific biological architectures, representations, and learning rules of Vector-HaSH led it to significantly outperform fully end-to-end supervised trainable memory models with similar architectures, comparable or more parameters, and fewer constraints – a realization of the hypothesis that biological structures (inductive biases) can produce better performance than fully end-to-end trained models as commonly used in machine learning.

Methods

In[87], the MESH associative memory architecture was introduced, leveraging a three-layer network to store numerous independent memory states. This architecture allowed for a high-capacity memory with a trade-off between the number of stored patterns and the fidelity of their recall. However, MESH did not require specifically grid cell encodings, did not exhibit strong generalization in scaffold learning, and did not exhibit a high sequence capacity.

In Vector-HaSH, our memory scaffold consists of a recurrent circuit incorporating MEC grid cells and a hippocampal layer that may be interpreted as the proximal CA1 and distal CA3 regions of the hippocampal complex. Specifically, we represent the MEC grid cells as outlined in Ref.[136], where each grid module's state is expressed using a one-hot encoded vector that represents the module's phase (and thus the active grid cell group within the module). The states are on a two-dimensional discretized hexagonal lattice with period λ. Thus, the state of each grid module is represented by a vector with a dimensionality of λ^2.

M such grid modules are concatenated together to form a collective grid state $g \in \{0,1\}^{N_g}$, where the $N_g = \sum_M \lambda_M^2$. The continuous attractor recurrence in the grid layer[33] is represented by a module-wise winner-take-all dynamics, which ensures that the equilibrium states of g always correspond to a valid grid-coding state.

$$g(t+1) = CAN[g(t)]. \tag{1}$$

We represent these equilibrium states by $g_{\vec{x}}$, where we index the coding states by the two-dimensional location $\vec{x}$. For coprime periods λ_M, the grid states can encode a spatial extent of $N_{patts} = \prod_M \lambda_M^2$ spatial locations.

This layer of grid cells projects randomly onto the hippocampal layer, through a $N_h \times N_g$ random matrix W_{hg}, with each element drawn independently from a Gaussian distribution with a mean of zero and standard-deviation one $N(0,1)$. This matrix is sparsified such that only a γ fraction of connections is retained, leading to a sparse random projection. This projection constructs an N_h dimensional set of hippocampal sparse states, $h_{\vec{x}}$ defined as

$$h_{\vec{x}} = \text{ReLU}[W_{hg} g_{\vec{x}} - \Theta]. \tag{2}$$

The return weights from the hippocampal layer back to the grid cell layer is set up through Hebbian learning between the predetermined set of grid and hippocampal states, $g_{\vec{x}}$ and $h_{\vec{x}}$.

$$W_{gh} = \frac{1}{N_h} \sum_{\vec{x}} g_{\vec{x}} h_{\vec{x}}^T. \tag{3}$$

The dynamics of the hippocampal scaffold is then set up as

$$\begin{align} g(t+1) &= CAN[W_{gh} h(t)] \tag{4}\\ h(t+2) &= \text{ReLU}[W_{hg} g(t) - \Theta] \tag{5}\\ & \tag{6} \end{align}$$

These equations maintain each $g_{\vec{x}}, h_{\vec{x}}$ state as a fixed point of the recurrent dynamics, as we prove in SI Sec. C.1.

This constructed hippocampal memory scaffold is then used to generate independent memory locations to store information presented through a sensory encoding layer, representing the non-grid cell component of the Entorhinal cortex. Information to be stored is presented as a binary encoding of states in the sensory layer, and is 'tagged' onto a memory location $\vec{x}$ of the scaffold through pseudo-inverse learned heteroassociative weights.

$$W_{hs} = HS^{+}, \text{ and,} \tag{7}$$
$$W_{sh} = SH^{+}, \tag{8}$$

where H is a $N_h \times N_{patts}$ dimensional matrix with columns as the predetermined hippocampal states $h_{\vec{x}}$, and S is a $N_s \times N_{patts}$ dimensional matrix with columns as the encoded sensory inputs to be stored at location $\vec{x}$. To reduce computational time-complexity, we use an exact pseudoinverse rather than an iterative pseudoinverse for calculation of these inter-layer weights, unless otherwise specified.

Given the above equations, we can now perform bi-directional inference of sensory inputs from grid states and vice versa:

$$h(t+1) = \text{ReLU}[W_{hs}s(t)] \tag{9}$$
$$g(t+1) = CAN[W_{gh}h(t+1)] \tag{10}$$

and

$$h(t+2) = \text{ReLU}[W_{hg}g(t) - \Theta] \tag{11}$$
$$s(t+2) = \text{sgn}[W_{sh}h(t+2)] \tag{12}$$

The above two sets of equations can then be combined to use Vector-HaSH as a content-addressable memory to recover stored sensory inputs from corrupted inputs — first the grid states are inferred from the corrupted sensory input, and then the true sensory input is recalled from the inferred grid state.

The above equations have been written considering sensory inputs to be random binary states. In cases where sensory states are continuous valued (as in Fig. 3b, for example) the s reconstruction equation, Eq. (12) is replaced with simply $s(t+2) = W_{sh}h(t+2)$.

Equations (1-12) describe the core working of Vector-HaSH— this core version and its variants can then be used to generate item memory, spatial memory, episodic memory, as well as a wide range of experimental observations, such as those discussed in Fig. 7.

1 High-capacity pattern reconstruction

For the basic task of pattern storage and reconstruction, we utilize the simplest form of Vector-HaSH without any additional components. To examine reconstruction capacity, N_{patts} sensory cues are stored in the network via training the W_{hs} and W_{sh} weights as described in Eqs. (7-8).

The N_{patts} sensory cues need to be stored corresponding to distinct scaffold states. In our implementation, for simplicity, we selected scaffold states in a "hairpin" like traversal, similar to that shown in Fig. 5a *top middle* to achieve this.

Then, a clean or corrupted version of a previously stored pattern is presented to the network in the sensory encoding layer, which then propagates through the network via Eqs. (9-12), finally generating the recalled pattern s.

In all numerical examples we consider in the main text we either construct random binary $\{-1,1\}$ patterns, or consider images from `mini-imagenet`[137]. In particular, we took 3600 images from the first 6 classes {'house-finch', 'robin', 'triceratops', 'green-mamba', 'harvestman', 'toucan'} and center-cropped them to consider the middle 60×60 image and converted them to grayscale. We refer to this set of grayscale images as `bw-mini-imagenet`. In all models, the memorized patterns are a noise-free set, then we test memory recall with noise-free, partial, or noisy cues.

In Figs. 2, 3, the recall performance and quality was examined in networks with three grid modules, $\gamma = 0.6$, and $\theta = 0.5$.

The capacity in Fig. 2c(*right*) was evaluated by injecting a noise into the hippocampal layer of magnitude 20% of the magnitude of the hippocampal state vector, and requiring the iterated dynamics to return the hippocampal state to within 0.6% of the original hippocampal state (Here magnitudes and distances were calculated via an L^2 metric).

In Fig. 2d(*left*) and SI Fig. S2, the critical N_h^* is estimated as the smallest value of N_h such that all scaffold states have been stabilized as fixed points. The corresponding module periods for data points plotted in SI Fig. S2, for two modules are listed in Table. 1 left, and for three modules are listed in Table. 1 right. Similarly, the grid module periods for the data in Fig. 2c(*left*) are listed in Table 2.

λ	N_g	N_{patts}
2,3	13	36
3,4	25	144
4,5	41	400
5,6	61	900
6,7	85	1764
7,8	113	3136

λ	N_g	N_{patts}
1,2,3	14	36
2,3,5	38	900
3,4,5	50	3600
4,5,7	90	19,600
5,6,7	110	44,100

Table 1. Grid module periods, number of grid cells and total number of patterns for data in Fig. 2e.

λ	N_g	N_{patts}
7,8	113	3136

λ	N_g	N_{patts}
3,5,8	98	14400

λ	N_g	N_{patts}
3,4,5,7	99	176400

λ	N_g	N_{patts}
1,3,4,5,7	100	176400

Table 2. Grid module periods, number of grid cells and total number of patterns for data in Fig. 2f.

To estimate the basin sizes of the patterns stored in the scaffold, as shown in Fig. 2e, we compute the probability that a given pattern is perfectly recovered (i.e., remains within its correct basin) as we perturb the hippocampal states with a vector of increasing magnitude. We assume that the size of any given basin can be estimated as the typical magnitude of perturbation that keeps the system within the same basin of attraction — this is not generally true for non-convex basins, particularly in high-dimensional spaces. However, this estimate is relevant in the context of testing robustness under corruption with uncorrelated noise. Further, we later demonstrate in SI Sec. C.3 that the basins are indeed convex. Here grid module periods $\lambda = [3,4,5]$, number of grid cells $N_g = 50$, and $N_h = 400$ hippocampal cells were used. Probability that a given pattern remains within its correct basin was estimated by computing the fraction of runs where a given pattern was correctly recovered for a 100 different random realization of the injected noise.

Figure 2f examines the learning generalization in Vector-HaSH, i.e., the capability of Vector-HaSH to self-generate fixed points corresponding to scaffold grid-hippocampal states despite training on a smaller number of fixed points. For a given number of training patterns, we calculate the number of generated fixed points by counting the number of states that when initialized at a scaffold state remain fixed upon iteration through Eqs. 4,5. As discussed in the main text, when training on a given number of training patterns (that is less than the complete set of all patterns), the ordering of the patterns is crucial in controlling the generalization properties of the model. For Vector-HaSH, we order patterns such that a two-dimensional contiguous region of space is covered (see Sec. C.4 for additional details of the ordering and the freedom of possibilities in this ordering), resulting in the strongest generalization (Sec. C.4). For comparison, in Fig. 2f we also consider "shuffled hippocampal states", wherein scaffold states are randomized in order before subsets are selected for training. We also consider "random hippocampal states": here we consider each hippocampal state vector and randomize its indices, in effect constructing a new state vector with exactly the same sparsity and statistics, but now uncorrelated to the grid state corresponding to that hippocampal state. Then, we use bi-directional pseudoinverse learning between grid and hippocampal states and construct this as a scaffold. This lack of structured correlations between grid and hippocampal population vectors results in catastrophic forgetting, with no observed fixed points remaining once all scaffold states have been used for training.

All curves shown in Fig. 3c-f are averaged over 5 runs with different random initialization of the predefined sparse connectivity matrix W_{hg}, error bars shown as shaded regions represent standard deviation across runs. In Figs. 3b,e,h, grid module periods $\lambda = [3,4,5]$, $N_g = 50$, $N_s = 3600$ was used. The total capacity of the network in this case is capped by $N_{patts} = \prod_M \lambda_M^2 = 3600$. In Fig. 3d, all shown networks have $\approx 5\times 10^5$ synapses, with Vector-HaSH module periods $\lambda = [2,3,5]$, and layer sizes: $N_g = 38$, $N_h = 275$, $N_s = 900$. Number of nodes in other networks are as follows: (i) Hopfield network of size $N = 708$, synapses = N^2. (ii) Pseudoinverse Hopfield network of size $N = 708$, synapses = N^2. (iii) Hopfield network with bounded synapses was trained with Hebbian learning on sequentially seen patterns. Size of the network $N = 708$, synapses = N^2. (iv) Sparse Hopfield network (with sparse inputs) with a network size of $N = 708$, synapses = N^2, sparsity = $100(1-p)$. (v) Sparse Hopfield network. Size of the network N, synapse dilution κ, synapses = $\kappa\times N^2 = 10^5$. (vi) Tailbiting Overparameterized Autoencoder[86] with network layer sizes 900, 275, 38, 275, 900.

For stored patterns of size N, recall of an independent random vector of size N would appear to have a mutual information of $\sim 1/\sqrt{N}$, which when evaluating the total mutual information across all $\mathcal{O}(N)$ patterns or more would appear to scale as $\mathcal{O}(\sqrt{N})$, despite no actual information being recalled. To prevent this apparent information recall, in Fig. 3f if the information recall is smaller than $1/\sqrt{N}$ we then set it explicitly to zero.

To examine Vector-HaSH's performance on patterns with correlations, in Fig. 3e we trained it on `bw-mini-imagenet` images using grid module sizes $\lambda = [3,4,5]$, and layer sizes: $N_g = 50$, $N_h = 400$, $N_s = 3600$. The plotted curve shows the mean-subtracted cosine similarity between recovered and stored patterns illustrating that Vector-HaSH shows gradual degradation as the number of stored patterns is increased. The resultant curve is an average over 5 runs with different sparse random projections W_{hg}.

In Fig. 7b, Vector-HaSH with grid module periods $\lambda = [3,4,5]$, $N_g = 50$, $N_h = 400$, $N_s = 3600$ was used, with random binary patterns in *left*,*middle*, and `bw-mini-imagenet` patterns in*right*. The results are averaged over 20 runs. Iterative online psuedoinverse learning[91] (with $\varepsilon = 0.01$) was used to reinforce a subset of patterns by adding addition traces to all the learnable weight matrices W_{hs}, W_{sh} and W_{gh}. W_{hs} reinforcement mathematically doesn't change W_{hs} due to iterative pseudoinverse being perfect from sensory to hippocampal layer. However, since W_{sh} leads to a gradual decay of reconstructed s states (i.e., iterative pseudoinverse learning from hippocampal layer to sensory cell layer is lossy), therefore reinforcement in W_{sh} changes these weights. Additional results from each layer of Vector-HaSH while testing the Multiple-Trace Theory are shown in Fig. S15, right. Further, Fig. S15, left shows the results when only W_{sh} weights are reinforced, assuming pre-trained scaffold weights W_{gh}. In both case, same parameter settings were used as in Fig. 7b.

1.1 Multiple Traces Theory

In Fig 7b, we consider Vector-HaSH with $\lambda = [3,4,5]$, $N_g = 50$, $N_h = 400$, $N_s = 3600$, $\gamma = 0.6$, and $\theta = 0.5$. The model was trained on 3600 images from `bw-mini-imagenet`. For sensory inputs presented multiple times, the sensory hippocampal weights are reinforced multiple times using online pseudoinverse learning rule[91] (Fig 7b), and the grid hippocampal weights are reinforced multiple times using Hebbian learning (Fig 7b, right). Given a particular lesion size, the cells to be lesioned are randomly chosen from the set of all hippocampal cells, and their activation is set to zero. Sensory recovery error is defined as the mean L2-norm between the ground truth image and the image reconstructed by the model. During testing, the model receives the ground truth sensory image as input, and the reconstruction dynamics are as follows:

$$h(t+1) = \mathrm{ReLU}[W_{hs}s(t)] \tag{13}$$
$$g(t+1) = CAN[W_{gh}h(t+1)] \tag{14}$$
$$h(t+2) = \mathrm{ReLU}[W_{hg}g(t+1) - \Theta] \tag{15}$$
$$s(t+3) = \mathrm{sgn}[W_{sh}h(t+2)]. \tag{16}$$

2 Mapping, recall, and zero-shot inference in multiple spatial environments without catastrophic interference.

Here we add a path-integration component to Vector-HaSH, that utilizes a velocity input to change the grid cell population activity akin to Ref.[33], such that the phase represented by each module changes in correspondence to the velocity input. Corresponding to the discrete hexagonal lattice space used to represent each grid module, for simplicity the velocity is assumed to have one of six directions, and magnitude is assumed to be fixed at a constant such that the phase of each grid module updates by a single lattice point in a single timestep. This input velocity vector, that we call a velocity shift operator, $\vec{v}$, is thus represented by a six-dimensional one-hot encoded vector determining the direction of the shift.

In order to capture the inherent randomness and uncertainty present in real-world scenarios, a small amount of neuronal noise was introduced by adding random perturbations to the activation values of hippocampal cells in Vector-HaSH. This noise, generated from a uniform distribution between 0 and 0.1, mimics the fluctuations and disturbances observed in individual neurons, and corresponds to a noise magnitude of roughly 25% the magnitude of the hippocampal state vectors.

In Fig. 4a,c we first demonstrate bi-directional recall of grid states from sensory inputs and vice versa. Here we consider Vector-HaSH with $\lambda = [3,4,5]$, $N_g = 50$, $N_h = 400$, $N_s = 3600$. We train the model on a total of 600 sensory inputs taken from `bw-mini-imagenet` (including the 4 landmarks placed in the room shown in Fig. 4c). To demonstrate zero-shot recall in panel c, the model dynamics are simulated on a novel trajectory (right) through the same room with some locations overlapping with the previous trajectory. Note that the reconstructed landmarks do not have perfect recall. Instead, the reconstructions are degraded relative to the originally stored landmarks since the total number of stored landmarks in the model exceeds $N_h = 400$ (Fig. 2f).

For all other panels of Figure 4, we use Vector-HaSH with grid module periods $\lambda = [3,4,5,7]$, $N_g = 99$, $N_h = 342$, $\gamma = 0.1$, and $\theta = 2$. The total capacity of this grid coding space is $176400 \approx 2 \times 10^5$. Each room is stored by allocating a random 10×10 patch of the grid coding space to it (This is constructed by first choosing any random point in the room to map to a randomly chosen area of the grid coding space. Then as the model moves in the room, path integration correspondingly updates the grid phases in each grid module. The region of grid coding space explored as the model physically explores a room is then the patch of grid coding space storing the particular room).

To each of the 100 locations comprising a room, we simulate an independent sensory landmark as a binary $\{-1,1\}$ vectors. At initialization, before observing any room, we begin with a pretrained memory scaffold, wherein the W_{hg} and W_{gh} matrices have already been constructed and trained corresponding to Eqs. 2, 3.

When first brought to a room, the grid state is initialized to the grid state vector corresponding to the random region of grid coding space allocated to the room. Then, as path integration updates the grid state upon moving around the room, the observed sensory landmark states are associated with the corresponding grid-hippocampal scaffold states through learning the W_{hs} and W_{sh} matrices following Eqs. 7, 8.

In the first two tests of each room (first tested right after each room has been learned, and then tested after all rooms have been learned; shown in Fig. 4d) sensory landmark cues can be observed by Vector-HaSH. Using Eq. 9, the observed sensory landmarks can be used to reconstruct the hippocampal state, resulting in the reliably reconstructed hippocampal tuning curves as seen in Fig. 4e. For testing stable recall in dark (Fig. 4d,e), Vector-HaSH is provided a random single sensory landmark cue from any given room. This landmark is used to ascertain the grid state corresponding to that landmark through Eq. 9. Thereafter, path integration is used to construct the grid-hippocampal scaffold state as room is explored in the absence of any further sensory cues. As seen in Fig. 4e this also reliably reconstructs the hippocampal state at each location in every room.

In Fig. 4f, we examine the dark recall of 3600-dimensional sensory landmarks in each room in a continual learning setting. Here we begin again with simply the pretrained grid-hippocampal scaffold. As the i^{th} room is explored, the sensory-hippocampal weight matrices are updated to store the thus far observed landmarks and their locations. At each step of exploration within the i^{th} room, Vector-HaSH is queried on the current and all previous rooms in the following fashion: for any completed room j (i.e., $0 \leq j < i$), Vector-HaSH is dropped randomly anywhere in the room and allowed to observe the sensory landmark solely at that start location and no further sensory landmarks. Then the model moves around the room through path integration, and attempts to predict the sensory landmarks that would be observed at each location. We then compute the average mutual information recovered for each landmark at each position in the room, which is shown in Fig. 4f. For the partially completed room *i*, Vector-HaSH is similarly dropped randomly in the room, restricted to the set of previously observed locations within the room. The mutual information recovered during sensory prediction is similarly only evaluated over the previously observed portion of the room.

For the baseline model shown in Fig. 4f, we first construct the grid-hippocampal network through random hippocampal states with the same sparsity as those in Vector-HaSH, and bi-directional pseudoinverse learning between grid and hippocampal layers. Thereafter, the sensory landmarks are associated with the hippocampal layer as in Vector-HaSH described above, and this baseline model is subjected to an identical test protocol to examine continual learning. The number of nodes in the baseline model is kept identical to Vector-HaSH.

For Fig. 4h, we follow the same analysis as in the experiment[100]. Dot product between population vectors (PVs) across all combinations of the eleven test rooms were computed. To construct the population vectors, we record the activations of hippocampal cells for each of the 10×10 positions in the simulated room. We stack these into 100 composite population vectors (PVs), one for each position in the room. To compute overlaps between representations, the activation of each hippocampal cell in any particular room was expressed as a ratio of its activation to the maximal activation of that cell across all rooms. The overlap was then calculated as the normalized dot product between the hippocampal cell activation vectors in two rooms i.e., the sum of the products of corresponding components divided by the total number of hippocampal cells ($N_h = 342$) for a given position/pixel, averaged over 100 positions. The color-coded matrix in Fig. 4h shows the average dot product values for PVs across rooms ($\binom{11}{2} = 55$ room pairs). Repeated exposures to three familiar rooms were also added to this analysis leading to a total of $\binom{14}{2} = 91$ room pairs.

For Fig. 4j, we plot the distribution of PV normalized dot products computed above (for multiple visits to all the rooms) and use this PDF to compute the corresponding CDF. Similarly, the CDFs for shuffled data are computed through the same procedure, but using shuffled data to compute the PV normalized dot products. Shuffled data is obtained either by random assignment of rate maps across rooms (shuffle room) or by shuffling of cell identities within rooms (shuffle cells) or by a combination of the two procedures (shuffle room and cells). The number of different shuffles generated in each case was 1000.

3 Path learning in the hippocampal scaffold

Here again, we add a path-integration component to Vector-HaSH as described in the section above, such that a velocity shift operator, $\vec{v}$, can be used to path integrate and update the grid cell population activity akin to Ref.[33], such that the phase represented by each module changes in correspondence to the input shift.

For learning of trajectories in space, this vector $\vec{v}$ is either associated with spatial locations and corresponding hippocampal state vectors (as in path learning) or with sensory landmark inputs (as in route learning).

The results in Fig. 5 and Fig. 6 were generated using $N_h = 500$, $\gamma = 0.6$, $\theta = 0.5$, $M = 3$ and $N_s = \prod_M \lambda_M^2$ with $\lambda = [5, 9, 13]$ in Fig. 5b-e, Fig. 6b and $\lambda = [3, 4, 5]$ in Fig. 5f,g, Fig. 6c.

All networks in Fig. 6e were constructed to have approximately 5×10^5 synapses, with network parameters identical to those in Fig. 3d. All panels in Fig. 6 considers random binary patterns, apart from Fig. 6c,d which considers `bw-mini-imagenet` images.

3.1 Path learning

Learning associations from the hippocampal layer directly to the velocity inputs through pseudoinverse learning would result in perfect recall for only $N_{seq} \leq N_h$, which may be much smaller than the grid coding space, and would hence result in an incapability to recall very long sequences. To obtain higher capacity, we learn a map from the hippocampal cell state to the corresponding velocity inputs at that spatial location through a multi-layer perceptron, MLP. For all the results shown in Fig. 5 and Fig. 6c,h, we use a single hidden layer in the MLP with 250 nodes.

The dynamics of the network are as follows:

$$\vec{v}(t) = \mathrm{MLP}[h(t)] \tag{17}$$

$$g(t+1) = \mathrm{PI}[g(t);\vec{v}(t)] \tag{18}$$

$$h(t+1) = \mathrm{ReLU}[W_{hg}g(t+1) - \Theta] \tag{19}$$

$$s(t+1) = \mathrm{sgn}[W_{sh}h(t+1)] \tag{20}$$

$$\tag{21}$$

3.2 Route learning

Since detailed sensory information cannot be recalled at very high capacities, route learning is performed by learning associations between the *recollection* of the sensory inputs at a location $\vec{x}$, and the velocity shift vector $\vec{v}$ determining the direction of motion of the trajectory being learned at that location. This association can be learned directly through pseudoinverse learning as

$$W_{vs} = VS_r^+, \tag{22}$$

where, S_r is a $N_s \times N_{seq}$ dimensional matrix with columns as the recalled sensory inputs $s_{\vec{x}}$, and V is a $6 \times N_{seq}$ dimensional matrix with columns as the corresponding velocities. These associations can then be used to recall long trajectories through

$$\vec{v}(t) = \mathrm{WTA}[W_{vs}s(t)] \tag{23}$$

$$g(t+1) = \mathrm{PI}[g(t);\vec{v}(t)] \tag{24}$$

$$h(t+1) = \mathrm{ReLU}[W_{hg}g(t+1) - \Theta]s(t+2) = \mathrm{sgn}[W_{sh}h(t+1)] \tag{25}$$

$$\tag{26}$$

As argued in Sec. D.6, this results in perfect sequence recall for $N_{seq} \leq N_s$, which can scale as the exponentially large capacity of the grid coding space.

3.3 Goal and context-based remapping

When initialized in a new environment, we model the grid state population activity to be randomly initialized in the grid-coding space (a mechanistic model for such random initialization will be discussed in future work), i.e., the grid state undergoes *remapping*. This grid coding state, along with the corresponding hippocampal coding state and sensory observations at that location are then stored in the corresponding weight matrices, i.e., W_{hs} and W_{sh}, via Eqs. (7-8). When brought back to a previously seen environment, these weight matrices in Vector-HaSH use the observed sensory observations to drive the hippocampal cell (and hence grid cell) population activity to the state initialized at the first traversal of that environment.

Similar to new environments, we also model contextual information (such as goals, rewards, start-end location pairs) to be appended to the sensory inputs. We allow new contextual information to also trigger reinitialization of grid state, which then permits storage of multiple paths that involve the same spatial location, provided that they are distinguished by a contextual signal.

We use this set up of manual reinitialization of the grid state to reproduce the experimental observations of splitter cells[110], route dependent place cells[111], directional place fields in one-dimensional environments[112] and on directed routes in two-dimensional environments[113] in Fig. 7c-j; and of directional place fields in a radial eight-arm maze[113] in Fig. S17. In all of these cases, we first generate trajectories corresponding to the paths that the animals are constrained to traverse in the given experiment. These trajectories, are then stored in Vector-HaSH at a random location in the grid coding space through a path learning mechanism. At new contexts, the grid state in the model is reinitialized and the agent then continues at a new location in the grid coding space. This results in different spatial firing fields, irrespective of whether the agent is at the same spatial location as in a different previous context.

For all the simulations in Fig. 7c-j and Fig. S17, Vector-HaSH with $\lambda = [3,4,5,7]$, $N_h = 500$, $N_g = 99$, $\theta = 2.0$ and $\gamma = 0.10$ was used. The total size of the grid coding space is $420 \times 420 \approx 10^5$. In order to capture the inherent randomness and uncertainty present in real-world scenarios, a small amount of neuronal noise was introduced by adding random perturbations to the activation values of hippocampal cells in Vector-HaSH. This noise, generated from a uniform distribution between 0 and 0.1, mimics the fluctuations and disturbances observed in individual neurons.

Splitter cells : For Fig. 7c,g, we follow an analysis method similar to the analysis done on the experimental data[110]. The central stem is divided into 4 equal regions (Fig. S18b), and the mean activation of every hippocampal cell is computed in each of the four regions. Figure S18c plots mean activations in each of the four regions, of cells that show different activity patterns as Vector-HaSH traverses the central stem on Left-Turn and Right-Turn trials. The "activation ratio" on Right-Turn trials versus Left-Turn trials is then calculated for each cell in the region for which the given cell has maximum difference in activations. The distribution of these activation ratios is plotted in Fig. 7g, that shows the frequency distribution of cells with preferential firing associated with Left-Turn or Right-Turn trials. Note that the distribution of cells preferring left-turn and right-turn trials is approximately even. The percentage of hippocampal cells with non-differential firing was found to be $\approx 3.896\%$, and the percentage of hippocampal cells with differential firing was found to be $\approx 96.103\%$ in Vector-HaSH (using a threshold of 2 on the activation ratio).

Route encoding : In Fig. 7d,h we employed an ensemble analysis approach mirroring that used in[111] to validate if hippocampal cells demonstrate route-dependent activity. Our simulated session comprised four blocks, each representing one of four routes (0-3), with 11 trials per block. We performed ensemble analysis on the maze region common to all routes.

We compared the population vector (PV)—activations of all hippocampal cells on an individual trajectory—to the average activation of these cells across all trajectories on each route (route-PV). Specifically, we compared the PVs for each trajectory to the average activation population vectors (route-PVs) of all four routes, excluding the trajectory in consideration from its route-PV calculation to avoid bias.

Using cosine similarity, we assessed the likeness between each trajectory PV and each of the four route PVs. We then calculated the fraction of correct matches (the highest similarity score was with its corresponding route-PV) and incorrect matches (a higher similarity score was with a different route-PV). The comparison results are shown in Fig. S16a, left.

We repeated the process 10,000 times with randomized data to estimate the chance probability of correct matches. We randomized the session data by shuffling trials across blocks, randomly assigning each trajectory to one of the four routes, thereby disrupting any correlation between the hippocampal cell activations and a specific route. Fig. S16a,right depicts a typical result from one such shuffle.

For each matrix element (i, j), we plotted the distribution of data from these 10,000 matrices in Fig. S16b. We then estimated the Probability Density Function (PDF) from this distribution using a Gaussian kernel (Python's `scipy.stats.gaussian_kde` method). To gauge the chance probability of correct matches in our original, unshuffled analysis, we calculated the percentile position of our observed match proportion, referencing the same matrix element (i, j) from the unshuffled matrix in Fig. S16a.

Fig. 7h presents the probability of correct matches in the unshuffled analysis based on these distributions from 10,000 shuffles. Low diagonal values indicate that trajectories significantly match only their corresponding route-PVs.

Directional cells : For Figs. 7i,j and Fig. S17, the directionality index is defined similar to that defined for the experimental data analysis[112],[111]. Given the activation (A) of a hippocampal cell in positive and negative running directions (A_+ and A_-), we define the directionality index as $|A_+ - A_-| \,/\, |A_+ + A_-|$. By this definition, a directionality index of one indicates activity in one direction only, and a directionality index of zero indicates identical activity in both directions.

We use the same definition of directionality index to compute the directionality of the grid cells in Vector-HaSH, shown in Fig. S19.

References

1. Proust, M. *À la recherche du temps perdu* (1913).

2. Milner, B. The medial temporal-lobe amnesic syndrome. *Psychiatr. Clin. North Am.* **28**, 599–611, 609 (2005).

3. Reed, J. M. & Squire, L. R. Impaired recognition memory in patients with lesions limited to the hippocampal formation. *Behav. neuroscience* **111**, 667 (1997).

4. Zola, S. M. *et al.* Impaired recognition memory in monkeys after damage limited to the hippocampal region. *J. Neurosci.* **20**, 451–463 (2000).

5. Manns, J. R., Hopkins, R. O., Reed, J. M., Kitchener, E. G. & Squire, L. R. Recognition memory and the human hippocampus. *Neuron* **37**, 171–180 (2003).

6. O'Keefe, J. & Dostrovsky, J. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research* (1971).

7. Nadel, L. & O'Keefe, J. The hippocampus as a cognitive map (1978).

8. Taube, J. S., Muller, R. U. & Ranck, J. B. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *J. Neurosci.* **10**, 420–435 (1990).

9. Wilson, M. A. & McNaughton, B. L. Dynamics of the hippocampal ensemble code for space. *Science* **261**, 1055–1058 (1993).

10. Wilson, M. A. & McNaughton, B. L. Reactivation of hippocampal ensemble memories during sleep. *Science* **265**, 676–679 (1994).

11. Lee, A. K. & Wilson, M. A. Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron* **36**, 1183–1194 (2002).

12. Hyman, J. M., Wyble, B. P., Goyal, V., Rossi, C. A. & Hasselmo, M. E. Stimulation in hippocampal region ca1 in behaving rats yields long-term potentiation when delivered to the peak of theta and long-term depression when delivered to the trough. *J. Neurosci.* **23**, 11725–11731 (2003).

13. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **436**, 801–806 (2005).

14. Ji, D. & Wilson, M. A. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nat. neuroscience* **10**, 100–107 (2007).

15. Hassabis, D., Kumaran, D., Vann, S. D. & Maguire, E. A. Patients with hippocampal amnesia cannot imagine new experiences. *Proc. Natl. Acad. Sci.* **104**, 1726–1731 (2007).

16. Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B. & Moser, E. I. Representation of geometric borders in the entorhinal cortex. *Science* **322**, 1865–1868 (2008).

17. Fenton, A. A. *et al.* Unmasking the ca1 ensemble place code by exposures to small and large environments: more place cells and multiple, irregularly arranged, and expanded place fields in the larger space. *J. Neurosci.* **28**, 11250–11262 (2008).

18. Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.* **31**, 69–89 (2008).

19. Lever, C., Burton, S., Jeewajee, A., O'Keefe, J. & Burgess, N. Boundary vector cells in the subiculum of the hippocampal formation. *J. Neurosci.* **29**, 9771–9777 (2009).

20. Colgin, L. L. *et al.* Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature* **462**, 353–357 (2009).

21. Stensola, H. *et al.* The entorhinal grid map is discretized. *Nature* **492**, 72–78 (2012).

22. Bonnevie, T. *et al.* Grid cells require excitatory drive from the hippocampus. *Nat. neuroscience* **16**, 309–317 (2013).

23. Buzsáki, G. & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nat. neuroscience* **16**, 130–138 (2013).

24. Kraus, B. J., Robinson, R. J., White, J. A., Eichenbaum, H. & Hasselmo, M. E. Hippocampal "time cells": time versus path integration. *Neuron* **78**, 1090–1101 (2013).

25. Krupic, J., Bauza, M., Burton, S., Barry, C. & O'Keefe, J. Grid cell symmetry is shaped by environmental geometry. *Nature* **518**, 232–235 (2015).

26. Høydal, Ø. A., Skytøen, E. R., Andersson, S. O., Moser, M.-B. & Moser, E. I. Object-vector coding in the medial entorhinal cortex. *Nature* **568**, 400–404 (2019).

27. Marr, D., Willshaw, D. & McNaughton, B. *Simple memory: a theory for archicortex* (Springer, 1991).

28. Skaggs, W., Knierim, J., Kudrimoti, H. & McNaughton, B. A model of the neural basis of the rat's sense of direction. *Adv. neural information processing systems* **7** (1994).

29. Burgess, N., Recce, M. & O'Keefe, J. A model of hippocampal function. *Neural networks* **7**, 1065–1081 (1994).

30. McClelland, J. L., McNaughton, B. L. & O'Reilly, R. C. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychol. review* **102**, 419 (1995).

31. Blum, K. I. & Abbott, L. F. A model of spatial map formation in the hippocampus of the rat. *Neural computation* **8**, 85–93 (1996).

32. Hartley, T., Burgess, N., Lever, C., Cacucci, F. & O'keefe, J. Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus* **10**, 369–379 (2000).

33. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology* **5**, e1000291 (2009).

34. Yoon, K. *et al.* Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat. neuroscience* **16**, 1077–1084 (2013).

35. Trettel, S. G., Trimper, J. B., Hwaun, E., Fiete, I. R. & Colgin, L. L. Grid cell co-activity patterns during sleep reflect spatial overlap of grid fields during active behaviors. *Nat. Neurosci.* **22**, 609–617 (2019).

36. Gardner, R. J. *et al.* Toroidal topology of population activity in grid cells. *Nature* **602**, 123–128 (2022).

37. Derdikman, D., Knierim, J. J. *et al.* Space, time and memory in the hippocampal formation. Tech. Rep., Springer.

38. O'Reilly, R. C., Bhattacharyya, R., Howard, M. D. & Ketz, N. Complementary learning systems. *Cogn. science* **38**, 1229–1248 (2014).

39. Bush, D., Barry, C., Manson, D. & Burgess, N. Using grid cells for navigation. *Neuron* **87**, 507–520 (2015).

40. Hardcastle, K., Ganguli, S. & Giocomo, L. M. Environmental boundaries as an error correction mechanism for grid cells. *Neuron* **86**, 827–839 (2015).

41. Moser, M.-B., Rowland, D. C. & Moser, E. I. Place cells, grid cells, and memory. *Cold Spring Harb. perspectives biology* **7**, a021808 (2015).

42. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *Elife* **5**, e10094 (2016).

43. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. neuroscience* **20**, 1643–1653 (2017).

44. Eichenbaum, H. On the integration of space, time, and memory. *Neuron* **95**, 1007–1018 (2017).

45. Keinath, A. T., Epstein, R. A. & Balasubramanian, V. Environmental deformations dynamically shift the grid cell spatial metric. *Elife* **7**, e38169 (2018).

46. Behrens, T. E. *et al.* What is a cognitive map? organizing knowledge for flexible behavior. *Neuron* **100**, 490–509 (2018).

47. Banino, A. *et al.* Vector-based navigation using grid-like representations in artificial agents. *Nature* **557**, 429–433 (2018).

48. Ocko, S. A., Hardcastle, K., Giocomo, L. M. & Ganguli, S. Emergent elasticity in the neural code for space. *Proc. Natl. Acad. Sci.* **115**, E11798–E11806 (2018).

49. Chaudhuri, R., Gerçek, B., Pandey, B., Peyrache, A. & Fiete, I. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. neuroscience* **22**, 1512–1520 (2019).

50. Agmon, H. & Burak, Y. 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* **9**, e56894 (2020).

51. Whittington, J. C. *et al.* The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263 (2020).

52. Uria, B. *et al.* The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains. *BioRxiv* 2020–11 (2020).

53. Sanders, H., Wilson, M. A. & Gershman, S. J. Hippocampal remapping as hidden state inference. *Elife* **9**, e51140 (2020).

54. George, D. *et al.* Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nat. communications* **12**, 2392 (2021).

55. Whittington, J. C., McCaffary, D., Bakermans, J. J. & Behrens, T. E. How to build a cognitive map. *Nat. neuroscience* **25**, 1257–1272 (2022).

56. Scoville, W. B. & Milner, B. Loss of recent memory after bilateral hippocampal lesions. *J. neurology, neurosurgery, psychiatry* **20**, 11 (1957).

57. Penfield, W. & Milner, B. Memory deficit produced by bilateral lesions in the hippocampal zone. *AMA archives Neurol. & Psychiatry* **79**, 475–497 (1958).

58. Milner, B., Corkin, S. & Teuber, H.-L. Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of hm. *Neuropsychologia* **6**, 215–234 (1968).

59. O'Keefe, J. & Dostrovsky, J. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **34**, 171–175 (1971).

60. Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.* **31**, 69–89 (2008).

61. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **436**, 801–806 (2005).

62. Yoon, K. *et al.* Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat. Neurosci.* **16**, 1077–1084 (2013).

63. Trettel, S. G., Trimper, J. B., Hwaun, E., Fiete, I. R. & Colgin, L. L. Grid cell co-activity patterns during sleep reflect spatial overlap of grid fields during active behaviors. *Nat. Neurosci.* **22**, 609–617 (2019).

64. Gardner, R. J., Lu, L., Wernle, T., Moser, M.-B. & Moser, E. I. Correlation structure of grid cells is preserved during sleep. *Nat. Neurosci.* **22**, 598–608 (2019).

65. O'Keefe, J. & Nadel, L. *The hippocampus as a cognitive map* (Oxford University Press, Oxford, 1978).

66. Buzsáki, G. & Tingley, D. Space and time: the hippocampus as a sequence generator. *Trends cognitive sciences* **22**, 853–869 (2018).

67. Aronov, D., Nevers, R. & Tank, D. W. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature* **543**, 719–722 (2017).

68. Killian, N. J., Potter, S. M. & Buffalo, E. A. Saccade direction encoding in the primate entorhinal cortex during visual exploration. *Proc. Natl. Acad. Sci.* **112**, 15743–15748 (2015).

69. Constantinescu, A. O., O'Reilly, J. X. & Behrens, T. E. J. Organizing conceptual knowledge in humans with a gridlike code. *Science* **352**, 1464–1468 (2016).

70. Eichenbaum, H. Is the rodent hippocampus just for 'place'? *Curr. Opin. Neurobiol.* **6**, 187–195 (1996).

71. Eichenbaum, H. What versus where: Non-spatial aspects of memory representation by the hippocampus. *Curr. Top. Behav. Neurosci.* **37**, 101–117 (2018).

72. Whittington, J. C. R. *et al.* The Tolman-Eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263.e23 (2020).

73. Neupane, S., Fiete, I. R. & Jazayeri, M. Vector production via mental navigation in the entorhinal cortex. *bioRxiv* 2022–12 (2022).

74. Gardner, E. The space of interactions in neural network models. *J. Phys. A Math. Gen.* **21**, 257–270 (1988).

75. Abu-Mostafa, Y. S. & St Jacques, J. Information capacity of the hopfield model. *IEEE Trans. Inf. Theory* **31**, 461–464 (1985).

76. Krotov, D. & Hopfield, J. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996* (2020).

77. Witter, M. P. & Groenewegen, H. J. Laminar origin and septotemporal distribution of entorhinal and perirhinal projections to the hippocampus in the cat. *J. Comp. Neurol.* **224**, 371–385 (1984).

78. Amaral, D. G. & Witter, M. P. The three-dimensional organization of the hippocampal formation: a review of anatomical data. *Neuroscience* **31**, 571–591 (1989).

79. Witter, M. P. & Amaral, D. G. Entorhinal cortex of the monkey: V. projections to the dentate gyrus, hippocampus, and subicular complex. *J. Comp. Neurol.* **307**, 437–459 (1991).

80. Witter, M. P., Doan, T. P., Jacobsen, B., Nilssen, E. S. & Ohara, S. Architecture of the entorhinal cortex a review of entorhinal anatomy in rodents with some comparative notes. *Front. Syst. Neurosci.* **11**, 46 (2017).

81. Stensola, H. *et al.* The entorhinal grid map is discretized. *Nature* **492**, 72–78 (2012).

82. Fiete, I. R., Burak, Y. & Brookings, T. What grid cells convey about rat location. *J. Neurosci.* **28**, 6858–6871 (2008).

83. Sreenivasan, S. & Fiete, I. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nat. Neurosci.* **14**, 1330–1337 (2011).

84. Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U. S. A.* **81**, 3088–3092 (1984).

85. Brun, V. H. *et al.* Place cells and place recognition maintained by direct entorhinal-hippocampal circuitry. *Science* **296**, 2243–2246 (2002).

86. Radhakrishnan, A., Belkin, M. & Uhler, C. Overparameterized neural networks implement associative memory. *Proc. Natl. Acad. Sci.* **117**, 27162–27170 (2020).

87. Sharma, S., Chandra, S. & Fiete, I. Content addressable memory without catastrophic forgetting by heteroassociation with a fixed scaffold. In *International Conference on Machine Learning*, 19658–19682 (PMLR, 2022).

88. Rajasethupathy, P. *et al.* Projections from neocortex mediate top-down control of memory retrieval. *Nature* **526**, 653–659 (2015).

89. Eichenbaum, H. Prefrontal–hippocampal interactions in episodic memory. *Nat. Rev. Neurosci.* **18**, 547–558 (2017).

90. Simons, J. S. & Spiers, H. J. Prefrontal and medial temporal lobe interactions in long-term memory. *Nat. reviews neuroscience* **4**, 637–648 (2003).

91. Tapson, J. & van Schaik, A. Learning the pseudoinverse solution to network weights. *Neural Networks* **45**, 94–100 (2013).

92. O'Reilly, R. C. Six principles for biologically based computational models of cortical cognition. *Trends cognitive sciences* **2**, 455–462 (1998).

93. Personnaz, L., Guyon, I. & Dreyfus, G. Information storage and retrieval in spin-glass like neural networks. *J. de Physique Lettres* **46**, 359–365 (1985).

94. Personnaz, L., Guyon, I. & Dreyfus, G. Collective computational properties of neural networks: New learning mechanisms. *Phys. Rev. A* **34**, 4217 (1986).

95. Parisi, G. A memory which forgets. *J. Phys. A: Math. Gen.* **19**, L617 (1986).

96. Fusi, S. & Abbott, L. Limits on the memory storage capacity of bounded synapses. *Nat. neuroscience* **10**, 485–493 (2007).

97. Tsodyks, M. V. & Feigel'man, M. V. The enhanced storage capacity in neural networks with low activity level. *EPL (Europhysics Lett.* **6**, 101 (1988).

98. Dominguez, D., Koroutchev, K., Serrano, E. & Rodríguez, F. B. Information and topology in attractor neural networks. *Neural computation* **19**, 956–973 (2007).

99. Chaudhuri, R. & Fiete, I. Bipartite expander hopfield networks as self-decoding high-capacity error correcting codes. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (2019).

100. Alme, C. B. *et al.* Place cells in the hippocampus: eleven maps for eleven rooms. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 18428–18435 (2014).

101. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.* **5**, e1000291 (2009).

102. Klukas, M. *et al.* Fragmented spatial maps: State abstraction and efficient planning from surprisal. *bioRxiv* (2021).

103. Whittington, M., Traub, R., Kopell, N., Ermentrout, B. & Buhl, E. Inhibition-based rhythms: experimental and mathematical observations on network dynamics. *Int. J. Psychophysiol.* **38**, 315–336 (2000).

104. Sanders, H., Wilson, M. A. & Gershman, S. J. Hippocampal remapping as hidden state inference. *Elife* **9** (2020).

105. Kleinfeld, D. & Sompolinsky, H. Associative neural network model for the generation of temporal patterns. theory and application to central pattern generators. *Biophys. J.* **54**, 1039–1051 (1988).

106. Herz, A. V., Li, Z. & Van Hemmen, J. Statistical mechanics of temporal association in neural networks with transmission delays. *Phys. review letters* **66**, 1370 (1991).

107. Fiete, I. R., Senn, W., Wang, C. Z. H. & Hahnloser, R. H. R. Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. *Neuron* **65**, 563–576 (2010).

108. Hertz, J. A., Krogh, A. & Palmer, R. G. *Introduction to the theory of neural computation.*, vol. 1 of *The advanced book program* (Addison-Wesley, 1991).

109. Klukas, M., Lewis, M. & Fiete, I. Efficient and flexible representation of higher-dimensional cognitive variables with grid cells. *PLoS Comput. Biol.* **16**, e1007796 (2020).

110. Wood, E. R., Dudchenko, P. A., Robitsek, R. J. & Eichenbaum, H. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron* **27**, 623–633 (2000).

111. Grieves, R. M., Wood, E. R. & Dudchenko, P. A. Place cells on a maze encode routes rather than destinations. *Elife* **5**, e15986 (2016).

112. Dombeck, D. A., Harvey, C. D., Tian, L., Looger, L. L. & Tank, D. W. Functional imaging of hippocampal place cells at cellular resolution during virtual navigation. *Nat. neuroscience* **13**, 1433–1440 (2010).

113. Markus, E. J. *et al.* Interactions between location and task affect the spatial and directional firing of hippocampal neurons. *J. Neurosci.* **15**, 7079–7094 (1995).

114. Bird, C. M. & Burgess, N. The hippocampus and memory: insights from spatial processing. *Nat. reviews neuroscience* **9**, 182–194 (2008).

115. Nadel, L. & Moscovitch, M. Memory consolidation, retrograde amnesia and the hippocampal complex. *Curr. opinion neurobiology* **7**, 217–227 (1997).

116. Yates, F. A. *The Art of Memory* (1966).

117. unknown, A. Rhetorica ad herennium (∼80 B.C.).

118. Cicero. De oratore (55 B.C.).

119. Quintilian. Institutio oratoria (∼95 A.D.).

120. Maguire, E. A., Valentine, E. R., Wilding, J. M. & Kapur, N. Routes to remembering: the brains behind superior memory. *Nat. neuroscience* **6**, 90–95 (2003).

121. Ericsson, K. A. *et al.* Memory skills mediating superior memory in a world-class memorist. *Memory* **25**, 1294–1302 (2017).

122. Qureshi, A., Rizvi, F., Syed, A., Shahid, A. & Manzoor, H. The method of loci as a mnemonic device to facilitate learning in endocrinology leads to improvement in student performance as measured by assessments. *Adv. physiology education* **38**, 140–144 (2014).

123. Reggente, N., Essoe, J. K., Baek, H. Y. & Rissman, J. The method of loci in virtual reality: explicit binding of objects to spatial contexts enhances subsequent memory recall. *J. Cogn. Enhancement* **4**, 12–30 (2020).

124. Sousa, A. E., Mahdid, Y., Brodeur, M. & Lepage, M. A feasibility study on the use of the method of loci for improving episodic memory performance in schizophrenia and non-clinical subjects. *Front. Psychol.* **12**, 612681 (2021).

125. Krotov, D. & Hopfield, J. J. Dense associative memory for pattern recognition. *Adv. neural information processing systems* **29**, 1172–1180 (2016).

126. Ramsauer, H. *et al.* Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217* (2020).

127. Hasselmo, M. E. A model of episodic memory: mental time travel along encoded trajectories using grid cells. *Neurobiol. learning memory* **92**, 559–573 (2009).

128. Agmon, H. & Burak, Y. 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* **9** (2020).

129. Benna, M. K. & Fusi, S. Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence. *Proc. Natl. Acad. Sci.* **118**, e2018422118 (2021).

130. Sharma, S., Curtis, A., Kryven, M., Tenenbaum, J. & Fiete, I. Map induction: Compositional spatial submap learning for efficient exploration in novel environments. *ICLR* (2022).

131. Tsoi, S. Y. *et al.* Telencephalic outputs from the medial entorhinal cortex are copied directly to the hippocampus. *Elife* **11**, e73162 (2022).

132. Sipser, M. & Spielman, D. A. Expander codes. *IEEE transactions on Inf. Theory* **42**, 1710–1722 (1996).

133. Demircigil, M., Heusel, J., Löwe, M., Upgang, S. & Vermet, F. On a model of associative memory with huge storage capacity. *J. Stat. Phys.* **168**, 288–299 (2017).

134. Dragoi, G. & Tonegawa, S. Preplay of future place cell sequences by hippocampal cellular assemblies. *Nature* **469**, 397–401 (2011).

135. Dresler, M. *et al.* Mnemonic training reshapes brain networks to support superior memory. *Neuron* **93**, 1227–1235 (2017).

136. Yim, M. Y., Sadun, L. A., Fiete, I. R. & Taillefumier, T. Place-cell capacity and volatility with grid-like inputs. *Elife* **10**, e62702 (2021).

137. https://www.kaggle.com/datasets/whitemoon/miniimagenet.

A Supplementary Information

This SI is structured as follows: First, we present the quantification metrics and tools used to generate the numerical results presented in this paper in SI Sec. B. Then, in SI Sec. C-D, we provide theoretical guarantees of the results about Vector-HaSH, first in SI Sec. C focusing on the grid-hippocampal memory scaffold and in SI Sec. D focusing on heteroassociative learning with the sensory cells. In particular, in SI Sec. C we prove that the setup of the memory scaffold described in the main text results in a network with an exponentially large number of robust fixed points with large basins of attraction. Then, in SI Sec. D, we first demonstrate that heteroassociative pseudoinverse learning will result in a memory continuum with the desired properties, and then show that one may feasibly replace the pseudoinverse learning with simpler Hebbian learning and continue to obtain qualitatively similar results.

B Quantification Metrics

B.1 Software and Data

The source code for the models presented in this paper will be made available at the following GitHub repository upon acceptance:
`https://github.com/FieteLab/`

B.2 Mutual Information

In this Appendix, unless otherwise specified, we use s_i^μ to represent the i^{th} bit of the μ^{th} pattern stored in the network, and σ_i to represent the i^{th} bit of the pattern recovered by the network. Here we primarily consider the case of random patterns such that bits of s^μ are independently sampled from i.i.d. random variables. This allows us to calculate information theoretic quantities for a single bit, and then scale the calculation by the pattern length to obtain the corresponding quantities for entire patterns.

Further, for simplicity of notation in this section, we overload σ and s to also represent the random variables from which the stored patterns and recovered patterns are being sampled.

We characterize the quality of pattern recovery by a network through the *mutual information* between stored patterns s and recovered patterns σ. For discrete random variables, the mutual information can be quantified as:

$$MI(\sigma;s) = H(\sigma) - H(\sigma|s), \tag{27}$$

where $H(\sigma)$ is the information entropy of the recovered pattern σ,

$$H(\sigma) = -\sum_{\sigma} P(\sigma)\log P(\sigma) \tag{28}$$

and $H(\sigma|s)$ is the conditional entropy of the recovered pattern given the stored pattern s,

$$H(\sigma|s) = -\sum_s \sum_\sigma P(\sigma, s) \log P(\sigma|s). \tag{29}$$

As we now show in the following sections, the mutual information can be explicitly computed for dense and sparse random binary patterns.

B.2.1 Dense binary patterns

For unbiased random binary {-1,1} patterns,

$$H(\sigma) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1.$$

Further, since we assumed that each bit is independent, we obtain

$$P(\sigma|s) = (1 + m\sigma s)/2, \tag{30}$$

where m is the overlap between the stored and recovered pattern, $m = \frac{1}{N}\sum_i \sigma_i s_i$[98]. Using Eq. (29), this can be used to obtain

$$H(\sigma|s) = -\frac{1}{2}\left(\frac{1+m}{2}\log\frac{1+m}{2} + \frac{1-m}{2}\log\frac{1-m}{2}\right) - \frac{1}{2}\left(\frac{1-m}{2}\log\frac{1-m}{2} + \frac{1+m}{2}\log\frac{1+m}{2}\right) \tag{31}$$

$$= -\frac{1+m}{2}\log\left(\frac{1+m}{2}\right) - \frac{1-m}{2}\log\left(\frac{1-m}{2}\right). \tag{32}$$

Following Eq. (27) we thus obtain

$$MI(\sigma;s) = 1 + \frac{1+m}{2}\log\left(\frac{1+m}{2}\right) + \frac{1-m}{2}\log\left(\frac{1-m}{2}\right) \tag{33}$$

B.2.2 Sparse binary patterns

For sparse binary {0,1} patterns, let p denote the fraction of "1" bits in the stored pattern (i.e., the average activity of the stored pattern). Let the average activity of the recovered pattern be denoted as $q = \sum_i \sigma_i/N$.

Let P_{1e} be the probability of error in a bit of σ if the corresponding bit of s is 1, and P_{0e} be the error probability in a bit of σ if the corresponding bit of s is 0. Then,

$$H(\sigma) = -[q\log(q) + (1-q)\log(1-q)] \tag{34}$$

$$H(\sigma|s) = -p[P_{1e}\log(P_{1e}) + 1 - P_{1e}\log(1-P_{1e})] - (1-p)[P_{0e}\log(P_{0e}) + 1 - P_{0e}\log(1-P_{0e})] \tag{35}$$

$$\tag{36}$$

To obtain the probabilities P_{1e} and P_{0e}, we compute the overlap m and the average activity of the recovered pattern q in terms of these probabilities as

$$m = (1/N)\sum_i \sigma_i s_i = p(1-P_{1e}), \tag{37}$$

$$q = \sum_i \sigma_i/N = p(1-P_{1e}) + (1-p)P_{0e} = m + (1-p)P_{0e}. \tag{38}$$

These equations can then be solved to obtain

$$P_{1e} = 1 - m/p, \tag{39}$$

$$P_{0e} = \frac{q-m}{1-p}, \tag{40}$$

which can then be used to compute $MI(\sigma;s)$ using Eq. (27).

B.2.3 Continuous random normal patterns

The calculation of mutual information so far has been restricted to the case of discrete binarized patterns. For continuous valued patterns (as in Fig. 3), entropy is ill-defined via Eq. (28). Instead, in this case we can defined the differential entropy as

$$h(X) = -\int_{-\infty}^{\infty} \phi(x) \log \phi(x) dx = \mathbb{E}[-\log \phi(x)], \tag{41}$$

where $\phi(x)$ is the probability density function of the random variable X.

For random continuous patterns with patterns are sampled from a normal distribution with zero mean and unit variance, this gives

$$h(X) = \log \sqrt{2\pi e}. \tag{42}$$

The conditional entropy can similarly be calculated as

$$h(X) = \log \sqrt{2\pi e(1-r^2)}, \tag{43}$$

where r is the correlation coefficient between X and Y. This can be used to obtain the mutual information

$$MI(X;Y) = \log \sqrt{\frac{1}{1-r^2}} \tag{44}$$

Thus in the case of random normal patterns the mutual information between the stored pattern s and the recovered pattern σ can be computed directly through the correlation between them using Equation 44 above.

B.3 Metrics

We quantify the recovery error, i.e., the error between the stored pattern and the recovered pattern in the network by computing the L^2 norm of the difference between stored and recovered patterns. This recovery error is then used to quantitatively apply a recovery threshold to ascertain the capacity of the memory scaffold.

After choosing a recovery threshold (see *Methods*), the capacity of the network is defined as the largest number of stored patterns for which the average recovery error across patterns is below threshold.

C Theoretical Results on the Memory Scaffold

First, we prove that the memory scaffold network has $\prod_{i=1}^{M} \lambda_i^2$ fixed points, while having only $\mathcal{O}(M \sum_i \lambda_i^2)$ synapses, establishing an exponentially large number of fixed points. Then, we demonstrate that each of these basins are maximally large, and finally demonstrate that these basins are convex, ensuring robustness of basins and protection against adversarial input.

C.1 Justification for the exponentially large capacity of the memory scaffold

We first provide broad qualitative justification for why the memory scaffold as constructed in Vector-HaSH is capable of storing such a large number of fixed points, then present a mathematical proof in a simplified setting. In this subsection, for ease of notation, we denote the number of phases in the i^{th} grid module, λ_i^2, as l_i.

Unlike associative memory in the usual context of random patterns (as in the random shuffled hippocampal states considered in Figs. 2d,f and 3d), note that the hippocampal states are determined by a random projection of the structured grid states. As a result, the predefined hippocampal states inherit similar pattern-pattern correlations as the predefined grid cell states. This allows for Hebbian learning to act more efficiently in learning pairwise correlation resulting in a high capacity. Indeed, while in Hopfield networks any given fixed point is destabilized due to interference from other fixed points (resulting in catastrophic forgetting when a large number of fixed points have been memorized), the shared pattern-pattern correlations in the memory scaffold result in the interference terms being positively correlated with each fixed point (which also leads to the scaffold generalization properties Fig. 2f, Sec. C.4).

To show this result more quantitatively, recall that

$$g(t+1) = \mathrm{CAN}[W_{gh}h(t)]. \tag{45}$$

where $\mathrm{CAN}[x]$ is a nonlinear function that acts independently on each module of grid cells, such that $\mathrm{CAN}[x]$ will be a vector with exactly one element set to "1" in each of the l_i indices corresponding to each module. and all other elements set to "0". Further, the element set to 1 in a given module corresponds to the same index as the largest element x within that module.

Corresponding to the state h^μ, consider the pattern $h(t) = h^\mu + \zeta$, where ζ represents a random noise vector. For simplicity, we assume that ζ is a continuous-valued vector whose each component is drawn independently from a normal distribution with zero mean and variance ε^2.

From $h(t)$, we aim to recover $g(t+1) = g^\mu$ via the mapping W_{gh}. For ease of notation, we denote the prespecified random projection W_{hg} as W.

$$h^\mu = \Phi[Wg^\mu], \tag{46}$$

where Φ represents the neural transfer function for the grid to hippocampal synapses, which we implement as a thresholded rectifying function (see *Methods*). We implement W such that each element is independently sampled from a prespecified distribution (see *Methods*). Without loss of generality, we can assume that this distribution has zero mean and unit variance, since any transformations of the mean and variance can be absorbed into the nonlinearity Φ.

Now, from the definition of W_{gh} and h^μ,

$$\begin{aligned} g(t+1) &= \mathrm{CAN}[GH^T(\Phi(Wg^\mu) + \zeta)], & (47)\\ &= \mathrm{CAN}[L\Phi(G^TW^T)\Phi(Wg^\mu)/N_h + G\Phi(G^TW^T)\zeta/N_h], & (48)\end{aligned}$$

where we have added a scaling factor $1/N_h$ that leaves the CAN continuous attractor dynamics unchanged, but will be useful for normalization of random variables later in our calculation.

For analytic simplicity, we make the assumption that the nonlinearity Φ in the above equation can be ignored. While this is a gross simplification, the obtained results are broadly consistent with the numerical observations in Fig. 2. This approximates the above equation to

$$g(t+1) = \mathrm{CAN}[GG^TW^TWg^\mu/N_h + GG^TW^T\zeta/N_h] = \mathrm{CAN}[Ag^\mu + Z], \tag{49}$$

where $A = (GG^T)(W^TW/N_h)$, and $Z = GG^TW^T\zeta/N_h$.

Since each element of the $N_h \times N_g$ matrix W was drawn independently from a normal distribution with unit variance, W^TW (and hence A) can be treated as a matrix random variable. Under the distribution of the matrix random variable A and the vector random variable Z we will compute the probability of $g(t+1) = g^\mu$. Note that this simplification of the problem into Eq. (49) has fundamentally relied on Eq. (46), which establishes hippocampal states as being derived from random projections of grid states. Qualitatively, hippocampal states being projections of grid states results in a similarity of state-state relationships between grid states and hippocampal states. As a result, overloading the scaffold network weights with a large number of patterns will not result in loss of previously stored information through interference; instead, pattern interference will re-inforce previously stored patterns (which also results in the strong generalization property, Sec. C.4). In contrast, if the hippocampal states were arbitrarily determined (such as through consideration of random sparse vectors, or sensory-input-dependent vectors), then interference due to storage of additional patterns would result in catastrophic forgetting, as in classic Hopfield memory.

We first focus on the structure of the matrix GG^T. This matrix will have a block structure, with the sizes of the blocks determined by the number of phases in each grid module, l_i. In particular we write GG^T as

$$GG^T = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \cdots \\ \Gamma_{21} & \Gamma_{22} & \cdots \\ \vdots & & \ddots \end{pmatrix}, \tag{50}$$

with each Γ_{ij} being a submatrix of size $l_i \times l_j$. From the structure of the grid code matrix G, it follows that

$$\Gamma_{ii} = \left(\prod_{k \neq i} l_k\right) \mathbb{I} = \gamma_{ii} \mathbb{I} \tag{51}$$

and

$$\Gamma_{ij} = \left(\prod_{k \neq i,j} l_k\right) \mathbf{1} = \gamma_{ij} \mathbf{1}, \tag{52}$$

where $\mathbb{I}$ is an appropriately sized identity matrix, and $\mathbf{1}$ is an appropriately sized matrix with each element equal to 1. This can be shown by noting that $GG^T = \sum_\mu g^\mu (g^\mu)^T$, and that each $g^\mu (g^\mu)^T$ will be a matrix with a single nonzero element equal to 1 in each Γ block of GG^T.

We now compute the distribution of the matrix random variable $W^T W$. As argued above, each element of the $N_h \times N_g$ matrix W can be assumed to be drawn independently from a distribution with zero mean and unit variance. As is justified later, we can assume that these elements are drawn from a normal distribution in particular, since we shall be applying central limit theorem which will wash away particulars of the shape of the distribution.

Thus $W^T W$ can thus be approximated to have each diagonal element distributed as the sum of the squares of N_h standard normal variables, and each off-diagonal element distributed as the sum of the products of N_h pairs of uncorrelated standard normal variables. Thus

$$W^T W \sim \begin{pmatrix} \chi^2(N_h) & \mathcal{NP}(N_h) & \cdots \\ \mathcal{NP}(N_h) & \chi^2(N_h) & \cdots \\ \vdots & & \ddots \end{pmatrix}, \tag{53}$$

where $\chi^2(N)$ is the sum of N i.i.d. χ^2 distributions, and $\mathcal{NP}(N)$ is the sum of N i.i.d. normal product distributions (i.e., the distribution of the product of two i.i.d. standard normal variables). Note that we have suppressed the indices on each matrix element, however it should be noted that each element is an independent sample from the distribution and are identical in distribution but not in value.

In the large N_h limit, each of these matrix elements is the sum of a large number of random variables and can hence be approximated as a normal distribution due to central limit theorem. Thus, $\chi^2(N_h) \sim \mathcal{N}(N_h, 2N_h)$, and $\mathcal{NP}(N_h) \sim \mathcal{N}(0, N_h)$, where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2.[3]

We thus treat $W^T W / N_h$ as a matrix random variable with elements on the diagonal being drawn from a distribution $\mathcal{D}$, having unit mean and a variance of $2/N_h$; and elements on the off-diagonal being drawn from a distribution $\mathcal{O}$, having zero mean and $1/N_h$ variance. For ease of calculation, we write this matrix as having a block structure similar to GG^T, given by

$$W^T W / N_h = \begin{pmatrix} w_{11} & w_{12} & \cdots \\ w_{21} & w_{22} & \cdots \\ \vdots & & \ddots \end{pmatrix}, \tag{54}$$

---

[3] Had we not earlier assumed that elements of W are drawn from a normal distribution we would have arrived at this same result with different intermediate distribution instead of χ^2 and $\mathcal{NP}$

with w_{ij} being an $l_i \times l_j$ matrix such that w_{ii} has diagonal entries drawn from $\mathcal{D}$ and off diagonal entries drawn from $\mathcal{O}$, and w_{ij} for $i \neq j$ being a matrix with all entries drawn from $\mathcal{O}$

We can now compute the distribution of the elements of A. The matrix A will have a similar block structure to GG^T,

$$\begin{aligned} A &= GG^T W^T W / N_h && (55)\\ &= \begin{pmatrix} A_{11} & A_{12} & \dots \\ A_{21} & A_{22} & \dots \\ \vdots & & \ddots \end{pmatrix}, && (56) \end{aligned}$$

with

$$A_{ij} = \sum_k \Gamma_{ik} w_{kj}. \tag{57}$$

Since each w_{ij} consists of elements drawn from random normal distributions, the distribution of the matrix variables A_{ij} can be exactly computed through sums of random normal variables. Even without explicit computation, we can ascertain certain properties of A given the symmetry of grid states across module-preserving permutations. In particular, A_{ii} will be a matrix random variable with diagonal element drawn from an i.i.d. distribution $\mathcal{A}^d_{ii}$, and each off-diagonal element drawn from a different i.i.d. distribution $\mathcal{A}^o_{ii}$. In contrast, A_{ij} for $i \neq j$ will have all elements drawn from an i.i.d. distribution $\mathcal{A}_{ij}$.

We first consider A_{ii}.

$$\begin{aligned} A_{ii} &= \sum_k \Gamma_{ik} w_{ki} && (58)\\ &= \Gamma_{ii} w_{ii} + \sum_{k \neq i} \Gamma_{ik} w_{ki} && (59)\\ &= \gamma_{ii} \begin{pmatrix} \mathcal{D} & \mathcal{O} & \dots \\ \mathcal{O} & \mathcal{D} & \dots \\ \vdots & & \ddots \end{pmatrix} + \gamma_{ik} \begin{pmatrix} \sum_{l_k \text{ terms}} \mathcal{O} & \sum_{l_k \text{ terms}} \mathcal{O} & \dots \\ \sum_{l_k \text{ terms}} \mathcal{O} & \sum_{l_k \text{ terms}} \mathcal{O} & \dots \\ \vdots & & \ddots \end{pmatrix}, && (60) \end{aligned}$$

where we have omitted subscripts on individual random variables for simplicity, but it should be noted that each random variable is i.i.d., including the summands in the above expressions. Thus,

$$\begin{aligned} \mathcal{A}^d_{ii} &\sim \gamma_{ii}\mathcal{D} + \sum_{k \neq i} \gamma_{ik} \sum_{l_k \text{ terms}} \mathcal{O} && (61)\\ \mathcal{A}^o_{ii} &\sim \gamma_{ii}\mathcal{O} + \sum_{k \neq i} \gamma_{ik} \sum_{l_k \text{ terms}} \mathcal{O}. && (62) \end{aligned}$$

A similar calculation can be done to obtain

$$\mathcal{A}_{ij} \sim \gamma_{ij}\mathcal{O} + \gamma_{ij} \left(\mathcal{D} + \sum_{l_j - 1 \text{ terms}} \mathcal{O} \right) + \sum_{k \neq i,j} \gamma_{ik} \sum_{l_k \text{ terms}} \mathcal{O}. \tag{63}$$

From the same symmetry as in A, we can also argue that elements of Z can also be split into a similar block structure, $Z^T = (Z_1^T Z_2^T \dots)^T$, with all λ_i^2 elements of Z_i drawn from an i.i.d distribution $\mathcal{Z}_i$. More specifically, $Z = GG^T W^T \zeta / N_h$. First note that $W^T \zeta$ will be a random vector with each element constructed from the sum of N_h i.i.d. normal product distributions multiplied by the scale of ζ, i.e., ε. Thus $W^T \zeta / N_h$ is identically distributed to $\varepsilon\mathcal{O}$. Left multiplying this vector with GG^T we obtain in the i^{th} subvector

$$\mathcal{Z}_i = \varepsilon \left[\gamma_{ii}\mathcal{O} + \sum_{j \neq i} \gamma_{ij} \sum_{l_j \text{ terms}} \mathcal{O} \right], \tag{64}$$

where again we have omitted subscripts on individual random variables for simplicity.

Let the mean and standard deviation of $\mathcal{A}^d_{ii}$, $\mathcal{A}^o_{ii}$, $\mathcal{A}_{ij}$ and $\mathcal{Z}_i$ be denoted as $\mu_{Ad_{ii}}$,$\sigma_{Ad_{ii}}$; $\mu_{Ao_{ii}}$,$\sigma_{Ao_{ii}}$; $\mu_{A_{ij}}$,$\sigma_{A_{ij}}$; and μ_{Z_i},σ_{Z_i} respectively. Since $\mathcal{D} \sim \mathcal{N}(1, 2/N_h)$ and $\mathcal{O} \sim \mathcal{N}(0, 1/N_h)$, we obtain

$$\mu_{Ad_{ii}} = \gamma_{ii} \tag{65}$$

$$\sigma^2_{Ad_{ii}} = \left[2\gamma^2_{ii} + \sum_{k\neq i} l_k \gamma^2_{ik}\right]/N_h \tag{66}$$

$$\mu_{Ao_{ii}} = 0 \tag{67}$$

$$\sigma^2_{Ao_{ii}} = \left[\gamma^2_{ii} + \sum_{k\neq i} l_k \gamma^2_{ik}\right]/N_h \tag{68}$$

$$\mu_{A_{ij}} = \gamma_{ij} \tag{69}$$

$$\sigma^2_{A_{ij}} = \left[\gamma^2_{ii} + \gamma^2_{ij}(2 + (l_j - 1)) + \sum_{k\neq i,j} l_k \gamma^2_{ik}\right]/N_h \tag{70}$$

$$\mu_{Z_i} = 0 \tag{71}$$

$$\sigma^2_{Z_i} = \varepsilon^2 \left[\gamma^2_{ii} + \sum_{j\neq i} l_j \gamma^2_{ij}\right]/N_h \tag{72}$$

Next, we note the conditions on these blocks to make $g(t+1) = g^\mu$, the condition necessary for g^μ to be a scaffold fixed point. Without loss of generality, we assume that g^μ corresponds to the grid state such that each l_i length subvector of g^μ has the first element set to 1 and all others set to zero. The i^{th} subvector of $Ag^\mu + Z$ will then have the first element given by

$$\mathcal{A}^d_{ii} + \sum_{j\neq i} \mathcal{A}_{ij} + \mathcal{Z}_i, \tag{73}$$

and all other elements given by

$$\mathcal{A}^o_{ii} + \sum_{j\neq i} \mathcal{A}_{ij} + \mathcal{Z}_i. \tag{74}$$

For this module to be correctly reconstructed through the continuous attractor network dynamics, we require that the first element of the subvector to be larger than the others. Thus, the probability of the correct reconstruction is given by

$$P(g(t+1) = g^\mu) = P(\text{Eq. (73) - Eq. (74)} > 0). \tag{75}$$

As seen earlier, each of these random variables are being drawn from a normal distribution (due to central limit theorem in the limit of large N_h). In terms of the parameters of these normal distributions, Eq. (75) can be written as

$$P(g(t+1) = g^\mu) = P(\mathcal{N}(\mu_{Ad_{ii}} - \mu_{Ao_{ii}}, \sigma^2_{Ad_{ii}} + \sigma^2_{Ao_{ii}} + 2\sum_{j\neq i} \sigma^2_{A_{ij}} + 2\sigma^2_{Z_i}) > 0). \tag{76}$$

For ease of notation, we define

$$\mu_{AZ} = \mu_{Ad_{ii}} - \mu_{Ao_{ii}}, \tag{77}$$

$$\sigma^2_{AZ} = \sigma^2_{Ad_{ii}} + \sigma^2_{Ao_{ii}} + 2\sum_{j\neq i} \sigma^2_{A_{ij}} + 2\sigma^2_{Z_i} \tag{78}$$

$$\tag{79}$$

such that the right-hand side of Eq. (76) is equal to $P(\mathcal{N}(\mu_{AZ},\sigma_{AZ}^2>0)$. This can then be computed as

$$P(l(t+1)=l^{\mu})=\frac{1}{2}\left[1+\mathrm{erf}\left(\frac{\mu_{AZ}}{\sigma_{AZ}\sqrt{2}}\right)\right]. \tag{80}$$

The above-derived expressions for the terms in μ_{AZ} and σ_{AZ} can be simplified to

$$\mu_{AZ}=\gamma_{ii} \tag{81}$$

and

$$N_h\sigma_{AZ}^2=\gamma_{ii}^2(2M+1)+(2M+1)\sum_{k\neq i} l_k\gamma_{ik}^2+2\sum_{j\neq i}\gamma_{ij}^2. \tag{82}$$

Recall that $\gamma_{ii}=P/l_i$ and $\gamma_{ij}=P/(l_il_j)$, for $P=\prod_i l_i$. Thus, the ratio μ_{AZ}^2/σ_{AZ}^2 simplifies to

$$\frac{\mu_{AZ}^2}{\sigma_{AZ}^2}=\frac{N_h}{2M+1+(2M+2)\sum_{k\neq i}(1/l_k)+2\sum_{k\neq i}(1/l_k^2)+2\varepsilon^2[1+\sum_{k\neq i}(1/l_k)]} \tag{83}$$

Inverting the obtained expression allows for computation of N_h^*,

$$N_h^*=c\left[2M+1+(2M+2)\sum_{k\neq i}\frac{1}{l_k}+2\sum_{k\neq i}\frac{1}{l_k^2}+2\varepsilon^2\left(1+\sum_{k\neq i}\frac{1}{l_k}\right)\right], \tag{84}$$

where $c=2\left[\mathrm{erf}^{-1}(1-2P)\right]^2$ and P is the threshold selected for accuracy of the recovered pattern. This allows us to estimate the critical number of hippocampal cells (as a function of the number of grid cell modules, M, the period of the modules $l_i=\lambda_i^2$,and the input noise ε) beyond which the hippocampal memory scaffold stores all grid-hippocampal states as fixed points.

If the number of grid cells far exceeds the number of modules (as would be expected[81]), then $\lambda_k\gg M$ and thus $l_k\gg M_2$ and the summands in Eq. (84) can all be ignored. This makes N_h^* asymptotically independent of the grid periods, and is given by $N_h^*\approx c(2M+1+2\varepsilon^2)$. This can be seen qualitatively in Fig. 2, where for a fixed M, the critical number of hippocampal cells N_h^* approaches a constant with increasing N_g (and hence increasing l_k). Moreover, if $\varepsilon\ll 1$ and $M\gg 1$, then $N_h^*=\mathcal{O}(M)$. This has been verified qualitatively in Fig. 2, where N_h^* increases linearly with increasing number of modules M. Note that due to the simplifications necessary for this analytic result, N_h^* obtained from Eq. (84) are not directly comparable to numerics in Fig. 2, however the above-mentioned qualitative trends all seem to hold.

These results thus demonstrate a crucial property of the hippocampal memory scaffold network — it has $\mathcal{O}(N_gN_h^*)=\mathcal{O}(M\sum_k\lambda_k^2)\sim\mathcal{O}(M^2\lambda^2)$ synapses while having $\prod_k\lambda_k^2\sim\lambda^{2M}$ fixed points. Thus, the number of fixed points grows exponentially faster than the number of synapses in the network, resulting in the network being useful as a memory scaffold as in MESH[87].

C.2 Memory Scaffold has Maximally Sized Basins of Attraction

Due to the symmetries in grid code, we argue here that the memory scaffold in Vector-HaSH has no spurious fixed points, and has convex, maximally sized basins of attraction that are equal in volume.

First, we note that as a result of the CAN dynamics in the grid layer (cf. Eq. (4)), the only possible grid states are the $\prod_i K_i$ modular one-hot states. Correspondingly, the hippocampal states (determined by random projections of the grid states) must then also be one of the $\prod_i K_i$ states, establishing that no spurious fixed points can arise.

Thus, the union of the basins about each of the predefined fixed points of the grid-hippocampal scaffold cover the entire space $\mathbb{R}^{N_h}$. Note next that each h^μ are equivalent, i.e., there is no special μ since each g^μ is equivalent up to a module-preserving permutation of bits and h^μ are determined by a random projection of g^μ. Thus, $\mathbb{R}^{N_h}$ must be partitioned into basins with equal volume that are maximally large (and hence are of the same volume as the Voronoi cell about these fixed points).

C.3 Convexity of Scaffold Basins

The existence of maximally sized equi-volumed basins around each predefined scaffold fixed point, as we have shown, is not sufficient to guarantee robustness to noise. A large basin could in principle have some boundaries that come arbitrarily close to the fixed points – such a situation holds for instance when a system is susceptible to adversarial inputs, where a very small perturbation of the input leads to a very different classification as an output. Noise robustness requires a second condition, that of basin convexity. Here we demonstrate that the obtained basins are convex, and thus the large basins must result in basin boundaries that are well separated from the fixed points themselves.

We are interested in the basins in the space $\mathbb{R}^{N_h}$: the hippocampus receives sensory input from the sensory layer, mediating the recall of scaffold states. Thus, noise robustness will hence be required there. The broad idea of the proof is as follows: first, we demonstrate that perturbations in the hippocampal latent space are equivalent to considering real-valued perturbations with small magnitudes in the grid-cell layer latent space. Then we show that the continuous attractor dynamics on grid cells result in convex basins in the grid-cell space, which directly translates to convex basins in the hippocampal space.

Consider a hippocampal population vector given by a small perturbation to a predefined hippocampal state fixed point h^μ, which we denote as $h = h^\mu + \varepsilon$. Let δ denote the magnitude of the perturbation ε. This hippocampal state is projected onto the grid cells through W_{gh} to obtain $\bar{g}$ before the continuous attractor dynamics, where

$$\begin{aligned}\bar{g} &= W_{gh}h = W_{gh}[h^\mu + \varepsilon] && (85)\\ &= \bar{g}^\mu + W_{gh}\varepsilon. && (86)\end{aligned}$$

Note that $W_{gh}\varepsilon$ will have a magnitude of approximately δ times the magnitude of $\bar{g}^\mu$, and further, the nonzero elements of ε are uncorrelated with h^μ, and hence $W_{gh}\varepsilon$ can be treated as an independent small real-valued perturbation to $\bar{g}^\mu = W_{gh}h^\mu$.

If we can now show that the continuous attractor dynamics on grid cells has a convex basin, that would indicate that all points near g^μ map to g^μ, and since points near h^μ map to points near g^μ, this would imply convexity of basins in p-space.

The symmetry of the grid code implies that it will suffice to show that the basin about any one fixed point is convex. Without loss of generality, we choose the fixed point g^μ as the grid population vector with the first bit in each module set to 1 and all other $\sum_i \lambda_i^2 - M$ bits set to 0. Let x and y be two vectors within the continuous attractor dynamics of g^μ, i.e., $CAN[x] = CAN[y] = g^\mu$. Thus, for the k^{th} module, $x_{k;1} > x_{k,i}$ and $y_{k;1} > y_{k,i}$ for $i > 1$. Adding the two inequalities with coefficients a and $(1-a)$, we obtain $ax_{k;1} + (1-a)y_{k,1} > ax_{k;i} + (1-a)y_{k,i}$ for all $i > 1$ for $0 \le a \le 1$. Thus, continuous attractor dynamics (which enforce modular winner-take-all dynamics) map the k^{th} module of $ax + (1-a)y$ to the k^{th} module of g^μ. Since this holds for all k, thus $\text{CAN}[ax + (1-a)y] = g^\mu$. Hence, for any two vectors x and y in the basin of g^μ, all vectors on the line from x to y also lie in this basin. By definition, this makes the basin of g^μ, and as argued earlier this imposes convexity of basins in the hippocampal cell space.

C.4 Scaffold weights can be learned on a vanishing fraction of all states

As shown in Fig. 2, when the hippocampus-to-grid cell synaptic weights (W_{gh}) are learned on a small number of grid states, the scaffold is able to generalize: all grid states become stable fixed points of the scaffold dynamics. Thus an animal only needs to traverse small regions in space after which the grid-hippocampal scaffold is recurrently stabilized for all other states. Here we show that in the large N_h limit, it will suffice to train on only $M \times \lambda_{max}^2$ patterns for stabilization of the complete scaffold, where λ_{max} is the largest period of any grid module.

Similar to Sec. C.1, we make the grossly simplified assumption that the nonlinearities in the hippocampus can be ignored. The grid cell state would then evolve as

$$g(t+1) = \text{CAN}[GG^T W^T W g^\mu / N_h], \tag{87}$$

where again we add an $1/N_h$ scaling factor that renders the continuous attractor network dynamics unchanged. For g^μ to be a fixed point of the recurrent scaffold dynamics, we thus require that $g(t+1)$ be equal to g^μ. Unlike Sec.

C.1, we assume here that G is the matrix constructed by appending grid cell population vectors over only the first N_{patts} number of states (rather than over the entirety of scaffold states).

As argued earlier in Sec. C.1, for large N_h, the matrix $W^T W/N_h$ can be considered to be a random variable with i.i.d $\mathcal{N}(1,2/N_h)$ random variables on the diagonal, and i.i.d $\mathcal{N}(0,1/N_h)$ entries on the off-diagonal. In the limit of $N_h \to \infty$, these distributions tend to Dirac delta distributions we can thus treat $W^T W/N_h$ as simply being an identity matrix. Thus, in this limit, it suffices to examine the scaffold fixed points under the dynamics

$$g(t+1) = \text{CAN}[GG^T g^\nu]. \tag{88}$$

As earlier, we write GG^T as a block matrix

$$GG^T = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \dots \\ \Gamma_{21} & \Gamma_{22} & \dots \\ \vdots & & \ddots \end{pmatrix}, \tag{89}$$

with each Γ_{ij} being a submatrix of size $\lambda_i^2 \times \lambda_j^2$. We define a construct a sequence of patterns g^μ as follows: let the first pattern g^1 be such that the first element in each λ_i^2 subvector is set to one, and all other elements set to zero. Then, each successive pattern shifts the active element by one, modulo the total number of elements in the subvector λ_i^2. Mapped to real space, this corresponds to the sequence of locations shown in Fig. S12a (*top left*). As we will now show, setting W_{gh} based on only $M \times \lambda_{max}^2$ grid patterns will suffice to stabilize all patterns.

Note that

$$GG^T = \sum_{\mu=1}^{N_{patts}} g^\mu (g^\mu)^T. \tag{90}$$

Each μ term of this summation, $g^\mu (g^\mu)^T$ will be a matrix with exactly one '1' in each block Γ_{ij}, at the location ($\mu \mod \lambda_i^2$, $\mu \mod \lambda_j^2$), and will be zero everywhere else.

For $i \neq j$, the periods λ_i and λ_j are coprime. In this case, we can see that if g^μ and g^ν contribute a '1' at the same location (m,n) in Γ_{ij} then $\mu - \nu$ must be a multiple of $\lambda_i^2 \lambda_j^2$. This can be seen since $m = \mu \mod \lambda_i^2$, thus $\mu = m \mod \lambda_i^2$. Similarly $\nu = m \mod \lambda_i^2$, implying that $\mu - \nu = 0 \mod \lambda_i^2$. Similar reasoning leads to $\mu - \nu = 0 \mod \lambda_j^2$ and thus $\mu - \nu = 0 \mod \lambda_i^2 \lambda_j^2$. Crucially, this means that if $\mu \neq \nu$, then $\mu - \nu$ must be at least $\lambda_i^2 \lambda_j^2$, which is equal to the number of elements in Γ_{ij}.

Thus, if both μ and ν contribute a '1' to (m,n) in Γ_{ij}, it must be that all other elements in Γ_{ij} have been increased by 1 due to patterns between μ and ν. In essence, elements of Γ_{ij} increase sequentially through increasing terms in the summation Eq. (90). Starting from all elements at zero before any learning, all patterns increase to 1 one-at-a-time up to $\lambda_i^2 \lambda_j^2$, all patterns increase up to 2 through the next $\lambda_i^2 \lambda_j^2$ patterns and so on. Crucially, at any point during learning, the largest element of Γ_{ij}, which we denote $\max \Gamma_{ij}$, can differ from the smallest element of Γ_{ij}, which we denote $\min \Gamma_{ij}$, by at most 1.

Next, we observe that for Γ_{ii}, the pattern μ contributes a '1' at the location ($\mu \mod \lambda_i^2$, $\mu \mod \lambda_i^2$). This leads to two observations: first, that Γ_{ii} will have nonzero entries only on its diagonal; second, the smallest element on the diagonal will be $\lfloor N_{patts}/\lambda_i^2 \rfloor$ and the largest element on the diagonal will be $\lceil N_{patts}/\lambda_i^2 \rceil$.

Now, consider the matrix GG^T constructed using Eq. (90), trained through the first N_{patts} patterns. We apply Eq. (88) for a given g^μ, for μ that need not be within $\{1 \cdots, N_{patts}\}$. Note g^μ has a 1 at only one location per module. Thus the i^{th} subvector of $GG^T g^\mu$ can have values as small as $\lfloor N_{patts}/\lambda_i^2 \rfloor + \sum_{j \neq i} \min \Gamma_{ij}$ at the index where g^μ equals 1 in the i^{th} module; and, it can have values as large as $\sum_{j \neq i} \max \Gamma_{ij}$ at the other entries. For this subvector to map to g^μ under the continuous attractor network dynamics, we thus require

$$\lfloor N_{patts}/\lambda_i^2 \rfloor + \sum_{j \neq i} \min \Gamma_{ij} > \sum_{j \neq i} \max \Gamma_{ij}. \tag{91}$$

Thus,

$$\lfloor N_{patts}/\lambda_i^2 \rfloor > \sum_{j\neq i} (\max \Gamma_{ij} - \min \Gamma_{ij}) \tag{92}$$

$$\geq M - 1. \tag{93}$$

Thus $N_{patts} \geq M\lambda_i^2$. Since the correct subvector needs to be recovered for all modules, thus $N_{patts} \geq M\lambda_{max}^2$ for stabilization of *all* grid states g^μ in the large N_h limit.

Note that the proof above relies on the particular ordering of grid and hippocampal states described above. As we demonstrate, this ordering is optimal, and no other ordering of grid states can result in 'faster' generalization to all scaffold fixed points. To see this, note that we showed above that the particular ordering choice made ensures that the largest and smallest elements of Γ_{ij} can differ by at most 1. Moreover, this difference of at most one resulted in the generalization result proved above. Correspondingly, any other ordering that maintains this difference between the largest and smallest elements of Γ_{ij} will also demonstrate generalization to all scaffold states at $N_{patts} \geq M\lambda_{max}^2$. Faster generalization could only be possible if the elements of Γ_{ij} we all identical, leading to a difference of zero. This is however impossible, since the sum of elements in Γ_{ij} is equal to N_{patts}, which is increasing in steps of 1 and is thus not always divisible by the number of entries in Γ_{ij}, i.e., $\lambda_i^2 \times \lambda_j^2$. Conversely, any other ordering, will result in a potentially larger difference between the smallest and largest elements of Γ_{ij}, which (following Eq. 92) will thus require a larger number of patterns to generalize to all scaffold states.

However, as noted in Fig. 2g, other contiguous orderings of grid states result in generalization upon learning an approximately similar number ($\mathcal{O}(M\lambda_{max}^2)$) of patterns.

D Theoretical Results on Heteroassociative Learning

Here we demonstrate that pseudoinverse learning first perfectly recovers the hippocampal states provided that $N_s > N_{patts}$ (in the noise-free case). Following the memory scaffold results proven earlier, reconstruction of the correct hippocampal states then results in correct retrieval of the corresponding label layer states. Next, we prove that for $N_{patts} < N_h$, the reconstructed feature layer states are also perfectly reconstructed, and for larger N_{patts} the overlap of the stored and recovered patterns decays gracefully as described in the main text. We then prove that given an ideal memory scaffold, heteroassociative *Hebbian* learning is also sufficient to obtain a memory scaffold with the same qualitative properties, with only a smaller prefactor on the memory capacity.

D.1 Perfect Reconstruction of Hippocampal States Through Heteroassociative Pseudoinverse Learning

The projection of the learned sensory inputs onto the hippocampus is given by $W_{hs}S = HS^+S = H\Lambda_S$, where $\Lambda_S = S^+S$ is an orthogonal projection operator onto the range of S^T. If $N_s \geq N_{patts}$, S has linearly independent columns, and $\Lambda_S = \mathbb{I}$, the identity matrix. Thus, $W_{hs}S = H$, i.e., cuing any memorized patterns results in accurate reconstruction of the corresponding hippocampal scaffold state

However, to examine Vector-HaSH as an associative memory, it is necessary to examine to reconstruction of the correct hippocampal scaffold state when cued with noisy or corrupted versions of the memorized patterns as well. Following the results of SI Sec. C.3 and C.2, we see that the memory scaffold has maximally large convex basins of attraction. Note that once W_{hs} has been trained with pseudo-inverse learning, the mapping from the sensory layer to the hippocampal layer is simply a linear transformation, that maps stored sensory patterns to their corresponding hippocampal scaffold states. Thus, the regions in sensory space that map to a given scaffold state must simply be a lower-to-higher-dimensional linear transformation of convex basins in the scaffold space about the chosen state.

Hence, the basins of attraction for a given scaffold state must be convex regions in the sensory input state that include the sensory pattern that has been associated with that scaffold state.

D.2 Perfect Reconstruction of N_h Sensory States Through Heteroassociative Pseudoinverse Learning

To show that up to $N_{patts} \leq N_h$ sensory inputs can be perfectly reconstructed through Vector-HaSH, we require that the matrix of fixed point hippocampal states H be strongly full rank. While we do not rigorously prove that H is full rank, we provide intuitive justification for the same. First note that $\text{rank}(W_{hg}G) = \text{rank}(G) = N_G - M + 1$[136], which does not need to be as large as N_h.

Applying a thresholded rectifying function, $H = \text{ReLU}[W_{hg}G - \Theta]$, effectively acts as an independent random perturbation to the elements of H. Assuming that these perturbations are truly random, H (and submatrices of H formed by selecting varied numbers of fixed points) will become full rank. This is numerically verified in Fig. S10, where the rank can be seen to be $\min(N_{patts}, N_h)$.

We can now show that the "knee" of the Vector-HaSH memory continuum must be at N_h, with $N_{patts} \leq N_h$ sensory states being perfectly reconstructed.

The projection of the hippocampus states onto the sensory layer is given by $W_{sh}H = SH^+H = S\Lambda_H$, where $\Lambda_H = H^+H$ is an orthogonal projection operator onto the range of H^T. Since H is strongly full rank (as justified above), thus for up to $N_{patts} \leq N_h$, the projection operator Λ_H will equal $\mathbb{I}$, the identity matrix. Thus $W_{sh}P = S$.

D.3 Mutual information recalled in Vector-HaSH scales as $1/N_{patts}$

Let $\bar{s}^\mu$ be the reconstruction of pattern s^μ in the feature layer before the application of the sign nonlinearity in Eq. (12), i.e., $\bar{s}^\mu = W_{sh}h^\mu$. Correspondingly, let $\bar{S}$ be the matrix constructed with $\bar{s}^\mu$ as its columns, i.e., $\bar{S}_{i\mu} = \bar{s}^\mu_i$. In this notation, we wish to prove that $s^\mu \cdot \bar{s}^\mu / |s^\mu|^2 = N_h/N_{patts}$.

As earlier, $\bar{S} = S\Lambda_P$. Since $N_{patts} > N_h$, $\text{rank}(H) = N_h$, and the projection operator Λ_H is thus no longer an identity operator. Instead, Λ_H projects on to the N_h-dimensional hyperplane $\mathcal{P}_H$ spanned by the rows of H. Notationally, let $\bar{s}_i$ be the vector corresponding to the i^{th} row of $\bar{S}$, and similarly, let s_i be the vector corresponding to the i^{th} row of S. In this notation, the vectors $\bar{s}_i$ (i.e., the rows of $\bar{S}$) are the vectors obtained by projecting s_i (i.e., the rows of S) onto $\mathcal{P}_H$.

By construction s_i are N_{patts}-dimensional random vectors with no privileged direction. Thus, $|s_i|^2$, the squared magnitude along each dimension, will on average be equally divided across all dimensions. Hence, on average, the component of s_i projected onto $\mathcal{P}_H$ (i.e., $\bar{s}_i$) will have a squared magnitude of $N_h|s_i|^2/N_{patts}$ and thus $|\bar{s}_i| = |s_i|\sqrt{N_h/N_{patts}}$. However, $|\bar{s}_i|$ is also the cosine of the angle between s_i and the hyperplane $\mathcal{P}_H$, and hence averaged over i,

$$s_i \cdot \bar{s}_i = |s_i||\bar{s}_i|\sqrt{N_h/N_{patts}} = |s_i|^2\sqrt{N_h/N_{patts}}. \tag{94}$$

Note that $\sum_i (s_i \cdot \bar{s}_i) = \sum_\mu (s^\mu \cdot \bar{s}^\mu)$, and $\sum_i |s_i|^2 = \sum_\mu |s^\mu|^2$. Thus the above equation can be rewritten as

$$\langle s^\mu \cdot \bar{s}^\mu \rangle_\mu = \langle |s^\mu|^2 \rangle_\mu \sqrt{N_h/N_{patts}}, \tag{95}$$

where $\langle\rangle_\mu$ denotes an average over all patterns μ.

In the notation of Eq. 33, this gives $m = \sqrt{N_h/N_{patts}}$. In the limit of small m, note that $\log(1+m) \approx m$, and the right-hand side of Eq. 33 can simply be approximated in the asymptotic limit as

$$MI \approx m^2,$$

and thus mutual information scales as

$$MI \approx \frac{N_h}{N_{patts}} \tag{96}$$

As a consequence of this result, note that since the mutual information is always positive and only smoothly degrades 1 with increasing N_{patts}, thus the recovered state only gradually moves away from the true pattern in sensory space (cf. Fig. 3g). For random uncorrelated sensory patterns, the boundary of the Voronoi cell about a memorized pattern corresponds to the boundary at which no information is being recovered specific to one particular pattern. Thus, an always positive mutual information indicates that the recovered pattern always remains with the Voronoi cell corresponding to the true pattern, i.e., the recovered pattern is always closer to the correct patterns as compared to any other pattern.

D.4 Space- and time-complexity of memory in Vector-HaSH

We show in SI Sec. D.1 that the number of sensory cells can scale as fast or faster than the maximal scaffold capacity $\mathcal{O}(K^M)$. Thus, $N_s \gg N_h, N_g$, and thus the number of synapses in the model, # synapses $= 2N_h(N_s+N_g)+M*K^2$ is dominated by $2N_s*N_h$. Moreover, the number of hippocampal cells is asymptotically constant for large K at a fixed M (Fig. 2d *right*, SI Fig. S2), and thus the number of synapses scales as $\mathcal{O}(N_s)$. Further, the number of nodes, $N_s+N_h+N_g$ also scales as $\mathcal{O}(N_s)$. Since the number of patterns perfectly reconstructed N_h is constant (with respect to the asymptotic scaling of the number of synapses at a fixed M), the continuum in Vector-HaSH ranges from storing $\mathcal{O})(1)$ patterns with $\mathcal{O}(N_s) = \mathcal{O}(\text{\# synapses})$ each, up to storing $\mathcal{O}(N_s) = \mathcal{O}(\text{scaffold size})$ patterns with positive recovered information.

The memory storage requirement for Vector-HaSH is equal to the number of synapses in the model, which as noted above scales as $\mathcal{O}(N_sN_h)$. This permits storage of N_h patterns perfectly, consisting of a total of N_sN_h bits of information. Thus, as in Hopfield and Hopfield-like networks, the total information stored and recalled in Vector-HaSH scales as the number of synapses. The time complexity for perfect recovery of all $\mathcal{O}(N_sN_h) = \mathcal{O}(\text{\# synapses})$ bits of information scales as $\mathcal{O}(N_sN_h^2) = \mathcal{O}((\text{\# synapses})^{3/2}\sqrt{N_h/N_s}) \leq \mathcal{O}((\text{\# synapses})^{3/2})$. In contrast, for Hopfield and Hopfield-like networks, the time complexity for recovery of $\mathcal{O}(\text{\# synapses})$ scales as $\mathcal{O}((\text{\# synapses})^{3/2})$. Vector-HaSH thus has an asympototically faster time complexity than Hopfield-like networks for recovery for the same number of total bits of information (when normalized by the number of synapses in the model).

When the number of patterns stored is larger and scales with the number of sensory cells, cN_s for $0 < c \leq 1$, Vector-HaSH partially recovers the stored information (Fig. 3). In this regime, Vector-HaSH has additionally improved time and space complexity as compared with the number of synapses: a space requirement of only $\mathcal{O}(\text{\# synapses})$ and a time complexity of only $\mathcal{O}(cN_s^2N_h) = \mathcal{O}(c(\text{\# synapses})^{3/2}\sqrt{N_s/N_h})$ is needed to store an input of $\mathcal{O}(c(\text{\# synapses}) \times (N_s/N_h))$ bits of information.

The above scalings can also be reinterpreted in terms of the total information, I, stored in the networks (where for perfect recovery $I = N_hN_s =$ # synapses, and for partial recovery $I = cN_s^2 = c \times (\text{\# synapses}) \times N_s/N_h$). For perfect recovery the space complexity requirements scale as I, and time complexity scales as $I^{3/2}\sqrt{N_h/N_s}$. For partial recovery the space complexity scales as $IN_h/(cN_s)$, and time complexity scales as $I^{3/2}N_h/(N_s\sqrt{c})$.

D.5 One-Step Heteroassociation Leads to a Memory Continuum even with Hebbian Learning

The memory continuum in Vector-HaSH is a result of the one-step heteroassociation from the hidden to the feature layer, given a memory scaffold that perfectly recovers the hidden states. This holds irrespective of the nature of heteroassociation (pseudoinverse learning or Hebbian learning).

Here we consider a simpler scenario where W_{sh} is trained through Hebbian learning and the hippocampal states are assumed to be correctly reconstructed (corresponding to pseudoinverse learning from S to H).

We assume here that the sensory patterns being stored are random unbiased binary vectors drawn uniformly from $\{-1,1\}^{N_s}$. If the weights from the hippocampal cells to the sensory inputs, W_{sh} are trained using Hebbian learning:

$$W_{sh} = (1/N_h)\sum_{\mu} s_i^{\mu} h_j^{\mu}. \tag{97}$$

To evaluate the accuracy of a recovered sensory state through this weight matrix, we estimate the probability that the i^{th} bit of s^{ν} is recovered correctly. Since we assume in this simplified scenario that the hippocampal state p^{ν} has been correctly recovered:

$$\begin{aligned} s_i(t+1) &= \operatorname{sgn}\left[\frac{1}{N_h}\sum_{j=1}^{N_h}\sum_{\mu=1}^{N_{patts}} s_i^{\mu} h_j^{\mu} h_j^{\nu}\right], \\ &= \operatorname{sgn}\left[s_i^{\nu}\left(\frac{1}{N_h}\sum_{j=1}^{N_h} h_j^{\nu} h_j^{\nu}\right) + \frac{1}{N_h}\sum_{\mu\neq\nu}\sum_j s_i^{\mu} h_j^{\mu} h_j^{\nu}\right]. \end{aligned} \tag{98}$$

Here we have separated the pattern ν from all the other patterns. Next, we multiply the second term on the right-hand side by a factor $f_i^{\nu} f_i^{\nu} = 1$, and pull f_i^{ν} out of the argument of the sign-function since $f_i^{\nu} = \pm 1$.

$$s_i(t+1) = s_i^{\nu}\operatorname{sgn}\left[1 + \frac{1}{N_h}\sum_{\mu\neq\nu}\sum_j s_i^{\mu} s_i^{\nu} h_j^{\mu} h_j^{\nu}\right] = s_i^{\nu}\operatorname{sgn}[1 - a_{i\nu}], \tag{99}$$

where

$$a_{i\nu} = -\frac{1}{N_h}\sum_{\mu\neq\nu}\sum_j s_i^{\mu} s_i^{\nu} h_j^{\mu} h_j^{\nu} \tag{100}$$

Successful recovery of the i^{th} bit of s_{ν} will occur if $s_i(t+1) = s_i^{\nu}$, which holds provided that $a_{i\nu} > 1$. The probability of an error in a given randomly chosen bit can thus be calculated as the probabilty that $a_{i\nu} > 1$.

Since the sensory states were assumed to have been drawn uniformly from $\{-1,1\}^{N_s}$, we can treat the product $s_i^{\mu} s_i^{\nu}$ as being $+1$ or -1 with equal probability. We assume that the distribution of hippocampal cell activity has mean μ_p and variance σ_p^2. Further, assuming that h_j^{μ} and h_j^{ν} can be treated as independent random variables, the product $h_j^{\mu} h_j^{\nu}$ would then have mean $\tilde{\mu} = \mu_h^2$ and variance $\tilde{\sigma}^2 = \sigma_h^4 + 2\sigma_h^2\mu_h^2$. Accounting for the random sign introduced by $s_i^{\mu} s_i^{\nu}$, the summand in Eq. (100) can then be treated as a random variable X with mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$ with probability 0.5, and mean $-\tilde{\mu}$ and variance $\tilde{\sigma}^2$ with probability 0.5. For large N_h and N_{patts}, we are summing over a large number of random variables — thus due to the Central Limit theorem, the precise details of the distribution will not matter, apart from an estimate of its mean and variance. By symmetry, the mean of X will be zero. This variance can be calculated to be $\tilde{\sigma}^2 + \tilde{\mu}^2$.

Summing over $(N_{patts} - 1)N_h \approx N_{patts}N_h$ terms in Eq. (100), evaluated through central limit theorem, thus gives a normal distribution, with zero mean, and variance $(\tilde{\sigma}^2 + \tilde{\mu}^2)N_{patts}N_h$. This gives

$$a_{i\nu} \sim \mathcal{N}\left(0, \sigma^2 = \frac{N_{patts}}{N_h}(\tilde{\sigma}^2 + \tilde{\mu}^2)\right), \tag{101}$$

with

$$\tilde{\sigma}^2 = \sigma_p^4 + 2\sigma_p^2\mu_p^2, \tag{102}$$

$$\tilde{\mu} = \mu_p^2. \tag{103}$$

$$\tag{104}$$

The probability of error in the activity state of neuron i is therefore given by:

$$P_{error} = \frac{1}{\sqrt{2\pi}\sigma}\int_1^{\infty} e^{\frac{-x^2}{2\sigma^2}}\,dx = \frac{1}{2}\left[1 - \operatorname{erf}\left(\sqrt{\frac{N_h}{2N_{patts}(\tilde{\sigma}^2 + \tilde{\mu}^2)}}\right)\right]. \tag{105}$$

Thus the probability of error increases with the ratio N_{patts}/N_h. The mutual information between the stored and recovered Sensory States is then:

$$MI_{perinbit} = 1 + P_{error} \log P_{error} + (1 - P_{error}) \log(1 - P_{error}). \tag{106}$$

Since $\mathrm{erf}(x) \approx x$ and $log(1+x) \approx x$ for small x, in the limit of a large N_{patts} the above expression can be approximated to

$$MI_{perinbit} \approx \frac{N_h}{2N_{patts}(\tilde{\sigma}^2 + \tilde{\mu}^2)} \propto \frac{N_h}{N_{patts}}. \tag{107}$$

This asymptotic scaling is verified in Fig. S6

D.6 Pseudoinverse learning of mappings from sensory states to velocities and memory palace items

Similar to pseudoinverse learning done from hippocampal cells to sensory cells, an exactly equivalent mathematical theory applies for pseudoinverse learning from sensory cells to the one-hot representation of velocities associated with each sensory state in 6 (see *Methods* for details of the learned velocity representations). In particular, as seen in Sec. D.2, pseudoinverse learning of a matrix W_{yx} that maps from a layer X of dimensionality N_x to a layer Y of dimensionality N_y is successful in exact recovery of all patterns in layer Y when learning up to $\mathrm{rank}(X)$ patterns. In the case that $\mathrm{rank}(X) = N_x$, as is the case when learning mappings from S to P and vice-versa (Sec. D.1, D.2), the number of learned patterns that can be perfectly reconstucted is simply N_x.

Thus, learning mappings from sensory cells to either velocity representations or memory palace task items will be exactly successful for up to N_s velocities of memory palace items provided that the mappings are being learned from patterns that form a full rank matrix. However, as seen in Fig. 6, mappings must be learned from the reconstructed sensory states rater than the ground truth sensory states, since with increasing number of stored patterns the reconstructed states deviate from the ground truth states.

Thus, even if the sensory states form a full rank matrix, for successful mappings, it will be necessary that the rank of the recovered sensory states must be N_s. Following the results presented in Sec. D.3, it would appear that the the recovered sensory patterns would form a matrix of rank N_h, the dimensionality of the hyperplane $\mathcal{P}_H$. However, this would only be the case if the recovered sensory states were obtained directly from $W_{sh}H$ wihtout any additional nonlinearity. For the case of binary sensory states, the recovered sensory patterns are given by $\mathrm{sgn}[W_{sh}H]$. This sign nonlinearity in effect behaves like a small random perturbation to each of the N_s bits of $W_{sh}H$, rendering the reconstructed sensory states to be full rank (assuming that the ground-truth sensory states matrix is full rank). Thus, reconstruction of velocity mappings and memory palace task items (or indeed any other readout from sensory the sensory cells) will be successful for up to N_s patterns, provided that the mappings are being learned from the reconstructed sensory states (rather than the ground truth sensory states).

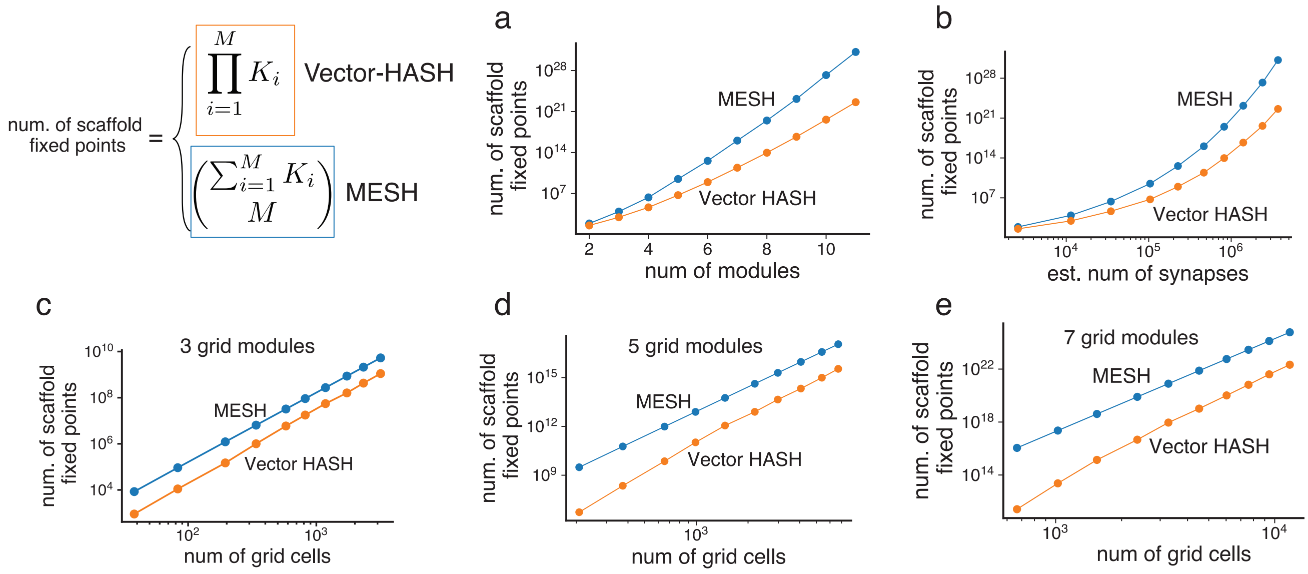

Figure S1. **Theoretical capacity results in Vector-HaSH relative to MESH.** The number of scaffold fixed points increases exponentially in the number of modules (a), and faster than a power law, but slower than exponentially in the number of synapses (b). The number of synapses with increasing number of modules were estimated based on a number of hippocampal cells extrapolated from Fig. 2f. The number of fixed points increases as a power law with the number of grid cells at a fixed number of modules, with the power law exponent increasing with the number of modules.

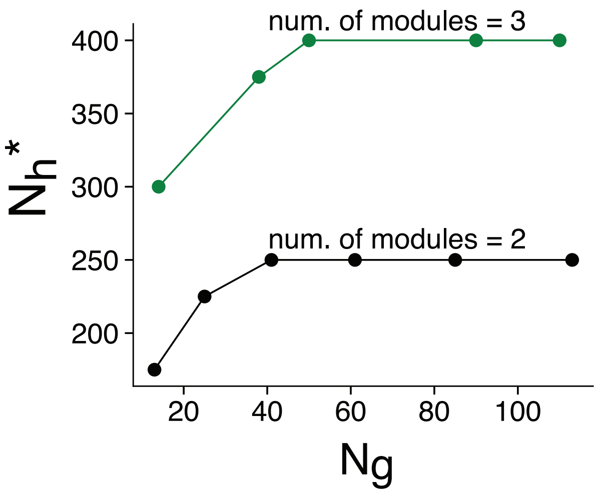

Figure S2. **Critical number of hippocampal cells necessary to support all scaffold fixed points is asymptotically independent of the number of grid cells** For a given number of modules, the critical number of hippocampal cells, N_h^* increases slowly with the number of grid cells, but then asymptotically approaches a constant, as expected from the theoretical results in Sec. C.1.

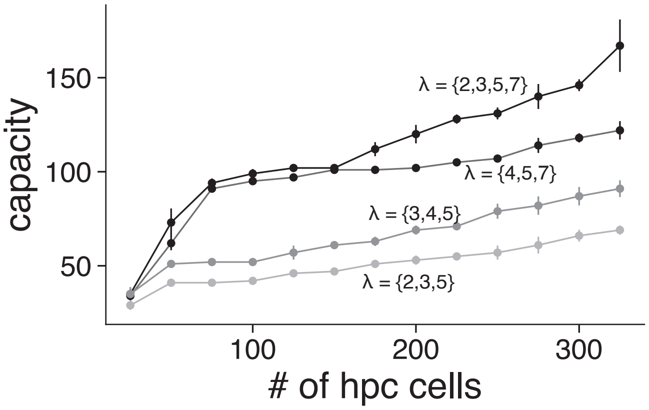

Figure S3. Scaffold constructed with bi-directional learning between grid states and sparse hippocampal states has low capacity As seen in Fig. 2f *inset*, construction of random sparse hippocampal states with bidirectional learning between grid and hippocampal states results in a scaffold that exibhits catastrophic forgetting. We calculate the capacity of the network as the largest number of trained patterns such that all trained patterns are stored as fixed points. Note that this capacity is limited by the number of hippocampal cells, as might be expected from Hopfield like capacity bounds.

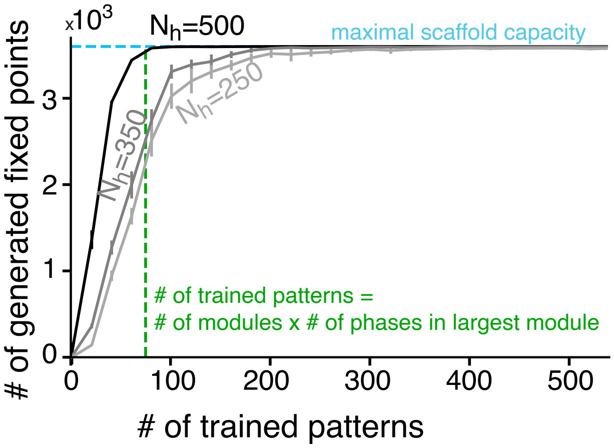

Figure S4. Learning generalization approaches theoretical expectations with increasing N_h The number of generated fixed points approaches the maximal scaffold capacity for a very small number of learned patterns (see also Fig. 2f). As the number of hippocampal cells increases, the number of learning patterns necessary for complete generalization approaches the theoretical expectaion of $M \times K_{max}$, as proved in SI Sec. C.4.

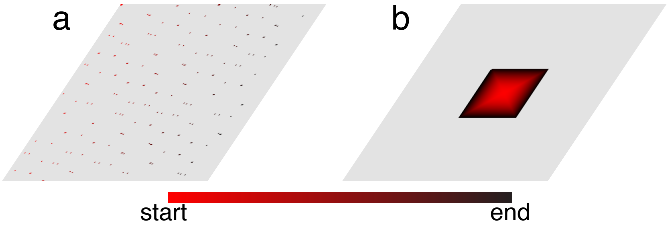

Figure S5. Minimum learning region for generalization of fixed point dynamics at all grid patterns As seen in Figs. 2f, S4 all the exponentially many scaffold states are stabilized after learning from only a small number of grid patterns. Here we show visually the minimum learning region that results in complete generalization to all scaffold fixed points. (a) shows the minimal learning region for the fastest possible generalization to all scaffold states (see SI Sec. C.4 for an analytic proof), (b) shows the smallest region needed for a path that spans a two-dimensional contiguous region, generated by a spiraling outward path. Both (a) and (b) are shown corresponding to a scaffold size of 44100, generated with $\lambda = \{2,3,5,7\}$. As argued in SI Sec. C.4, the minimum learning area as a fraction for complete generalization approaches zero with increasing scaffold sizes.

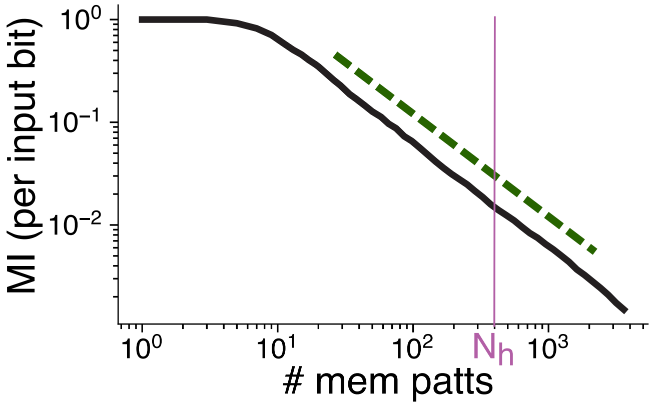

Figure S6. Hebbian learning between sensory layer and scaffold also produces memory continuum A memory continuum is obtained in Vector-HaSH even if the weights between the sensory and hippocampal layers are bi-directionally trained using Hebbian learning (instead of pseudoinverse learning, as in Fig. 3. This continuum is also asymptotically proportional to the theoretical bound on memory capacity (forest green dashed line indicative of slope of theoretical upper bound, vertical and horizontal position of dashed line is arbitrary). However, the proportionality constant is lower, with the gradual degradation of information recall occurring well before N_h. Vector-HaSH parameters identical to Fig. 3c with $\lambda = \{3,4,5\}$.

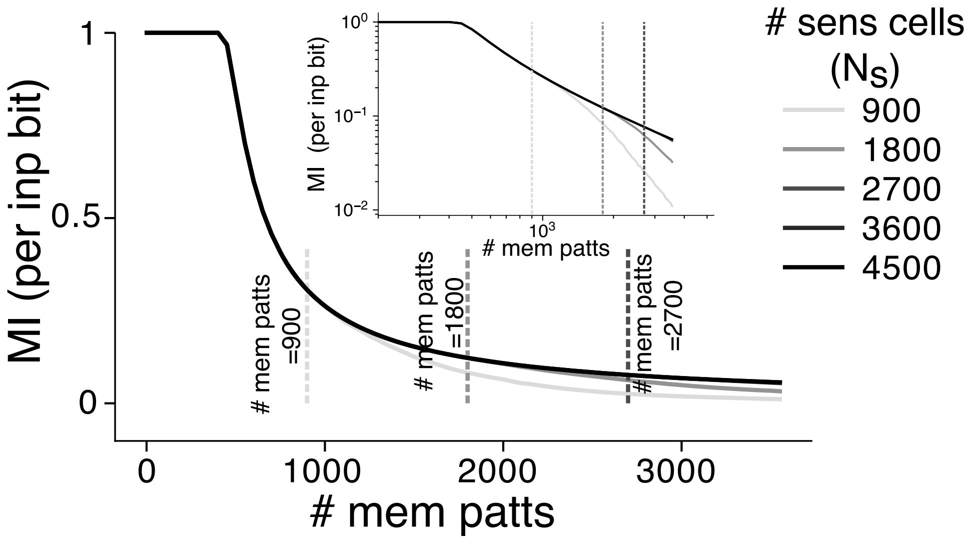

Figure S7. Effect of varying N_s on memory continuum As shown in SI Sec. D.1, the number of sensory cells determines the number of scaffold states that can be exactly recovered through the sensory-to-hippocampal weights. For N_s less than the total number of scaffold states, the obtained memory continuum is distorted towards the tail for larger than N_s patterns stored. For all N_s larger than or equal to the number of scaffold states, the memory continuum is identical, corresponding to the results shown in Fig. 3.

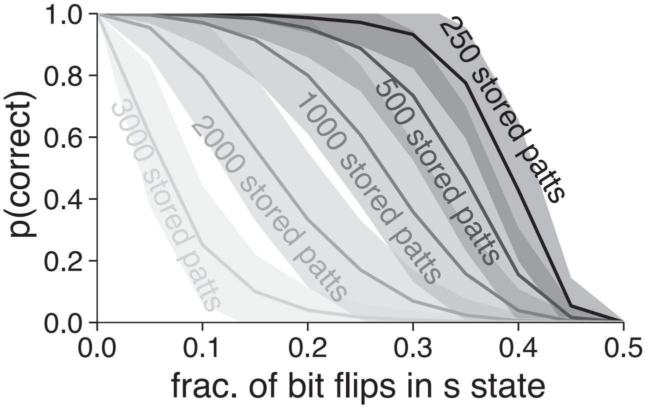

Figure S8. Basin structure for recovery of sensory hippocampal and grid states varies with number of stored patterns While the scaffold has a large number of well-strucutred basins (cf. Fig. 2, SI Sec. C.1), the basins for sensory recovery are additionally governed by the heteroassociative leanring between the sensory states and the scaffold. As a result, the basin sizes reduce with increasingly large number of stored patterns, due to overcrowding of the number of stored states within the sensory-to-hippocampal weights. The grid periods were set to $\lambda = \{3,4,5\}$, with $N_h = 400$, resulting in a maximal scaffold capacity of 3600 patterns, with perfect sensory recovery up to 400 patterns. For more than 400 stored patterns, $p(correct)$ refers to the probability of exact recovery of grid and hippocampal states, and probability of reliable recovery of the sensory state (which is not exact due to being in the memory continuum regime, Fig. 3.

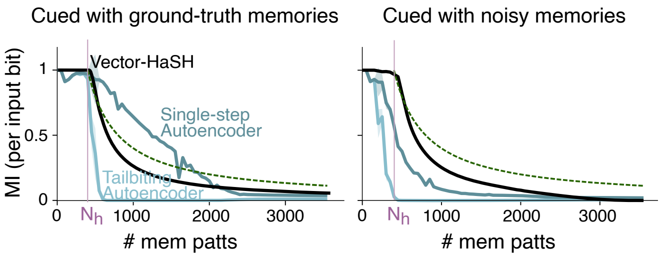

Figure S9. Vector-HaSH outperforms Autoencoders, particularly when recovering from noisy cues *Left* When cued with ground-truth memorized sensory patterns, Vector-HaSH recovers a gradually degraded amount of information per pattern (cf. Fig. 3), unlike the memory cliff shown by tailbiting Autoencoders trained as associative memories[86]. Naively however, it appears that this memory cliff is absent in a single step (i.e., non-tailbiting) of the Autoencoder. However, we see in *Right* that single-step Autoencoders are not associative memories, since they are unable to reconstruct memories from corrupted cues. Here grid periods were set to $\lambda = \{3,4,5\}$, with $N_h = 400$. Stored sensory cues were random binary $\{-1,1\}$ patterns, and noisy cues were generated by flipping 10% of bits from a given memorized sensory pattern.

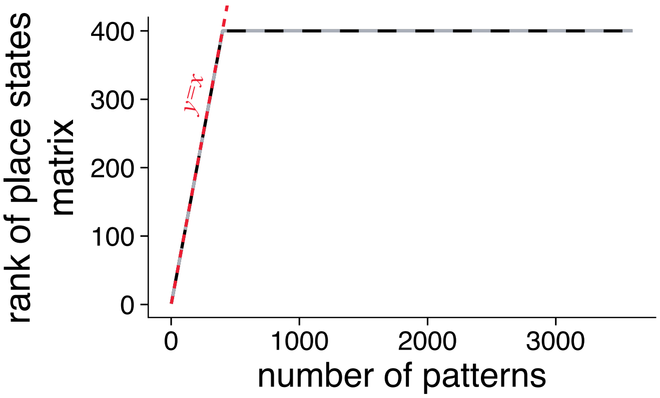

Figure S10. hippocampal states form a strongly full rank matrix Rank of the $N_h \times N_{patts}$ hippocampal states matrix for varying number of patterns N_{patts} for two different random permutations of the ordering of hippocampal states shown in black and grey. For up to $N_{patts} \leq N_h$ the rank of the matrix is N_{patts} (as indicated by the red $y = x$ line), and is there after N_h for larger numbers of patterns.

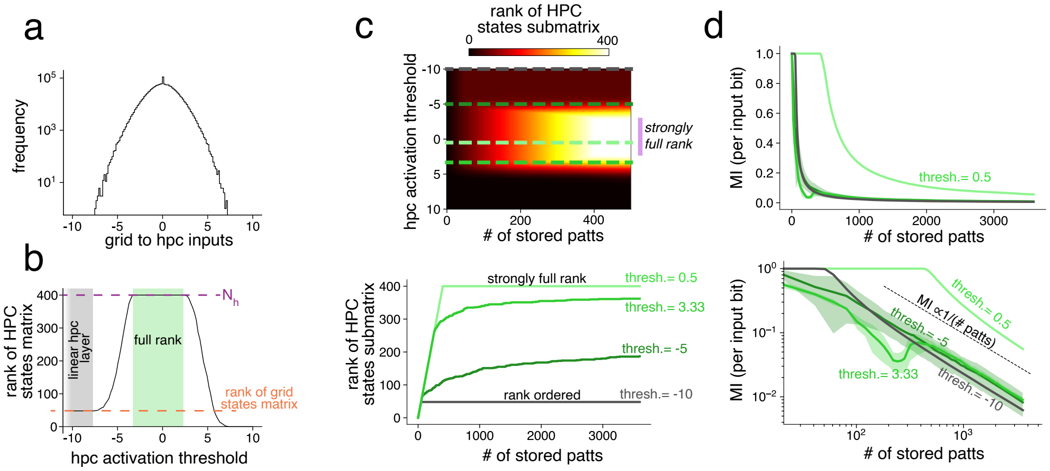

Figure S11. Activation threshold applied in the hippocampal layer dictates nature of memory continuum (a) distribution of pre-nonlinearity inputs to the hippocampal layer from grid cells. Any actication threshold above the largest value (~ 7.5) results in zero hippocampal activity, and any threshold below the smallest value (~ -7.5) results in a purely linear hippocampal layer. (b) A linear hippocampal layer (corresponding to thresholds in the grey region) results in a HPC states matrix of rank equal to the rank of the grid cell states matrix (which equals $N_g - M + 1$ as shown in Ref.[136]), whereas a range of thresholds (shown in green) result in a full rank HPC states matrix. (c)*Top*: The rank of the $N_h \times N_{patts}$ submatrix of hippocampal states constructed over the first N_{patts}. Here the hippocampal states have been ordered according to the optimal order that leads to fastest scaffold learning generalization (Sec. C.4). *Bottom*: Rank versus N_{patts} for the particular values of thresholds considered in panel (d). At a threshold of 0.5 (the value used in almost all simulations in the main text, see *Methods* for more details) we see that the hpc states matrix is strongly full rank. Moreover, as seen in Fig. S10, this matrix is strongly full rank independent of the ordering of the scaffold states. At the lowest threshold value, corresponding to a linear hippocampal layer, the matrix appears to be rank ordered. However, for a linear hippocampal layer the rank ordering of the matrix is dependent on the ordering of the scaffold states, as examined in Fig. S12. (d) Information recovered per input bit as a function of the number of patterns stored in the network (similar to Fig. 3d) for varying threshold values on a linear scale (*top*) and a logarithmic scale (*bottom*). The strongly full rank matrix (identical the the $\lambda = \{3, 4, 5\}$ curve in Fig. 3d) and the rank ordered matrix both demonstrate perfect recovery up to a knee; all values of thresholds result in a smooth decay of recovered information that is asymptotically proportional to a theoretically expected bound that scales inversely with the number of stored patterns

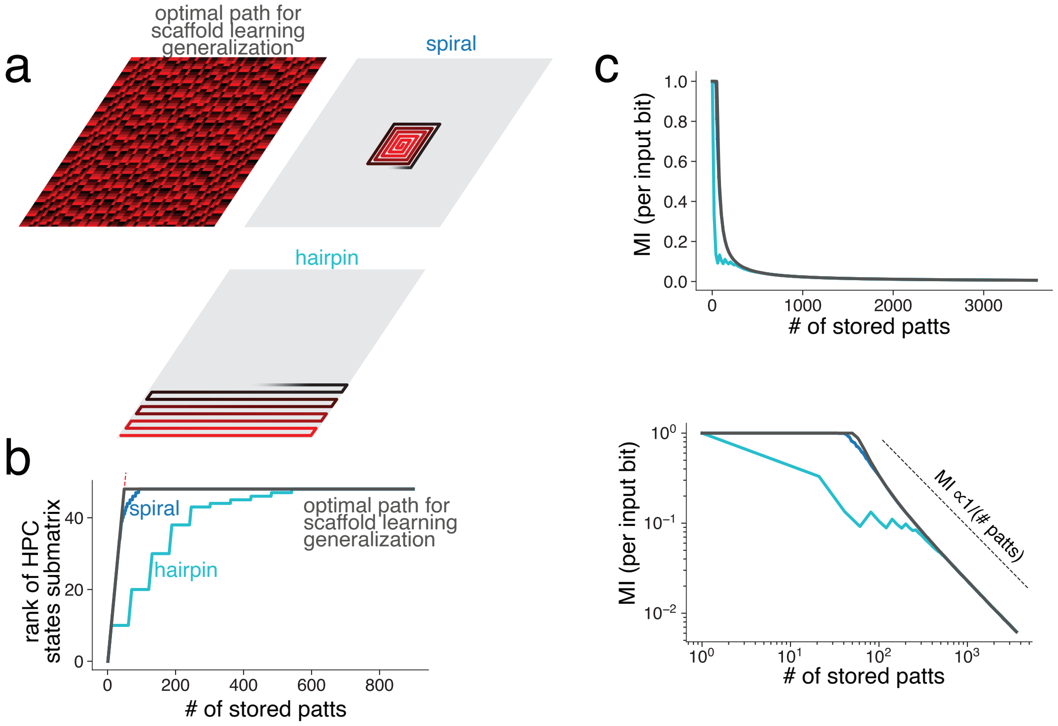

Figure S12. Linear hippocampal layer generates memory continuum only for specific ordering of scaffold states during learning (a) Three examples of potential ordering of scaffold states that could be considered: *top left* the discontinuous path that leads to fastest scaffold learning generalization (Sec. C.4) shown for $\lambda = \{3,4,5\}$; *top right* a continuous spiral path; *bottom* a continuous 'hairpin' path. (b) The hpc states submatrix is rank ordered along the optimal path, and approximately rank ordered for the continuous path. The hairpin path however is significantly deviated from a rank ordered matrix. (c) Information recovered per input bit as a function of the number of patterns stored in the network (similar to Fig. 3d). The rank ordered matrix demonstrates perfect recoverey up to a knee at the rank of the grid states matrix; this is also closely approximated by the spiral ordered matrix. A hairpin ordering however results in poor information recovery even at a small number of patterns. In all cases, the asymptotic decay of information is inversely proportional to the number of patterns, as would be expected from theoretical information bounds.

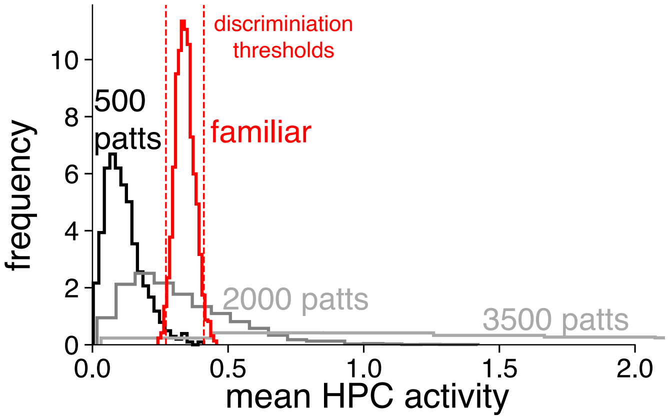

Figure S13. **Mean activity in hippocampal layer can be used for novelty detection** The mean activity in the hippocampal layer for familiar patterns presents a narrow distribution. The mean hippocampal activity for novel patterns is strongly dependent on the number of stored patterns. The narrowness of the familar pattern distribution allows for discrimination thresholds to be placed on either side (at two standard deviations away from the mean) to result in classification accuracy as shown in Fig. 3i

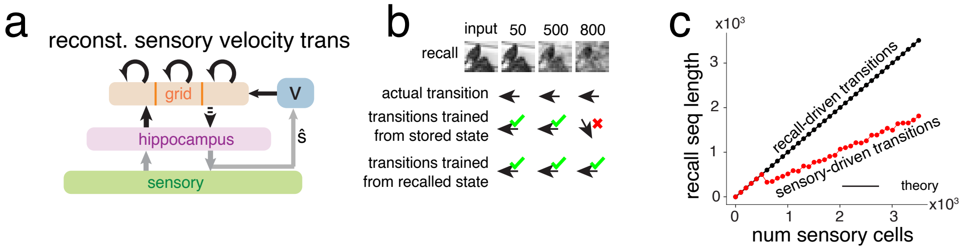

Figure S14. **Sensory driven transitions must be reconstructed from recalled states** (a) Architecture for sensory based reconstruction of next-step transitions for sequence learning, Fig. 6a, *bottom*. (b) As the number of stored patterns increases, the recalled sensory state gradually degrades; as a result, reconstruction from mapping trained on ground truth sensory states can lead to inaccuracies. (c) More quantitatively, transitions trained on recalled sensory states result in sequence reconstruction of length up to the number of sensory cells (theory in SI Sec. D.6), whereas transitions trained on ground truth sensory states has a lower sequence capacity.

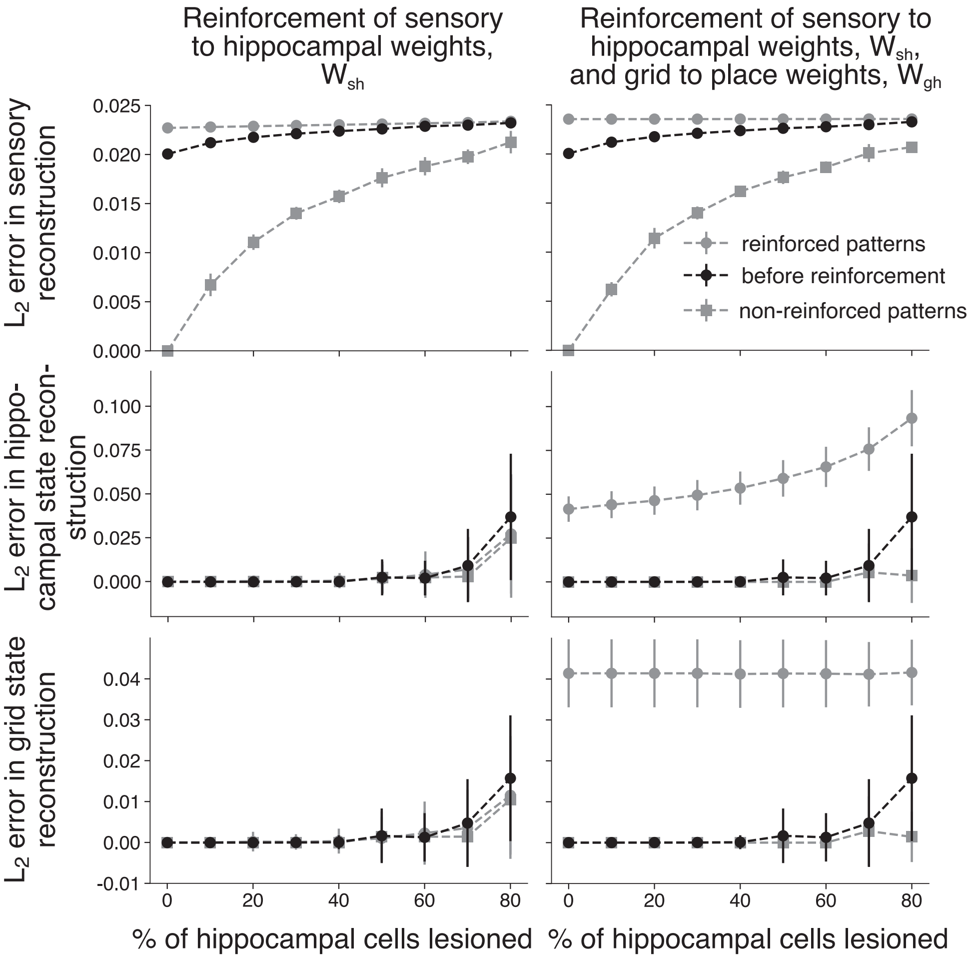

Figure S15. **Reconstruction error in each layer of Vector-HaSH when tested for MTT by reinforcing the model weights for a subset of repeated patterns.** Left: Results when only W_{sh} weights are reinforced, assuming pre-trained scaffold weights W_{gh}. Right: Results when all of the learnable weights in Vector-HaSH W_{hs}, W_{sh} and W_{gh} are reinforced. Note that W_{hs} reinforcement mathematically doesn't change W_{hs} as describe in Sec. 1.

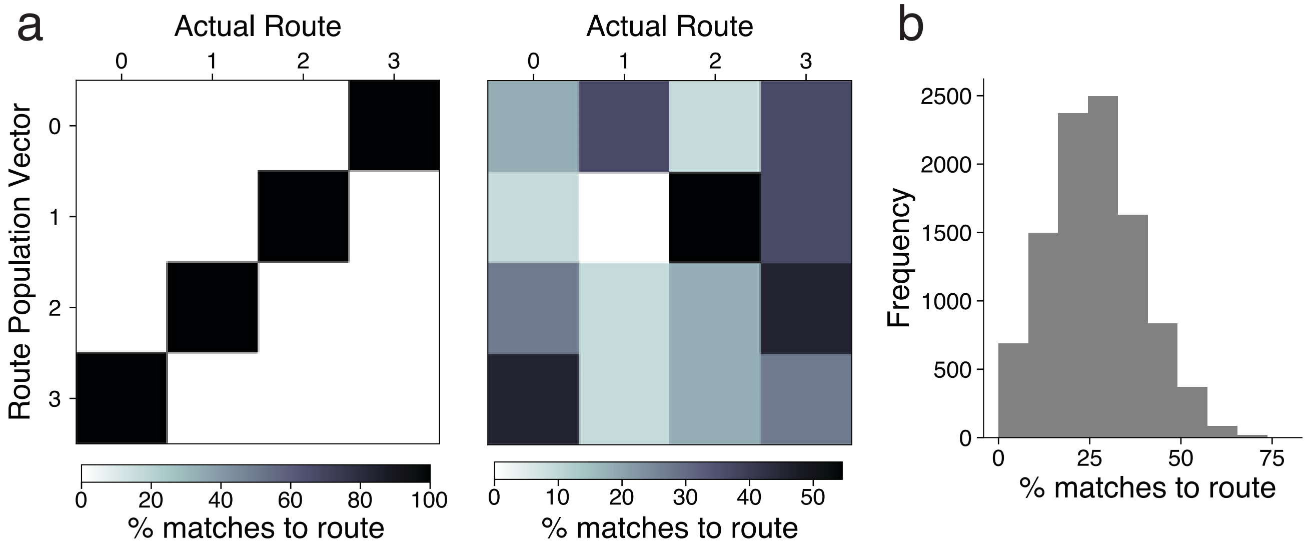

Figure S16. **Analysis method for route encoding results in Fig. 7b.** (a) Left: Trajectory population vectors (PVs) were compared to route-PVs and matched according to highest cosine similarity score. Elements of the matrix show the percentage number of each trajectory matched to each of the four route-PVs. Right: Matches were also made using shuffled data, where each trajectory was randomly assigned to one of the four routes, thus shuffling the route identity of the trajectories. The matrix elements here show the same as *a* except that this data is for one representative shuffle (10000 were conducted in total). (b) Distribution of percentage correct matches for trajectory PV of Route 3 to its route-PV for all 10000 shuffles.

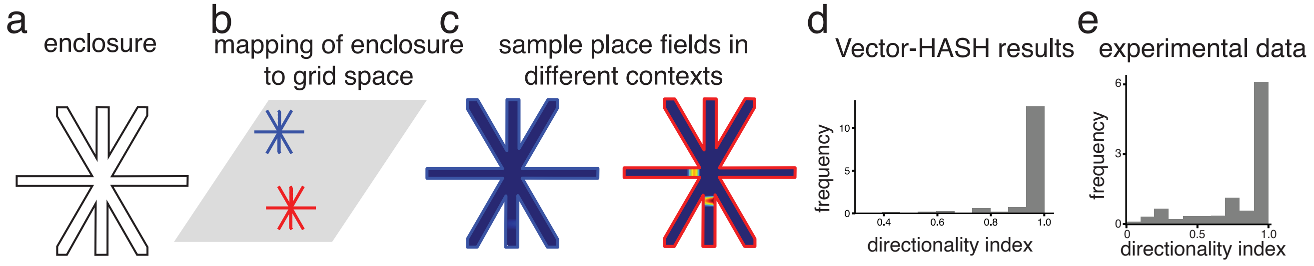

Figure S17. **Vector-HaSH reproduces directional place fields on an 8-arm radial maze.** (a) An 8-arm radial maze apparatus similar to the experiment[113]. (b) Inbound (towards the center) and outbound (away from the center) trajectories on the 8-arm radial maze represented separately in the grid coding space. (c) Fields of a representative hippocampalcell on inbound trajectories (left) and outbound trajectories (right) (d) Directionality index of place cells from Vector-HaSH showing that majority of the cells have directional fields. (e) Directionality index of place cells from the experimental data[113].

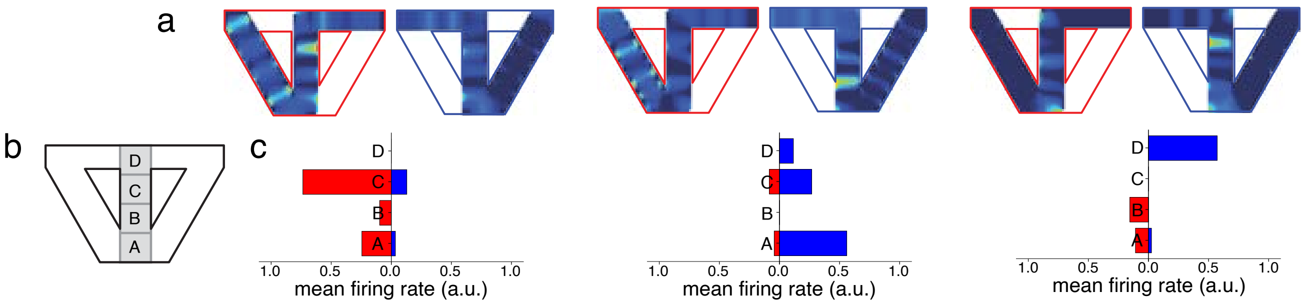

Figure S18. Splitter cells. (a) Fields of three representative hippocampal cells on the Right-Turn and Left-Turn trials. (b) The central stem of the continuous alternation task apparatus is divded into 4 equal regions for data analysis following the analysis conducted on the experimental data[110]. (c) Mean activation of the three hippocampal cells shown in *(a)* computed for each of the four regions defined in *(b)*. The cells show different activity patterns as Vector-HaSH traverses the central stem on Left-Turn and Right-Turn trials.

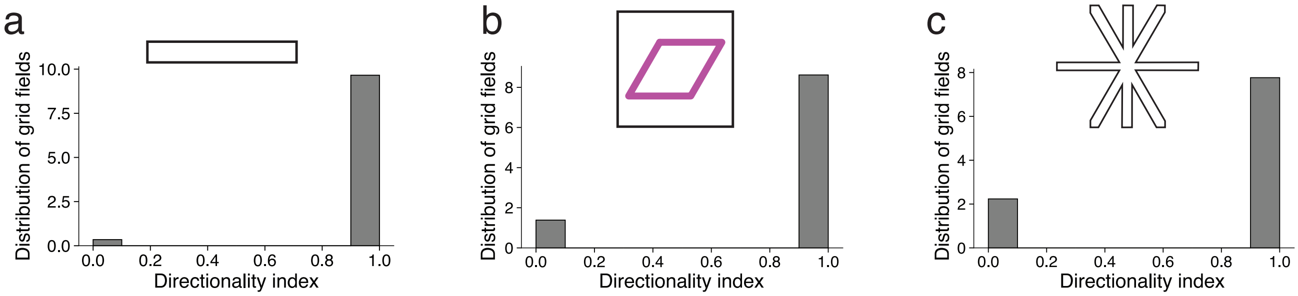

Figure S19. Vector-HaSH predicts directional grid fields. Directionality index of grid cells showing that in Vector-HaSH majority of the grid cells have directional fields in one dimensional environments (environment in (a)[112]), and on directed routes in two dimensional environments (environments in b,c[113]).