Episodic and associative memory from spatial scaffolds in the hippocampus

https://doi.org/10.1038/s41586-024-08392-y

Received: 28 November 2023

Accepted: 13 November 2024

Published online: 15 January 2025

Check for updates

Sarthak Chandra[1,4], Sugandha Sharma[1,4], Rishidev Chaudhuri[2,3] & Ila Fiete[1]✉

Hippocampal circuits in the brain enable two distinct cognitive functions: the construction of spatial maps for navigation, and the storage of sequential episodic memories[1–5]. Although there have been advances in modelling spatial representations in the hippocampus[6–10], we lack good models of its role in episodic memory. Here we present a neocortical–entorhinal–hippocampal network model that implements a high-capacity general associative memory, spatial memory and episodic memory. By factoring content storage from the dynamics of generating error-correcting stable states, the circuit (which we call vector hippocampal scaffolded heteroassociative memory (Vector-HaSH)) avoids the memory cliff of prior memory models[11,12], and instead exhibits a graceful trade-off between number of stored items and recall detail. A pre-structured internal scaffold based on grid cell states is essential for constructing even non-spatial episodic memory: it enables high-capacity sequence memorization by abstracting the chaining problem into one of learning low-dimensional transitions. Vector-HaSH reproduces several hippocampal experiments on spatial mapping and context-based representations, and provides a circuit model of the 'memory palaces' used by memory athletes[13]. Thus, this work provides a unified understanding of the spatial mapping and associative and episodic memory roles of the hippocampus.

As we navigate through life, the hippocampus weaves threads of experience into a fabric of memory cross-linked by context. Thus, we can revisit scenes and events from only a few cues, as with Proust's famous madeleine[14]. Such memories enable inferences in the present and planning for the future. The hippocampal complex is responsible for this functionality[1,15–18], but it is unclear exactly how the architecture and representations of the hippocampus and adjoining cortical regions enable it.

Substructures of the hippocampal complex have been studied extensively[2–5,19–24], and experimental findings combined with modelling have led to marked progress in understanding local circuit mechanisms[8–10,25–45]. These works put us in an excellent position to build our understanding of how the combined system subserves memory storage and recall. A central question involves the dual role of this structure. The ability to form episodic memories, our catalogue of autobiographical experiences, is compromised by damage to the hippocampal complex[1]. Spatial memory—remembering the layout of our physical environment and our updated position within it as we move about—also centrally involves the hippocampus. Place cells fire at specific locations and environments[2], and entorhinal grid cells represent spatial displacements in the form of triangular grid firing patterns that repeat across environments[3]. It remains unknown why these two forms of memory are co-localized.

One hypothesis is that the circuit prioritizes spatial memory, such as where certain foods and dangers were found. In this view, episodic memory is an augmentation of the spatial memory system and representations are optimal for spatial, not episodic, memory. The second hypothesis is that the circuit is optimal for episodic memory, with spatial coordinates represented primarily as a stable and useful index for episodic memory[46]. The third hypothesis is that the circuit does not simply store spatial and episodic information side by side or with one in the service of the other, but that its highly structured architectures, representations and dynamics are equally optimal for both functions, even when the memory in question does not involve space. Thus, the low-dimensional states in the circuit that are interpreted as spatial may serve as equally critical scaffolds for linking together (potentially entirely non-spatial) elements of an episodic memory[18,45–50].

We propose a neocortical–entorhinal–hippocampal memory model based on properties of the biological circuit. It excels at item memory, spatial memory and sequential episodic memory with content-addressable recall, avoiding the full erasure (memory cliff) of existing neural memory models when adding inputs beyond a fixed low capacity (Fig. 1a). Critically, the memory model: (1) separates dynamical fixed point generation (for pattern completion and error correction) from content encoding, with the former exploiting the structure of grid cell states; and (2) converts the problem of sequence memory into a simple process of low-dimensional transitions on the grid states. Grid cells are thus equally essential for non-spatial memory, supporting the third hypothesis.

This circuit, which we call Vector-HaSH because it assigns an error-correcting hash code to each input (a hash is a unique label

[1]Department of Brain and Cognitive Sciences and McGovern Institute, MIT, Cambridge, MA, USA. [2]Center for Neuroscience, Department of Neurobiology, Physiology and Behavior, University of California Davis, Davis, CA, USA. [3]Department of Mathematics, University of California Davis, Davis, CA, USA. [4]These authors contributed equally: Sarthak Chandra, Sugandha Sharma. ✉e-mail: fiete@mit.edu

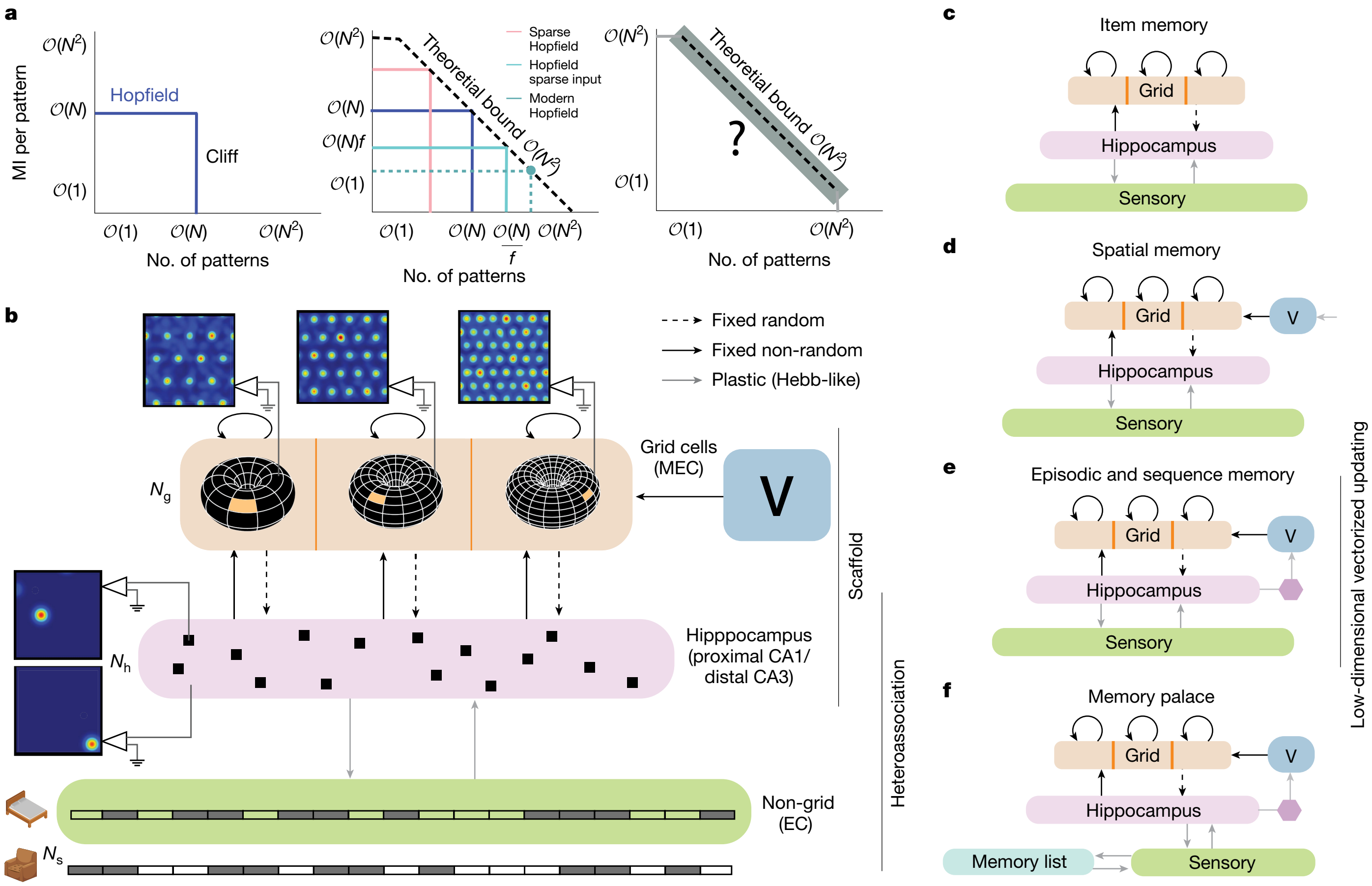


Fig. 1 | Biological memory: challenges and proposed architecture. **a**, Left, memory cliff. Mutual information (MI) between stored and retrieved patterns in a Hopfield network of N neurons drops sharply to zero after a critical number (approximately N) of stored patterns. Middle and right, MI in variants of Hopfield networks (sparse synapses; input patterns with sparseness p, $f = p\ln(p)$; modern Hopfield networks) all exhibit a cliff. Dashed black line indicates theoretical limit for neural networks with N^2 plastic synapses[11,12]. We explore whether it is possible for content-addressable memory to hug the bound, regardless of the number of stored patterns. **b**, Vector-HaSH architecture. Grid cell modules (orange) with fixed recurrent connectivity and states, hippocampal cells (pink) and non-grid entorhinal/neocortical (EC) processed sensory inputs (green). Solid and dashed black arrows indicate fixed connections and weights; grey arrows indicate plastic weights. N_g is the number of grid cells, N_h is the number of hippocampal neurons and N_s is the number of sensory input neurons. The light blue box represents a low-dimensional velocity 'shift mechanism' that changes the grid phases. **c**–**f**, Vector-HaSH variants that model item memory (**c**), spatial memory (**d**), episodic memory (**e**) and the memory palace technique (**f**).

independent of content) and exploits the metric structure of grid cell states to enable sequence storage via low-dimensional vector transitions.

Factorization of dynamics and content

Hopfield networks are the paradigmatic model of content-addressable neural network memory[25]. Structured as monolithic recurrent networks, they exhibit a steep memory cliff: in an N-neuron network, up to approximately N patterns (of N bits each) are perfectly recovered, but adding more patterns leads to total loss of even previously memorized patterns (Fig. 1a). Variants of Hopfield networks all exhibit a cliff[11,12,51].

Vector-HaSH is based on (known and inferred) connectivity of the hippocampus and entorhinal cortex[8,52]. Entorhinal grid cells, which project to hippocampus (Fig. 1b, orange), consist of multiple modules[23] with distinct periods λ. Each module expresses a set of low-dimensional states that are stabilized by recurrent connections and are invariant to task and behavioural state[33,34,36–38]. In non-spatial contexts, these states can be conceptualized as abstract representations that are constrained to lie on a two-dimensional torus. Processed extrahippocampal inputs carrying sensory and internally generated states enter the hippocampus (Fig. 1b, purple) via non-grid entorhinal neurons and a few other cortical areas (Fig. 1b, green).

Critically, connections from grid cells to hippocampus are set as random and fixed. Hippocampus projects back to entorhinal cortex; those to grid cells are set once (for example, during pre- or post-natal development) by associative learning, then held fixed. Connections of hippocampus to non-grid cells remain bidirectionally plastic for memory acquisition and are set by associative learning. Activity propagation between regions occurs in sequential order and discrete time, a simplification of the oscillations and synaptic latencies hypothesized to gate this information flow. We refer to the grid–hippocampal subcircuit, with its unchanging weights, as the fixed scaffold of the network. Separately, we refer to the hippocampal–non-grid cortical feedback loop as the heteroassociative part of the circuit. We will see that small variations of this basic circuit enable content-addressable memory in various settings, from spatial to non-spatial memory to sequential episodic memory and memory palaces (Fig. 1c–f).

A vast library of robust fixed points

Grid cells are partitioned into a few (M) independent modules: a module (the ith module) can occupy one of K_i states, which lie on a two-dimensional torus. Together, the modules express $\prod_i K_i \approx \langle K \rangle^M$ many distinct states, growing exponentially with M. An across-module grid state, if bidirectionally coupled to a hippocampal cell that

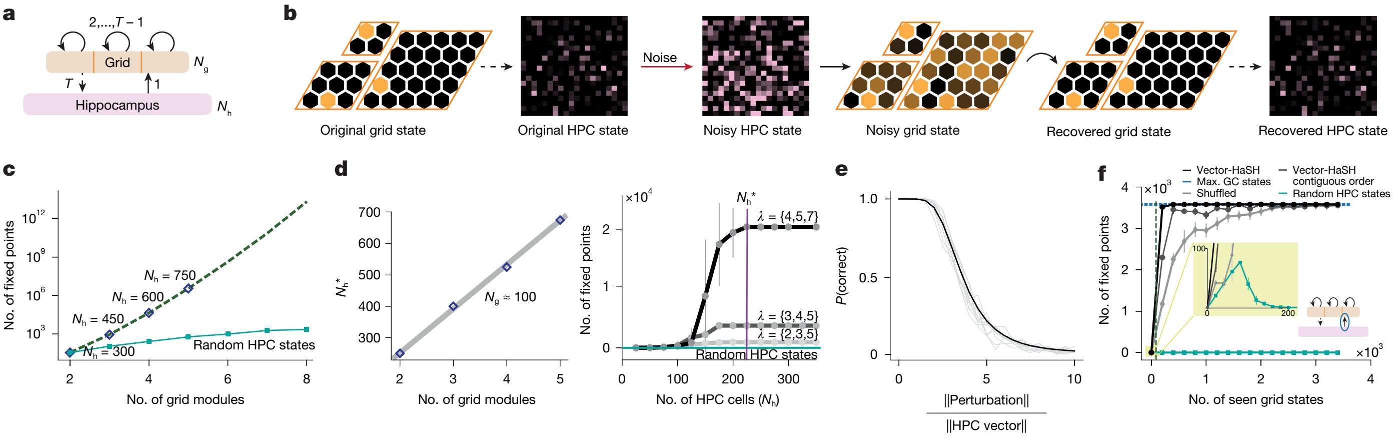


Fig. 2 | The scaffold generates exponentially many attractors with equally large basins. a, Scaffold architecture, as in Fig. 1. 1,...,T indicates the flow of dynamics (order of updating) in the circuit. **b**, Left two images, grid state and corresponding hippocampal (HPC) state. Other images: scaffold initialized with noisy version of hippocampal state from the left is cleaned up by hippocampal–grid, grid–grid and grid–hippocampal scaffold dynamics. **c**, Number of scaffold fixed points increases exponentially with number of grid modules[32,53], given enough hippocampal neurons. Diamonds are results of numerical simulation; dashed line represents an analytical curve (Supplementary Information, section C.1). Error bars represent s.d. over five runs (but are not visible). Light green line represents shuffled pairing of grid–hippocampal states, with grid–hippocampal weights bidirectionally learned. **d**, Left, minimal hippocampal size (N_h^*) to convert all grid states into fixed points is linear in number of grid modules. Right, N_h^* is nearly independent of the grid period λ. Error bars represent s.d. over five runs. **e**, Probability that noisy hippocampal state flows to correct attractor, versus noise magnitude (in multiples of mean hippocampal state norm). Grey curves show data for five different attractors. Black curve shows the average over 3,600 attractors. 100 random noise realizations per attractor. **f**, Strong generalization: number of formed scaffold attractors versus number of states used to associatively learn hippocampus-to-grid weights. Dark grey represents any contiguous sequence of grid states; black is an optimal sequence; light grey is a random sequence. Vertical dashed line represents the theoretical minimum number of states associatively learned to make all grid states into attractors. GC, grid cell; Max., maximum. Error bars represent s.d. over five runs.

couples the modules[30,32], can become a stable fixed point and enable error correction[32]. Because there are many more grid states than hippocampal neurons[53], this formulation cannot work for all grid states.

Remarkably, if grid cells project with random weights (a high-rank random projection) to hippocampus, and hippocampal cells (after thresholding their inputs) send return projections that are learned through Hebb-like learning to reinforce the corresponding grid states (Methods), then all grid states become stable fixed points (attractors) of the grid–hippocampal scaffold network (Fig. 2b,c), once the hippocampus is above a minimal size (N_h^*). We define a state as an attractor if it is restored to its non-noisy value through the scaffold dynamics after injection of noise of norm 0.25 times the average hippocampal state norm.

The minimum hippocampal size (N_h^*) to convert all grid states to attractors is small (Fig. 2c)—it scales only linearly with the number of grid modules (Fig. 2d, left and Supplementary Fig. 1; proof in Supplementary Information, section C.1) and is nearly independent of the scale (periodicity) of the grid modules (Fig. 2d, right and Extended Data Fig. 1; proof in Supplementary Information, section C.1). Thus, the number of attractors (approximately $\langle K\rangle^M$) is exponential in the total number of scaffold (grid and hippocampal) neurons ($N_h + M\langle K\rangle \approx M(c+K)$), where c is a constant. For example, $M = 10$, $K = 10^2$ (10 grid phases per dimension) would generate ~10^{20} scaffold attractors with only ~10^3 combined hippocampal and grid cells.

A correspondence between the grid states and grid-driven hippocampal states is critical: if the set of hippocampal states is randomly reassigned to the grid states, with bidirectional learning of grid–hippocampal weights for self-consistent activity reinforcement, the number of attractors collapses (Fig. 2c,d, light green curves; also see Supplementary Information Fig. 2). A similar collapse occurs if hippocampal-to-grid weights are randomly set while the grid-to-hippocampal weights are associatively learned for self-consistency (Supplementary Fig. 3).

We derive theoretically that the scaffold has no spurious fixed points or attractors, thus the scaffold attractor basins are maximally large (all hippocampal states form the scaffold attractor basins). The basins are also all convex and essentially identical (Supplementary Information, sections C.2 and C.3). We test these analytical results by numerically computing the probability that a noisy hippocampal state flows to the true attractor (Fig. 2e). Injected noise of magnitude several times the hippocampal state norm is reliably corrected, and all basins have identical probability curves.

Strong generalization property of scaffold

A key property of the scaffold is ‘strong generalization’: associatively learning the hippocampal-to-grid cell weights to stabilize the exponentially many (approximately $\mathcal{O}(K^M)$) grid states does not require visiting them all. Visiting and performing associative grid–hippocampal learning on a vanishingly small fraction of the states stabilizes them all (Fig. 2f). We show that visiting any $\mathcal{O}(MK_{\max})$ contiguous grid states, where $K_{\max}$ is the number of states in the largest module, is sufficient for stabilizing all (approximately $\langle K\rangle^M$) of them as scaffold fixed points (Extended Data Fig. 2 and Supplementary Information, section C.4); the results are robust to noise during learning (compare with Supplementary Fig. 5). Replacing grid states with random patterns of matched sparsity (for example, by shuffling within each multi-module grid-coding state), as in memory scaffold with heteroassociation (MESH)[54], results in near-total generalization loss (proof in Supplementary Information, section C.4), and visiting grid states in random order decreases the amount of generalization (Fig. 2f). Certain special sets of non-contiguous locations can also lead to strong generalization (Supplementary Information, section C.4 and Supplementary Fig. 4). Strong generalization implies that restricted spatial exploration by juveniles in a small environment is sufficient to set up the scaffold for the rest of the lifetime.

Heteroassociation of inputs onto scaffold

A content-addressable memory should store user-defined inputs and recall them from partial or corrupted inputs. The scaffold states can be used for content-addressable memory via heteroassociation of external cues with the scaffold.

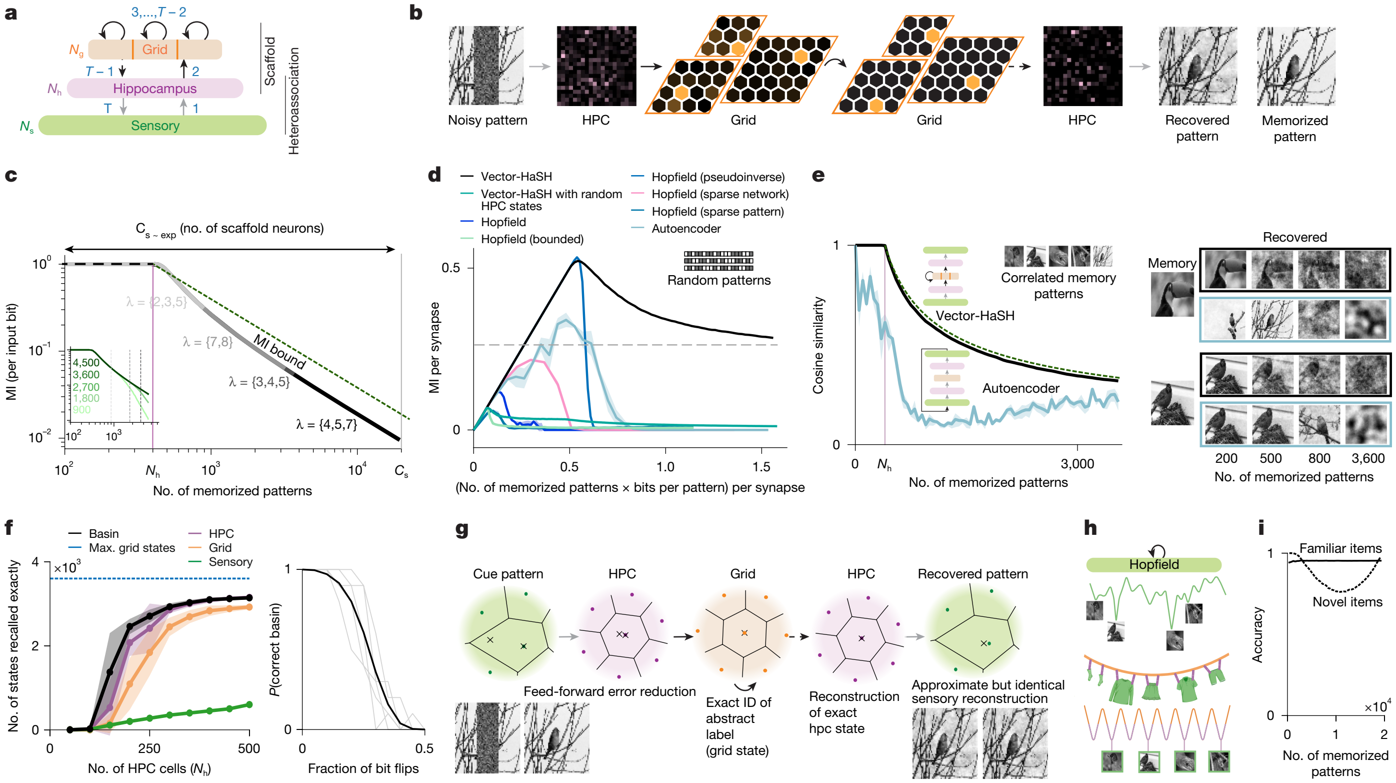


Fig. 3 | High-capacity content-addressable item memory via heteroassociation with scaffold. **a**, Scaffold receives sensory inputs which become associated via bidirectional associative learning. Numbers indicate update order for retrieval. **b**, Content-addressable recall from corrupted version of a memorized input. **c**, MI (normalized by number of stored patterns and bits per pattern) between stored and retrieved patterns. C_s is the number of scaffold attractors. Different grey and black curves represent different grid periods. Green dashed line represents theoretical MI bound. Inset curves represent varying N_s. **d**, The black line shows that MI (per synapse) between stored and retrieved patterns for Vector-HaSH tends to a non-zero constant (dashed grey line) as total number of memorized input bits (per synapse) increases. Coloured lines show that MI per synapse drops to zero for Hopfield network and variants. Assessed for random uncorrelated patterns (inset). **e**, Left, performance comparison (natural image patterns): Vector-HaSH (black) and fully trainable autoencoder of matched architecture and size, with identity output-to-input tail-biting weights (blue). Dashed line shows the analytical result (Supplementary Information, section D.3). Right, example reconstruction of two images, after variable numbers of stored patterns. **f**, Left, number of grid (orange), hippocampal (purple) or sensory (green) states recalled perfectly (zero error) when cued with corrupted sensory inputs (2.5% noise) as a function of N_h. Right, probability that the retrieved sensory state is in the correct basin versus fractional number of flipped bits in cue. Grey lines represent individual patterns; black line is an average over 500 patterns. **g**, Schematic of recovery in sensory, hippocampal and grid state spaces. Left to right, two different cues (one corrupted, one noiseless (bottom)), with evoked states (represented by x) progressing through the network. Dots indicate noiseless stored states. **h**, Energy landscape schematic. Top, the width, depth and positions of minima for Hopfield nets depend on pattern content and thus are highly variable. The Vector-HaSH scaffold functions as a 'clothesline' with equally deep, well-spaced minima, onto which arbitrary content is heteroassociatively 'hooked'. **i**, Accuracy of familiarity/novelty detection from hippocampal firing rates. In **c**–**f**, shaded regions represent the s.d. over five runs.

We will refer to inputs to the hippocampus from neocortex and non-grid entorhinal cells (Fig. 3a, green), as sensory inputs. A sensory input is associated with a randomly selected scaffold fixed point via a Hebb-like online implementation of the pseudoinverse rule between sensory input and the hippocampal state (simple Hebbian learning achieves the same asymptotic capacity with a reduced constant prefactor; Extended Data Fig. 3 and Supplementary Information, section D.6). These weights impose self-consistency so that hippocampal drive to the sensory states approximates the sensory activations input to the hippocampus.

Once acquired, memories can be reconstructed from partial sensory cues: these inputs drive a hippocampal state, which settles towards a scaffold fixed point via hippocampal–grid dynamics; finally, a sensory state is reconstructed by hippocampal-to-sensory weights. Thus, Vector-HaSH behaves as a content-addressable memory network (Fig. 3b).

A graceful item number–information trade-off

Memory recall in Vector-HaSH is perfect up to N_h input patterns (the circuit recovers all N_s bits per pattern correctly, where N_s is the sensory input dimension); after more than N_h patterns are stored, the information recovered for each pattern scales inversely with the number of patterns (Fig. 3c). Thus, Vector-HaSH exhibits a graceful trade-off or 'continuum' between pattern number and recall richness, rather than a memory cliff, and total information in the network remains parallel to the theoretical upper bound (the square of the number of synapses) out to the number of scaffold fixed points (Fig. 3c). Proof in Supplementary Information, section D shows perfect content-addressable recall for the first N_h states and near-optimal precision–pattern number trade-off beyond if the hippocampal states are a random projection from grid cells, passing through some nonlinear transformation; almost any nonlinear transformation, without fine-tuning, is sufficient (Supplementary Figs. 10 and 11).

Increasing the sensory input dimension and thus information per pattern does not reduce the information fraction recovered because although pattern size grows, so do the number of heteroassociative weights (Fig. 3c, inset; also see Supplementary Fig. 6). The information contributed per synapse approaches a constant non-zero value as the number of stored patterns increases (Fig. 3d), in contrast to Hopfield networks, where the value drops to zero[11,12] past the memory cliff (Fig. 3d). Other memory models (Supplementary Information, section D4) exhibit a similar cliff or only store one specific number of patterns for a fixed network architecture[51].

After most or all scaffold states have been used, there are three possibilities: (1) no further inputs are stored; (2) each new input replaces an existing memory (based on sensory overlap, age or random selection); or (3) all heteroassociative weights slowly decay so that older memories fade and those scaffold states are identified for reuse.

Comparison with end-to-end trained deep networks

Vector-HaSH can be unfolded for interpretation as an autoencoder[55] (Fig. 3e, left, architecture schematic) that has recurrent weights and is highly constrained: encoding and recurrent weights in the bottleneck layer are fixed, as are weights from the encoder to bottleneck and bottleneck to decoder layers. All remaining weights are set through biologically plausible associative learning. For comparison, consider an unconstrained and end-to-end gradient-optimized (via backprop) autoencoder of the same dimensions, with the addition of a tail-biting (output-to-input identity) connection for iterative reconstruction[55] (Fig. 3e, left, architecture schematic). Notably, Vector-HaSH substantially outperforms this autoencoder despite the much greater potential flexibility of the autoencoder (Fig. 3e), mirroring the results in ref. 54. The tail-biting exhibits a memory cliff, seen from curves (Fig. 3e, left) and reconstructions of a sample pattern (Fig. 3e, right). Vector-HaSH also outperforms optimized tail-biting and non-iterated autoencoders when cued with noisy sensory cues (Supplementary Fig. 7). Thus, the fixed scaffold provides a key inductive bias for robust high-capacity memory, which gradient optimization on an unconstrained architecture apparently cannot find or achieve.

Mechanisms of the memory continuum

We probe the circuit to understand its continuum behaviour. As N_h is varied above its threshold value, precision of reconstruction varies across the circuit (Fig. 3f, left). Grid and hippocampal states are nearly always recalled exactly. The sensory state is recalled approximately, with a continuous dependence on hippocampus size. Even when sensory retrieval is approximate, it falls in the correct basin (within the Voronoi region of the original sensory pattern (Fig. 3f, left)). The approximate recalled sensory state is identical, whether the cue is the true memory pattern (Fig. 3g, left) or a highly degraded version of it (Fig. 3f, right and Supplementary Fig. 8). In other words, although the precision of sensory reconstruction systematically decreases with the number of patterns (the distance of the recalled state within the Voronoi cell from the true sensory pattern increases)—accounting for the memory continuum—the reconstruction is reliable: regardless of the cue (which might be noiseless or corrupted), the reconstructed pattern is the same. This property will be important when modelling memory palaces.

Mechanistically, sensory-to-hippocampal projections partially denoise input cues (the hippocampal state is closer to its fixed point (Fig. 3g, green to pink) than the cue may have been to the true input). Next, scaffold dynamics recover the exact fixed point (Fig. 3g, pink to orange to pink) even deep in the memory continuum (analytical proof in Supplementary Information, section D.1) and for highly corrupted hippocampal states, because of the convex large-basin property of scaffold attractors. Finally, hippocampus-to-sensory projections decode the scaffold state to a reconstructed sensory state. Interference in the heteroassociative weights leads to growing approximation error with pattern number, but it remains continuous rather than cliff-like because decoding happens in a single feedforward pass rather than via iteration (proofs in Supplementary Information, sections D.2 and D.6). In sum, the factorization of attractor creation from content storage enables both pattern completion (exact recovery via recursion in the scaffold) and graded memory precision behaviour (feedforward decoding) in Vector-HaSH.

Conceptually, conventional autoassociative memory networks perform poorly because the locations, basin widths and depths of their attractors are governed by pattern content, leading to uneven, non-convex, small basins and many spurious minima. In Vector-HaSH, the attractor landscape is set by the regular structure of grid cell states, which produce well-spaced attractors with large basins and no spurious minima. Content is simply hooked onto these pre-structured states, in analogy with a clothesline (the scaffold) to which any clothes (sensory patterns) can be attached (via heteroassociation) (Fig. 3h).

One-shot robust recognition memory

Memorized inputs, because they are associated with scaffold attractors, generate grid-defined hippocampal states that form a narrow distribution with highly similar firing rates. When a novel sensory input drives the hippocampal state, that state, as well as the projection into grid cells and their initial return projection back to hippocampus, form patterns that are far outside the usual distribution for both cell populations (Supplementary Fig. 12). A simple decoder that detects firing rate deviations in either direction from the usual mean in the hippocampal state—implemented with two hidden units and fixed parameters from the time of scaffold formation (independent of the sensory inputs or number of memories stored; Supplementary Information, section D.7)—acts as a reliable familiarity/novelty discriminator (Fig. 3i). Such a recognition memory can be used to decide a sensory input should be cleaned up (pattern completed) for recall or trigger new memory creation (association with a fresh scaffold state).

Spatial inference and memory

When self-motion signals drive transitions between grid cell states (Fig. 4a), the architecture and dynamics of Vector-HaSH support spatial memory without catastrophic forgetting and zero-shot spatial inference along novel paths.

At a landmark or corner of a novel room, grid module phases are initialized randomly. Velocity inputs update the grid phases by path integration[8]. Vector-HaSH learns a map of the room via associations between the grid-driven scaffold states and sensory cues. Its structure permits successful reconstruction from either input (Supplementary Information, section D), thus grid states can be recalled from sensory cues or vice versa (Fig. 4a). If a familiar room is traversed without access to sensory cues (in the dark or in-between landmarks), Vector-HaSH path integrates to update grid (and thus hippocampal) states. At a landmark, the hippocampal state is updated via the sensory-hippocampal weights (Fig. 3a,b), resetting any path-integration errors. Thus, without threshold modifications, hippocampal states can be determined by grid inputs alone, sensory inputs alone, or a combination of these. Grid cells and hippocampal cells exhibit realistic spatial tuning, including the localized and usually single-bump tuning typical of place cells (Fig. 4b and Supplementary Fig. 13).

After very sparse exploration in a novel room (trajectory of Fig. 4c, left), Vector-HaSH is able to predict expected sensory observations along an entirely new route through the room (trajectory of Fig. 4c, right). This zero-shot inference ability arises from the path invariance of velocity integration: velocity updating generates accurate grid states even along novel paths, which then reconstruct (predict) sensory cues associated with those states.

After sequential exposure to a set of rooms (Fig. 4d), Vector-HaSH learns distinct spatial maps, assessed for each by testing sensory-cued grid state inference (without path integration) after seeing that room (Fig. 4d,e). To examine gradual interference between rooms (catastrophic forgetting), we test Vector-HaSH in all prior rooms right after learning the *i*th room (Fig. 4d,e). Recall of hippocampal and grid states from sensory cues remains unchanged for all prior environments after subsequent acquisition of up to 10 rooms, without replay or

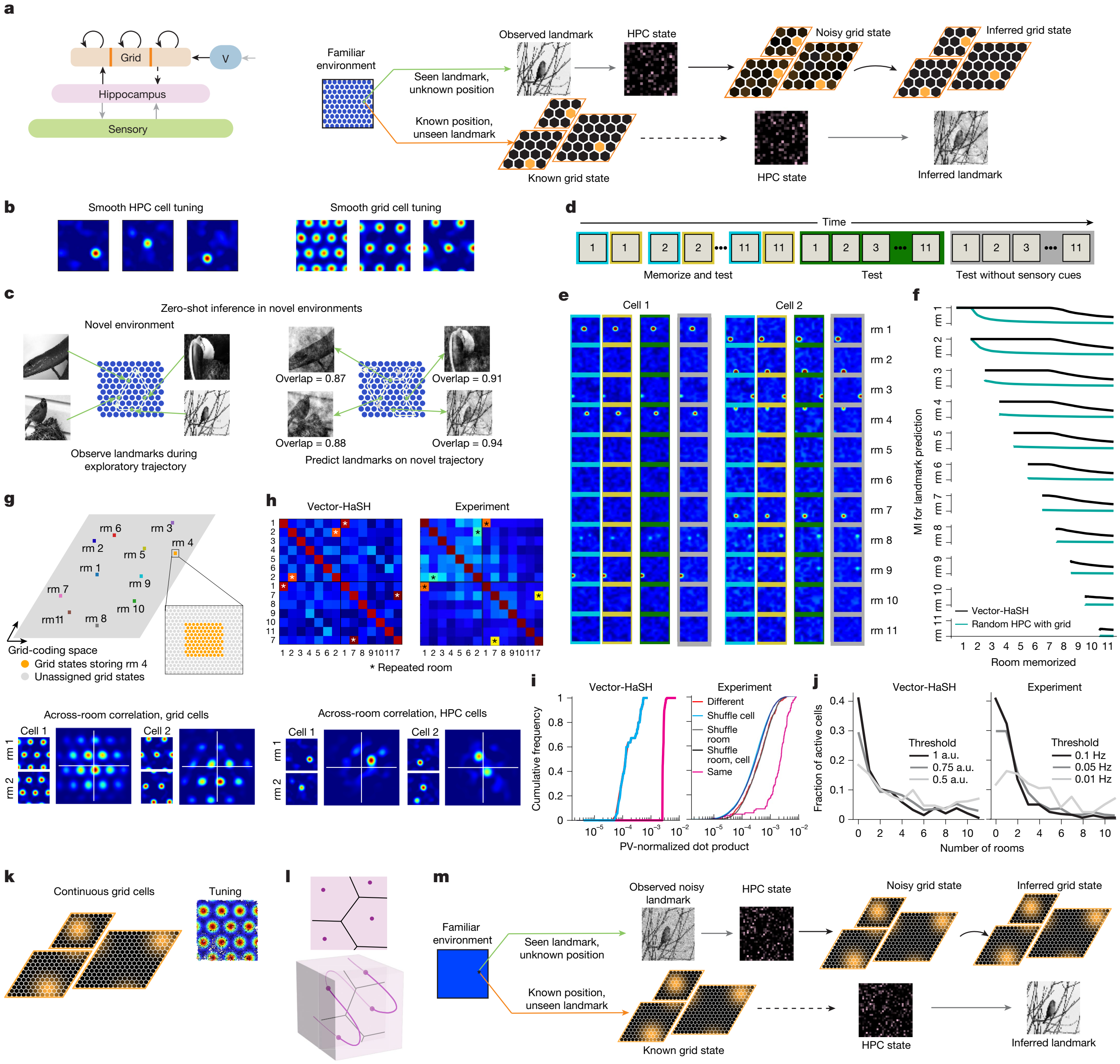


Fig. 4 | Memory, inference and lifelong learning in the spatial domain. **a**, Left, Vector-HaSH with velocity inputs that drive grid phase shifts. Right, in a familiar environment (after heteroassociative learning), bidirectional inference of position (internal grid states) from sensory cues (top), or prediction of sensory landmarks from a grid state (bottom). **b**, Tuning curves (smoothed) of three hippocampal cells and one cell from each grid module (unsmoothed curves in Supplementary Fig. 14). **c**, Left, first traversal of environment (white trajectory), with heteroassociative learning of landmarks. Right, prediction of landmarks on an entirely novel path (white trajectory) beginning at a known location. For realism (to put it on the memory continuum) Vector-HaSH previously made 596 other landmark–scaffold associations. **d**, Sequential learning protocol for 11 rooms: Vector-HaSH learns landmark–state associations on random trajectories in each room (cyan), and is immediately tested on scaffold state inference from landmarks (mustard) with a new trajectory. All rooms are re-tested for scaffold state inference from landmarks after learning all 11 rooms (green), then tested again for landmark inference from scaffold states ('dark condition' (grey)). **e**, Spatial tuning curves for **d**. rm, room. **f**, Black, landmark reconstruction accuracy for room *i* after learning rooms 1,...,*i* − 1. Cyan represents a control model, as in green curves in Figs. 2 and 3. **g**, Top, grey diamond represents the across-module grid-coding space. Random assignment of the starting grid phase generically results in non-overlapping room representations. Bottom, the shift in grid phase going from room 1 to 2 is different across modules (left versus right). **h**, Top, hippocampal state similarity matrix across rooms (with repeated exposures to some rooms, asterisks): Vector-HaSH (left) and experiments[72] (right). Bottom, total remapping of hippocampal cells across rooms. **i**, Hippocampal 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. **j**, Distribution of hippocampal cells active in *R* rooms versus *R*, in model and experiment. a.u., arbitrary units. **k**, Continuous-valued grid activations and space. Left, grid population vector, multiple modules. Right, raw (unsmoothed) spatial tuning of a grid cell (random trajectory). **l**, Top, schematic scaffold attractors (purple dots) in discrete Vector-HaSH. Bottom, similar to top, for a continuous version (line represents a 1D attractor for 1D space). **m**, As in **a**, with continuous Vector-HaSH.

consolidative rehearsal. This is due to the exponential scaling capacity and architecture of Vector-HaSH, in which random grid phase initializations result in well-separated maps (Fig. 4g). Thus the model avoids not only a memory cliff but also catastrophic forgetting in grid and hippocampal recall.

In the 'dark' (no visible landmarks), after the initial grid state is specified for each room, the model is able to recall a large amount of sensory information (1 cue per location in every room) over 11 rooms, although the sensory cues in all rooms are recalled less vividly after learning 11 rooms (rightmost part, black curves of Fig. 4f) because the circuit is in the memory continuum.

Hippocampal tuning is stable on repeated visits to the same room, with orthogonal representations of different rooms as in experiments (Fig. 4i,j). Additional properties, including the distribution of probabilities that a hippocampal cell has a field in multiple rooms, match experimental data (Fig. 4k).

In a continuous-activation and continuous-space implementation of Vector-HaSH (Methods), the discrete scaffold attractors form a folded two-dimensional continuous attractor within the N_h-dimensional space of hippocampal states. The basins remain large ($N_h - 2$-dimensional instead of N_h-dimensional; Fig. 4l illustrates this using a one-dimensional continuous attractor). Vector-HaSH still performs robust associative memory retrieval and inference in the continuous limit (Fig. 4m).

Vector updating of grid states

Episodic memory centrally involves sequences of events in time. Sequence memory can be modelled with asymmetric Hopfield networks[56] in which the user-defined pattern at one time drives, through learned weights, the next user-defined pattern. These models result in similar or stronger capacity limitations as for item memory[12,55]: such networks quickly fail (within approximately 50 steps) to reconstruct even an approximation of the next pattern (Fig. 5b). We will show that, remarkably, Vector-HaSH permits massive sequence memory by factorizing the problem to construct high-capacity abstract sequences (a sequence scaffold) and then affixing content via heteroassociation.

First, we hypothesized that modelling hippocampus as an asymmetric Hopfield-like network directly encoding the input patterns, with bidirectionally learned grid cell interactions to help denoise and pattern complete the imperfectly reconstructed next hippocampal state, might support high-capacity sequence reproduction. This roughly doubled the sequence capacity (to approximately 100 steps), but did not fundamentally alter capacity scaling with network size (Fig. 5b).

Next, in the full spirit of the scaffold network, we reasoned that learning an abstract sequence of scaffold states rather than user-defined hippocampal states might be the solution. Hippocampal states were given by random grid state projections and hippocampus-to-grid weights were associatively set to be consistent with the next (rather than the current) grid state. Despite the full benefit of the scaffold architecture, sequence capacity remained low (Fig. 5b, failure within approximately 30 steps). We hypothesized that this happened because even abstract grid states are large and specific activity patterns, for which the previous hippocampal state must provide sufficient information to reconstruct. This hypothesis gave us the critical insight that grid states, which can be specified from a previous one by a two-dimensional velocity input acting on the grid network via a velocity-shift mechanism, could enable efficient sequence specification by memorization of a sequence of two-dimensional (and thus very low-information) velocity vectors.

Consider using the previous hippocampal state to cue the next grid state, but via the drastic dimensionality and complexity reduction of the velocity-shift mechanism: the previous hippocampal state specifies a two-dimensional velocity that signals where to move in the grid-coding space. A small and simple feedforward network (a multi-layer perceptron (MLP)) (Fig. 5a, top) associated the previous grid state, via the hippocampus, with a two-dimensional velocity vector. This architecture resulted in the accurate reconstruction of abstract scaffold sequences of 1.4×10^4 states, using the same small number of cells in the scaffold network as before (Fig. 5a, left). Alternatively, recalled sensory states can drive the low-dimensional velocity transitions, without a separate MLP (Supplementary Information, section D.8).

To quantify sequence scaffold capacity, we statistically assessed how well the circuit could recall random velocity (shift) vectors assigned to each grid state (Fig. 5c). The sequence scaffold perfectly recalled approximately 1.5×10^5 state-velocity associations with $N_h = 500$ and $N_g = 275$ neurons, with grid periods 5, 9 and 13 (totalling approximately 3.4×10^5 grid states). The scaling of scaffold sequence length with the number of hippocampal cells is again sub-logarithmic, similar to scaffold capacity scaling for item memory (Fig. 5d, left); the number of MLP units needed to learn the hippocampal state-to-velocity mapping is linear with a very small coefficient (approximately 10^{-3}) (Fig. 5d, right).

Using the sequence scaffold, Vector-HaSH supports high-capacity episodic or sequence memory by hooking external inputs experienced over an episode onto a sequence scaffold (Fig. 5f). The abstract sequence scaffold may be formed concurrently with the learning of the heteroassociative weights that link sensory inputs to the scaffold, or inputs could be affixed to a pre-existing sequence scaffold that is learned once. For episodic memories without clear metric variables such as spatial position, the scaffold trajectory can be arbitrarily chosen.

Asymmetric Hopfield networks and tail-biting autoencoders[55] quickly diverge from the trained state sequence during recall; in a sequential version of the memory cliff, recalled states do not even approximately resemble the trained states (Fig. 5h). In Vector-HaSH, internal grid sequences are recalled with essentially perfect fidelity over very long sequences, and sensory states are recalled perfectly in terms of identity but only approximately in content depending on the total length of the memorized sequence (Fig. 5i) (the equivalent to the item memory continuum). Because sequence transitions are generated entirely within the scaffold, their continued fidelity does not depend on the fidelity of sensory reconstruction. As for item memory, the quality of sensory recall per state in the sequence (MI per input bit) degrades as the total memorized sequence content grows (Fig. 5i), but the information recalled per synapse remains finite and approaches a constant asymptotically (Fig. 5j), while it drops to zero for other models. To summarize, while in conventional memory models the current state and its recurrent projections must carry all the information to reconstruct the high-dimensional next state, in Vector-HaSH the current (scaffold) state and its recurrent projections must reconstruct merely the next two-dimensional velocity vector (Fig. 5k). The high-dimensional vector is then reconstructed via feedforward decoding in the sensory areas.

Our information-based hypothesis is that failure or success in sequence memory depends on how much information the current state must specify to construct the next state. We test this hypothesis by varying the amount of information that the network must recall at each step, by increasing the range of possible velocities (length of velocity vectors) to be recalled. The recalled sequence fraction decreased systematically with increasing velocity range (Fig. 5e), in proportion to the theoretically expected inverse proportionality to the number of information bits required to specify the velocity (blue). In sum, constraining sequence recall dynamics to a low-dimensional manifold where only low-dimensional tangent vectors (velocities) rather than the manifold states themselves must be reconstructed results in vast increases in recalled sequence length. Thus, the path integrability of the grid cell code can not only support spatial inference and mapping, but also serve as a scaffold for episodic and sequence memory even in the absence of any spatial inputs.

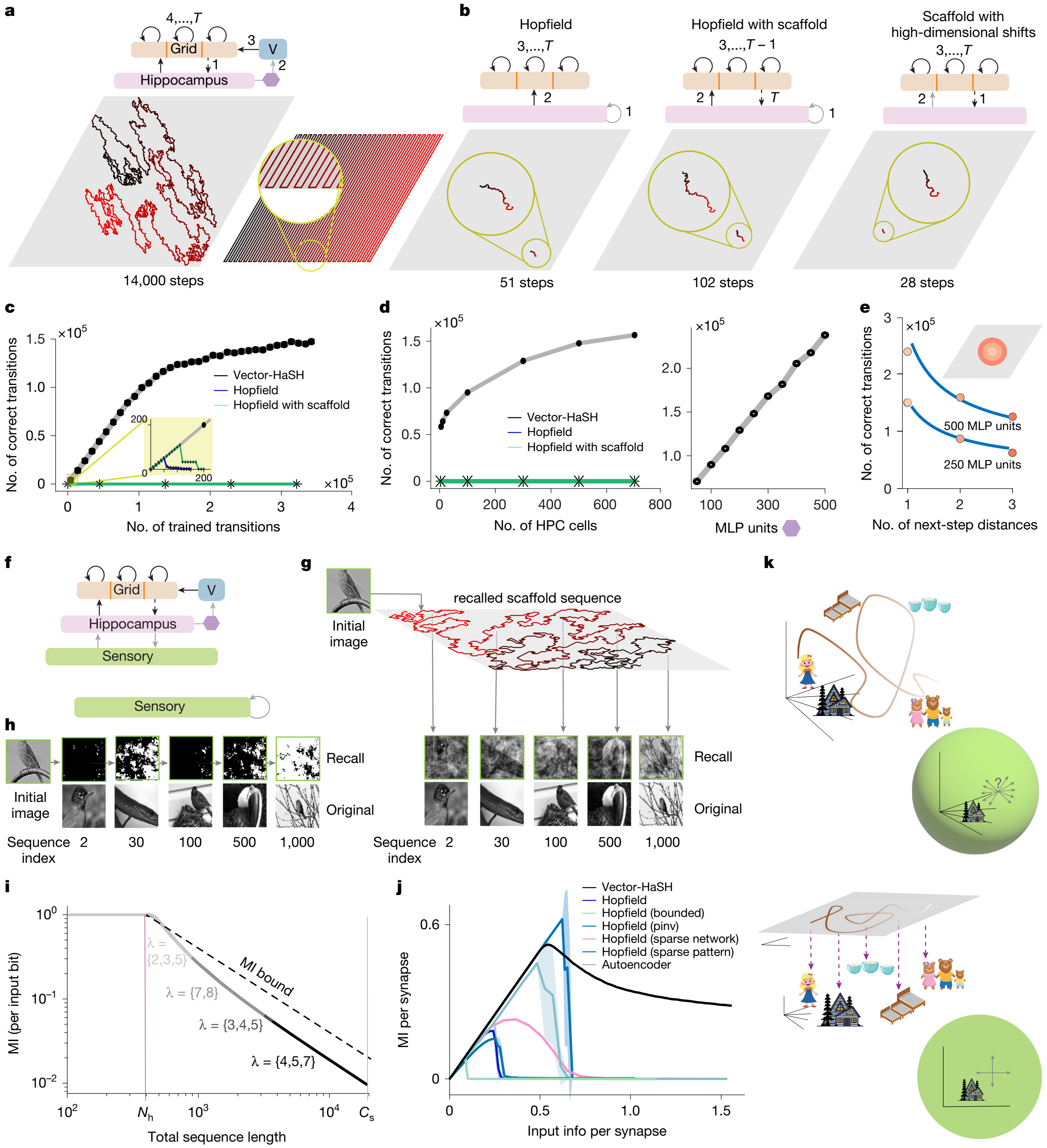


Fig. 5 | Sequence scaffold with low-dimensional grid cell shifts enables high-capacity episodic (sequence) memory. a, Top, sequence scaffold architecture. Hippocampal state drives next grid transition via the 2D velocity-shift mechanism. A small MLP (violet; single hidden layer with 250 units) maps from hippocampal state to 2D velocity. Bottom, accurate recall of a 14,000-step self-avoiding random Lévy flight sequence ($\lambda = \{5, 9, 13\}$ and $N_h = 500$) and a 3,600-step hairpin sequence over all scaffold states ($\lambda = \{3, 4, 5\}$). All sequences are visualized in grid space. **b**, Left, asymmetric Hopfield network. Middle, error correction of retrieved Hopfield states by scaffold. Right, scaffold network with hippocampus-to-grid connections directly driving the next grid state. Lévy flight sequence network parameters as in **a**. The numbers indicate recalled steps. **c**, Each state is randomly assigned a velocity or next state. Plot shows correctly recalled velocities/next state for Vector-HaSH versus Hopfield baselines. **d**, Number of recalled next states increases towards the exponentially large number of grid states as the number of hippocampal cells increases (left; trained on 585 × 585 random transitions) and grows linearly with the size of the hippocampus-to-velocity MLP (right). **e**, Correctly recalled transitions decrease as the number of potential next states for each transition increases. Blue lines show inverse entropy of the set of next states (1/(log of number of next states)). **f**, Vector-HaSH for episodic memory. Sensory inputs are heteroassociatively coupled to the sequence scaffold from **a**. **g**,**h**, Recalled natural image sequence in Vector-HaSH (**g**) and an asymmetric Hopfield network with matched synapse number (periods {3, 4, 5}, $N_g = 50$, $N_h = 400$, $N_s = 3{,}600$; $N_s = 3{,}600$ for Hopfield). **i**,**j**, MI per input bit (**i**) and MI per synapse (**j**) for sensory sequence recall (random binary patterns) across models. Shaded areas show s.d. over five runs. **k**, Top, in conventional sequence memory models, the previous state must reconstruct the entire content of the next state. Bottom, in Vector-HaSH, the previous state must reconstruct a mere two-dimensional vector, which requires far less information in the recurrent loop.

Entorhinal and hippocampal phenomenology

While serving as a general-purpose memory circuit, Vector-HaSH recapitulates many grid cell properties shared with continuous attractor models (which constitute its core), and several hippocampal and full-circuit properties. Immediately in novel environments, Vector-HaSH grid cells exhibit periodic activity bumps (Fig. 6a, top), which align with the regular pattern seen on further exploration[57] (Fig. 6a, bottom). Similar patterns are present during dark exploration (no sensory cues) over short times (Fig. 6b, left). Alternative models require extensive exploration and sensory cues for grid tuning emergence[9,45]. Vector-HaSH exhibits grid resetting by sensory cues: After dark navigation and phase drift in a familiar environment, a sensory cue ('lights' turned on) resets the hippocampal (and then grid) states via the sensory-to-hippocampal in one cycle through the circuit (Fig. 6b, right), as in refs. 40,57. Vector-HaSH co-modular grid cells exhibit invariant relationships seen across behavioural states and environments where spatial tuning curves change[33,34,36–38] (Fig. 6c, left) (Supplementary Fig. 18 shows relative phase invariance across environments of different dimensions), whereas hippocampal cells in Vector-HaSH and experiments globally remap[36] (Fig. 6c, right).

Vector-HaSH recapitulates grid–place cell correlations[45,58]: the fraction of grid cells whose fields overlap with that of a place cell across environments (dashed line; Fig. 6d) is significantly larger (*P* value of 0.0) than shuffle controls (place fields randomly reassigned across cells; Fig. 6d), consistent with other models where grid cells drive place fields[10,35,45].

Grid cells determine hippocampal states in Vector-HaSH, yet there is a strong reverse influence: If the hippocampus is lesioned when grid velocity inputs are noisy, grid cell spatial tuning is destroyed (Fig. 6e, left), as in ref. 59. Nevertheless, grid cells maintain their relative phase relationships (Fig. 6f; Methods), consistent with ref. 60. Thus, place cells are critical for reliable grid spatial tuning. By contrast, with sufficient sensory inputs, hippocampal tuning remains unchanged after grid lesioning[61] (Fig. 6e, right). Thus, the circuit exploits all available means to estimate position: velocity (via grid cell integration), external cues (via hippocampus), both, or either. It mechanistically reconciles the question of whether place cells emerge from grid cells or vice versa.

Vector-HaSH generates splitter cells[62–64], whose spatial tuning depends on context, recent memory and other factors, via the mechanism of grid phase remapping when internally generated context is appended to the hippocampal sensory input. In a spatial T-maze alternation task, context in the central stem is distinct for incoming trajectories from the right or left return tracks. A distinct context appended to the sensory input produces grid phase remapping in Vector-HaSH (differential grid phase shifts across modules; Supplementary Figs. 23 and 24), total hippocampal remapping (Fig. 6g) and hippocampal cells that are contextually selective. The same process yields directionally selective place cells on linear (Fig. 6i) and circular one-dimensional tracks, tree mazes, and radial mazes (Supplementary Figs. 19–22). The resulting ratios of splitter to non-splitter and direction-dependent to directionally untuned hippocampal cells were similar to values from experiment[64] (Fig. 6h,j) with the predictions that grid cells will possess splitter-like and directional tuning (Supplementary Fig. 24) and that splitter and directionally tuned hippocampal cells are not a separate biophysical type; this contrasts with the possibility that splitter (and direction-dependent) cell tuning is generated within the hippocampus by (possibly) distinct cell types.

An untested prediction is that grid activations should be periodic and hippocampal states that are much lower-dimensional than expected (relative to the dimensionality from randomly shuffling hippocampal fields) when traversing abstract domains and even recalling episodic memories, if the states are plotted as a function of the relevant abstract variable (Fig. 6k).

Finally, for hippocampally dependent memories, items that are seen or recalled repeatedly are more resistant to hippocampal damage[65], a phenomenon known as hippocampal memory consolidation. We exposed Vector-HaSH to several inputs, some of which were presented multiple times, leading to additional increments of the corresponding hippocampus-to-sensory weights (Fig. 6l). Memories thus reinforced were remembered with richer detail (Fig. 6m), and recall was relatively robust to neuron removal (Fig. 6n and Supplementary Fig. 16). Additionally incrementing hippocampus-to-grid and sensory-to-hippocampus weights produced no further consolidation (Fig. 6o). The primary role of directed synapses from hippocampus to cortex is consistent with and provides specific predictions about mechanisms for consolidation of hippocampus-dependent memory[66]. An alternative hypothesis is that repeating inputs form associations with multiple scaffold states. Vector-HaSH simulations (1,000 runs; data not shown) did not support this hypothesis: associating an input with two different scaffold states always (100% of runs) resulted in the activation of a third, unrelated scaffold state when presented with a partial sensory cue.

Mechanism for the memory palace technique

Vector-HaSH provides the first model to explain the power of the method of loci (memory palaces), a mnemonic technique that has been used for millennia and is currently widely exploited by memory athletes[13]. Given a randomly ordered deck of playing cards, memory athletes take an imagined walk through a familiar and richly remembered space, and 'place' the cards near landmarks they encounter along the way. At recall time, they walk through and 'collect' the items they had placed. Counterintuitively, by adding to their memory task the demand of also recollecting the correct item–landmark associations, they are able to perform highly accurate one-shot memorization and recall.

The hippocampus (and Vector-HaSH, via vector-based sequence memorization) already enables rich memory for sequences, as we have seen. However, deep in the continuum, sensory recall is only approximate: a long new sequence of items cannot be exactly memorized using the native mechanism (Fig. 3e, upper row). However, the whole of Vector-HaSH, working in the regime of recall of a familiar sequence in the memory continuum, can itself be converted into a new scaffold for precise memory, as follows. Suppose Vector-HaSH is initialized to a starting location in a highly familiar environment, and items at locations along the path taken by the memory athlete are recalled. Crucially, even though the recalled sensory states are only an approximation of the actual items, each time they are recalled they are reliably the same (Fig. 3g, green); thus, even deep in the memory continuum, these recalled sensory states can have the role of perfectly retrievable abstract scaffold states for heteroassociation of the new neocortical inputs (Fig. 7b,c). For this new extended scaffold, the heteroassociative linkages are well before its continuum regime, and the inputs can be recalled with high fidelity (Fig. 7d, blue) even when associated with sensory states that are a poor approximation of the original sensory inputs. Another advantage of using this extended scaffold is that the effective information capacity bottleneck, which was previously the size of the hippocampus, is now the size of the sensory input area and could be much larger, meaning that perfect memorization of a much larger number of inputs is possible (Fig. 7e). Memory athletes cycle between using different memory palaces to refresh recently used palaces; in terms of the model, this presumably corresponds to erasure of heteroassociative linkages with the palace and refreshes the ability to reuse that scaffold in a non-continuum regime.

A key prediction of Vector-HaSH is that during memory palace use, grid, hippocampal and sensory area cells should be reliably activated in a way that correlates with location in the palace, and the correlation with location should be more faithful than correlation with the neural responses to the actual sensory inputs that were present when experiencing and acquiring memories of the palace. Another is that

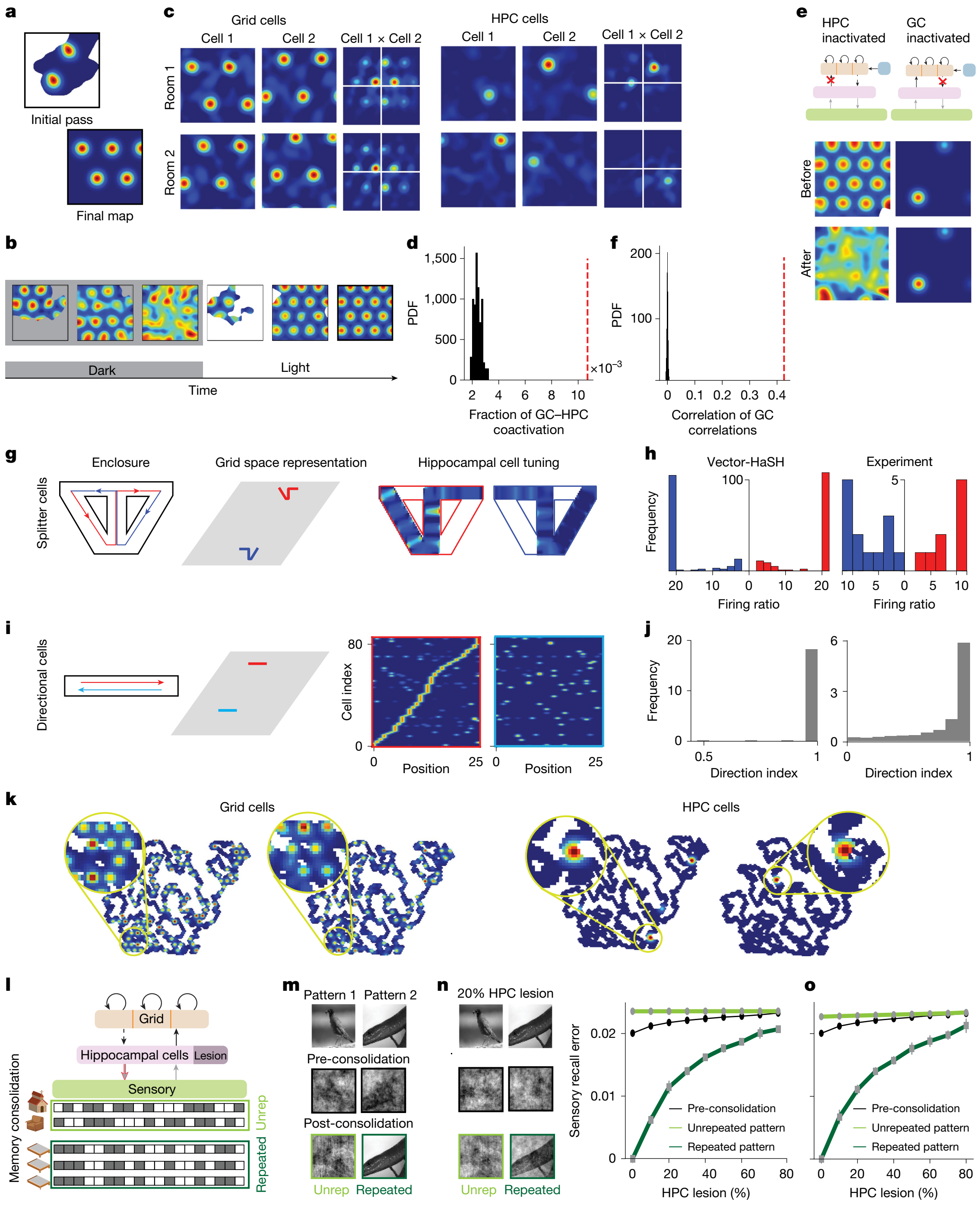


Fig. 6 | Vector-HaSH reproduces multiple aspects of entorhinal and hippocampal phenomenology. a, Grid fields are present in first steps in an environment (top) and are consistent with the map formed after fuller exploration (bottom). **b**, Grid fields form in dark; they may eventually wash out without corrective sensory input (left three panels); rapid phase reset occurs when cue is made visible if there was a phase error (remaining panels). **c**, Left, grid cell tuning curves shift across environments. Right, relative phases (closest peak to cross-hairs in cross-correlation) of co-modular neurons are preserved. This is not the case for hippocampal cells. **d**, Fraction of grid cells that exhibit across-environment coactivation with a hippocampal cell (red); the distribution after shuffling place field locations is shown in black. PDF, probability distribution function. **e**, Top left, grid cell spatial tuning with noisy velocity inputs before and after hippocampal inactivation. Top right, grid cell inactivation. Bottom, hippocampal tuning with sensory cues present, before and after grid network inactivation. **f**, Similarity in correlation matrix of co-modular grid cells, before and after hippocampal inactivation lesion (red), versus similarity after shuffled correlations (black). **g**, Left, alternation task, with left-to-right (red) and right-to-left (blue) trajectory contexts. Remapping in grid space (middle) yields distinct hippocampal tuning for the two contexts (right). **h**, Hippocampal left/right selectivity ratios in central stem[62]. **i**, Left, left/right trajectories in 1D. Middle, as in **g**. Right, hippocampal cells, ordered by rightward run field locations for rightward (red outline) and leftward (blue outline) runs. **j**, Directionality index[63,64] for Vector-HaSH (left) and experiments (right). **k**, Vector-HaSH grid and hippocampal cells have structured responses; plotted in the relevant continuous space (for example, time) for abstract domains (for example, traversal though an episodic memory). **l**, Memory consolidation. Repeated patterns drive further hippocampal-to-sensory weight increments (red outlined arrow). Unrep, unrepeated. **m**, Patterns (top) and recall (middle) in the continuum regime. Bottom, consolidation of pattern 2 but not pattern 1 leads to selectively richer recall. **n**, Patterns (top) and recall without consolidation after 20% hippocampal lesion (middle). Bottom, consolidated patterns (dark green outline) are protected at the cost of unconsolidated ones (light green outline). **o**, Same as **n**, with additional increments in sensory-to-hippocampus and hippocampus-to-grid weights. Error bars in **n**,**o** indicate s.d. across three runs.

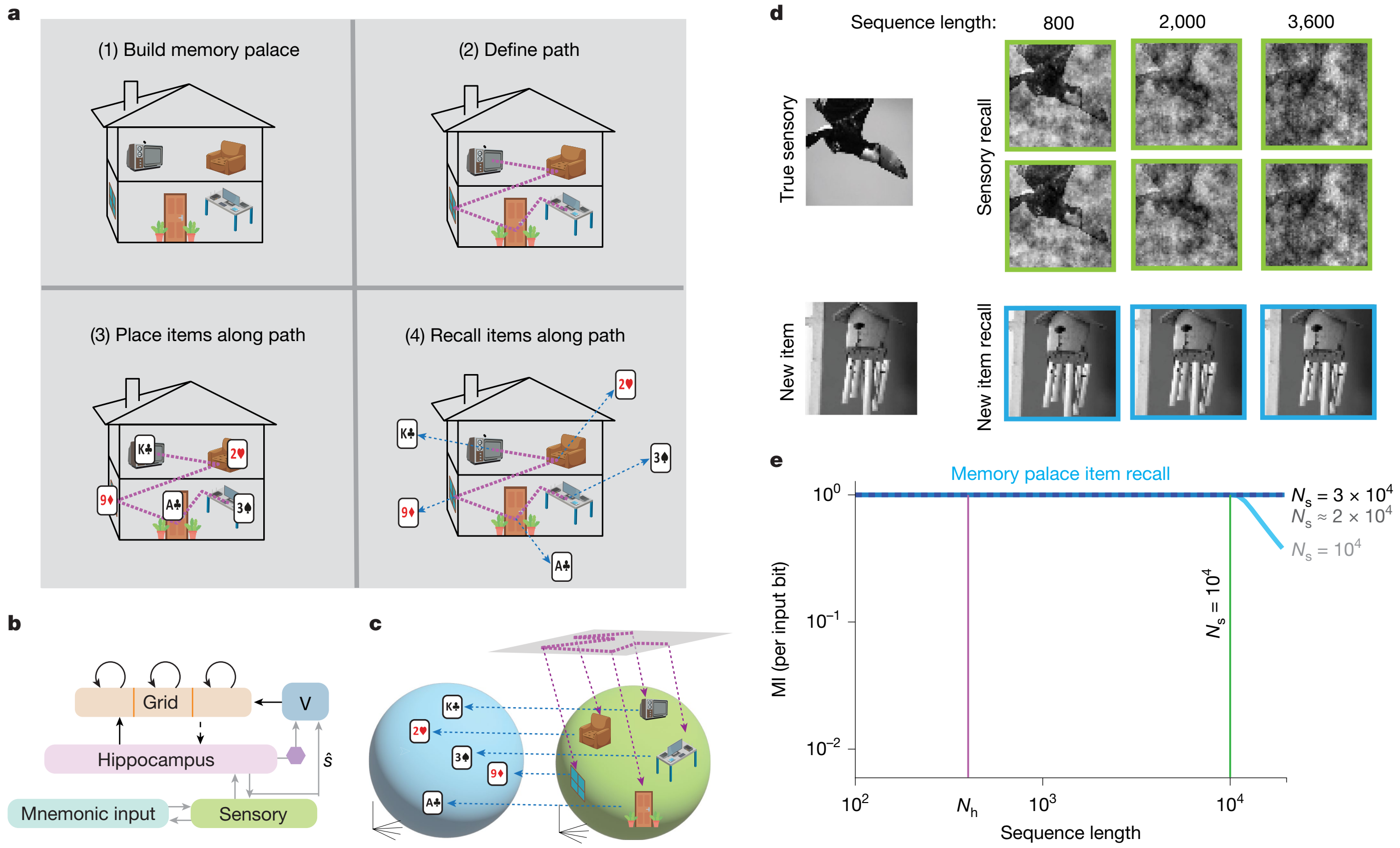


Fig. 7 | Accurate recall of arbitrary inputs by heteroassociation onto landmarks in a memory palace formed by Vector-HaSH. a, The memory palace technique. **b**,**c**, Circuit for the memory palace technique (**b**); the full Vector-HaSH circuit for episodic memory (compare with Fig. 5 or Supplementary Information, section D.8) is a scaffold for new inputs via heteroassociation onto recalled sensory states (**c**). **d**, Recalled sensory states in Vector-HaSH are approximate (top row), but at any given point in the continuum, a consistent state is reliably retrieved (compare first and second rows), thus they qualify as new abstract scaffold states. When new sensory data are (hetero)associatively linked to an approximately recalled sensory sequence (at any given depth in the memory continuum of Vector-HaSH, shown here for three depths), it is recalled with high fidelity (bottom). **e**, Memory curve for items remembered using Vector-HaSH as a memory palace scaffold (blue). The curve applies to newly remembered items using a given memory palace, regardless of where Vector-HaSH is in its memory continuum. Error bars show s.d. over 10 runs. In **d**,**e**, $\lambda = \{4, 5, 7\}$, $N_h = 400$; N_s is the size of the sensory output of Vector-HaSH.

memory palaces could be built from familiar non-spatial sequences involving rich sensory data.

Discussion

Extensions

There are many paths for extending Vector-HaSH, which include modelling the subregions of hippocampus; incorporating hippocampal cells that respond to a simultaneous combination of grid and sensory inputs to model partial remapping; including local field potential oscillations as possible gating processes for the between-region iterations that we have assumed occur in ordered cycles; modelling how the circuit enables goal-directed behaviours; and many others.

Relationship to anatomy

Anatomically, entorhinal projections to hippocampus derive from superficial layers and return projections arrive at deep layers, whereas Vector-HaSH predicts a tight and fully self-consistent loop from grid cells to hippocampus and back. This can be tested connectomically, and indeed, new discoveries still surprise: deep entorhinal layers send a copy of their outputs back to the hippocampus[67].

Random fixed grid-to-hippocampal weights are important and sufficient for several properties of Vector-HaSH, and several types of non-random weights are insufficient, but this does not eliminate the possibility of non-random solutions. For instance, expander graphs for error-correcting codes admit non-random solutions in principle, but they have been difficult to find, whereas random connections are sufficient. Of note, ref. 68 shows that the layer II MEC stellate cell circuit (these comprise most grid cells) is the first to mature, and activity in this network then drives maturation of the hippocampal circuit, followed by entorhinal layer 5, and finally, layer II of LEC. This maturation order is exactly consistent with what our model would predict on the basis of how it must be structured: the scaffold has to be formed first, and within it, the grid cell circuit is formed first, followed by the formation of fixed connections to hippocampus, then the hippocampal circuit, and finally the hippocampal–non-grid entorhinal circuit.

Experimental tests

We have highlighted a number of model predictions throughout. Central parts of our model could be invalidated if, for instance: grid-to-hippocampal synapses were plastic at a timescale faster than the hippocampal-to-grid synapses; if grid cell lesions did not affect episodic memory; or if, when imagining traversing a childhood home, the grid representations were distinct from those when using the same home as a memory palace when using the method of loci. Finally, because Vector-HaSH is a dynamical neural network, it can be directly queried for experimental predictions about representation, dynamics and learning under a large variety of conditions and perturbations.

Related models

Together with the bipartite expander network and MESH network models[44,54], Vector-HaSH defines a new class of memory models that we name robust hash-based memory: they create exponentially many fixed points with large basins as abstract error-correcting states for memory. Furthermore, existing models do not implement the low-dimensional shift mechanism of Vector-HaSH for efficient sequence memory.

Vector-HaSH resembles the models in refs. 7,10, which involve pre-structured grid cell representations interacting with hippocampal cells. Learning of grid-to-hippocampus weights prevents these models from having high-capacity, large robust basins and strong generalization, and avoiding the memory cliff and catastrophic forgetting. Vector-HaSH further contrasts with models that learn the structure of the explored space, such as successor representation[9], principal components analysis[41] and the Tolman–Eichenbaum machine[45], because their internal representations derive from the geometries and dimensionality of the environment so can be non-grid-like, and require sensory inputs to form representations in new environments.

Any model of hippocampal memory necessitates compression of cortical inputs. Modelling hippocampus as a bottleneck layer in an cortico-hippocampal autoencoder (Fig. 3 and refs. 55,69) is a form of content-based compression. These models lack the capacity, resistance to catastrophic forgetting and sequence memory properties of Vector-HaSH. In Vector-HaSH, the hippocampus is a content-independent pointer or hash for content localized in cortex, related to the models of refs. 70,71. The circuit performs locality-sensitive hashing for episodic memory, with locality defined in the temporal domain (temporally contiguous inputs map to contiguous grid states).

Nevertheless, commonalities among these models point towards a converging view of the hippocampal complex. The highly performant features of Vector-HaSH suggest a first-draft understanding of the circuit mechanisms of the hippocampal complex as a general memory system.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-024-08392-y.

1. Scoville, WilliamBeecher & Milner, B. Loss of recent memory after bilateral hippocampal lesions. *J. Neurol. Neurosurg. Psychiatry* **20**, 11 (1957).
2. 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).
3. 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).
4. Solstad, T., Boccara, C. N., Kropff, E., Moser, May-Britt & Moser, E. I. Representation of geometric borders in the entorhinal cortex. *Science* **322**, 1865–1868 (2008).
5. 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).
6. Samsonovich, A. & McNaughton, B. L. Path integration and cognitive mapping in a continuous attractor neural network model. *J. Neurosci.* **17**, 5900–5920 (1997).
7. Hasselmo, M. E. A model of episodic memory: mental time travel along encoded trajectories using grid cells. *Neurobiol. Learn. Mem.* **92**, 559–573 (2009).
8. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLoS Comput. Biol.* **5**, e1000291 (2009).
9. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643–1653 (2017).
10. 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).
11. Abu-Mostafa, Y. S. & St Jacques, J. Information capacity of the hopfield model. *IEEE Trans. Inf. Theory* **31**, 461–464 (1985).
12. Gardner, E. The space of interactions in neural network models. *J. Phys. A* **21**, 257–270 (1988).
13. Yates, F. A. *The Art of Memory* (Routledge & Kegan Paul, 1966).
14. Proust, M. *À la Recherche du Temps Perdu* (Grasset, 1913).
15. Reed, J. M. & Squire, L. R. Impaired recognition memory in patients with lesions limited to the hippocampal formation. *Behav. Neurosci.* **111**, 667 (1997).
16. Zola, S. M. et al. Impaired recognition memory in monkeys after damage limited to the hippocampal region. *J. Neurosci.* **20**, 451–463 (2000).
17. 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).
18. Eichenbaum, H. On the integration of space, time, and memory. *Neuron* **95**, 1007–1018 (2017).
19. 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).
20. Lee, A. K. & Wilson, M. A. Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron* **36**, 1183–1194 (2002).
21. Fenton, AndréA. 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).
22. Colgin, LauraLee et al. Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature* **462**, 353–357 (2009).
23. Stensola, H. et al. The entorhinal grid map is discretized. *Nature* **492**, 72–78 (2012).
24. Buzsáki, György & Moser, E. I. Memory, navigation and theta rhythm in the hippocampal–entorhinal system. *Nat. Neurosci.* **16**, 130–138 (2013).
25. Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl Acad. Sci. USA* **81**, 3088–3092 (1984).
26. Marr, D., Willshaw, D. & McNaughton, B. *Simple Memory: A Theory for Archicortex* (Springer, 1991).
27. Skaggs, W., Knierim, J., Kudrimoti, H. & McNaughton, B. A model of the neural basis of the rat's sense of direction. *Adv. Neural Inf. Process. Syst.* **7**, 173–180 (1995).
28. Burgess, N., Recce, M. & O'Keefe, J. A model of hippocampal function. *Neural Netw.* **7**, 1065–1081 (1994).
29. 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. Rev.* **102**, 419 (1995).
30. Brun, V. H. et al. Place cells and place recognition maintained by direct entorhinal–hippocampal circuitry. *Science* **296**, 2243–2246 (2002).
31. 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).
32. Sreenivasan, S. & Fiete, I. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nat. Neurosci.* **14**, 1330–1337 (2011).
33. Yoon, Ki. Jung et al. Specific evidence of low-dimensional continuous attractor dynamics in grid cells. *Nat. Neurosci.* **16**, 1077–1084 (2013).
34. Yoon, K., Lewallen, S., Kinkhabwala, A. A., Tank, D. W. & Fiete, I. R. Grid cell responses in 1D environments assessed as slices through a 2D lattice. *Neuron* **89**, 1086–1099 (2016).
35. Solstad, T., Moser, E. I. & Einevoll, G. T. From grid cells to place cells: a mathematical model. *Hippocampus* **16**, 1026–1031 (2006).
36. 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).
37. Gardner, R. J. et al. Toroidal topology of population activity in grid cells. *Nature* **602**, 123–128 (2022).
38. Gardner, R. J., Lu, L., Wernle, T., Moser, May-Britt & Moser, E. I. Correlation structure of grid cells is preserved during sleep. *Nat. Neurosci.* **22**, 598–608 (2019).
39. O'Reilly, R. C., Bhattacharyya, R., Howard, M. D. & Ketz, N. Complementary learning systems. *Cogn. Sci.* **38**, 1229–1248 (2014).
40. Hardcastle, K., Ganguli, S. & Giocomo, L. M. Environmental boundaries as an error correction mechanism for grid cells. *Neuron* **86**, 827–839 (2015).
41. 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).
42. Keinath, A. T., Epstein, R. A. & Balasubramanian, V. Environmental deformations dynamically shift the grid cell spatial metric. *eLife* **7**, e38169 (2018).
43. Ocko, S. A., Hardcastle, K., Giocomo, L. M. & Ganguli, S. Emergent elasticity in the neural code for space. *Proc. Natl Acad. Sci. USA* **115**, E11798–E11806 (2018).
44. Chaudhuri, R. & Fiete, I. Bipartite expander hopfield networks as self-decoding high-capacity error correcting codes. *Adv. Neural Inf. Process. Syst.* **32**, 4175 (2019).
45. Whittington, James C. R. et al. The Tolman–Eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263 (2020).
46. Buzsáki, György & Tingley, D. Space and time: the hippocampus as a sequence generator. *Trends Cogn. Sci.* **22**, 853–869 (2018).
47. Aronov, D., Nevers, R. & Tank, D. W. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature* **543**, 719–722 (2017).
48. Killian, N. J., Potter, S. M. & Buffalo, E. A. Saccade direction encoding in the primate entorhinal cortex during visual exploration. *Proc. Natl Acad. Sci. USA* **112**, 15743–15748 (2015).
49. 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).
50. Neupane, S., Fiete, I. & Jazayeri, M. Mental navigation in the primate entorhinal cortex. *Nature* **630**, 704–711 (2024).
51. Krotov, D. and Hopfield, J. Large associative memory problem in neurobiology and machine learning. Preprint at https://doi.org/10.48550/arXiv.2008.06996 (2020).
52. 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).
53. Fiete, I. R., Burak, Y. & Brookings, T. What grid cells convey about rat location. *J. Neurosci.* **28**, 6858–6871 (2008).
54. Sharma, S., Chandra, S. & Fiete, I. Content addressable memory without catastrophic forgetting by heteroassociation with a fixed scaffold. In *39th International Conference on Machine Learning* 19658–19682 (PMLR, 2022).

55. Radhakrishnan, A., Belkin, M. & Uhler, C. Overparameterized neural networks implement associative memory. *Proc. Natl Acad. Sci. USA* **117**, 27162–27170 (2020).
56. 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).
57. Fyhn, M., Hafting, T., Treves, A., Moser, May-Britt & Moser, E. I. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature* **446**, 190–194 (2007).
58. Huszár, R., Zhang, Y., Blockus, H. & Buzsáki, György Preconfigured dynamics in the hippocampus are guided by embryonic birthdate and rate of neurogenesis. *Nat. Neurosci.* **25**, 1201–1212 (2022).
59. Bonnevie, T. et al. Grid cells require excitatory drive from the hippocampus. *Nat. Neurosci.* **16**, 309–317 (2013).
60. Almog, N. et al. During hippocampal inactivation, grid cells maintain synchrony, even when the grid pattern is lost. *eLife* **8**, e47147 (2019).
61. Hales, J. B. et al. Medial entorhinal cortex lesions only partially disrupt hippocampal place cells and hippocampus-dependent place memory. *Cell Rep.* **9**, 893–901 (2014).
62. 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).
63. Grieves, R. M., Wood, E. R. & Dudchenko, P. A. Place cells on a maze encode routes rather than destinations. *eLife* **5**, e15986 (2016).
64. 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. Neurosci.* **13**, 1433–1440 (2010).
65. Nadel, L. & Moscovitch, M. Memory consolidation, retrograde amnesia and the hippocampal complex. *Curr. Opin. Neurobiol.* **7**, 217–227 (1997).
66. Yadav, N., Toader, A. & Rajasethupathy, P. Thalamic and prefrontal contributions to an evolving memory. *Neuron* **112**, 1045–1059 (2024).
67. Tsoi, SauYee et al. Telencephalic outputs from the medial entorhinal cortex are copied directly to the hippocampus. *eLife* **11**, e73162 (2022).
68. Donato, F., Jacobsen, R. I., Moser, May-Britt & Moser, E. I. Stellate cells drive maturation of the entorhinal–hippocampal circuit. *Science* **355**, eaai8178 (2017).
69. 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. USA* **118**, e2018422118 (2021).
70. Teyler, T. J. & Rudy, J. W. The hippocampal indexing theory and episodic memory: updating the index. *Hippocampus* **17**, 1158–1169 (2007).
71. Treves, A. & Rolls, E. T. Computational analysis of the role of the hippocampus in memory. *Hippocampus* **4**, 374–391 (1994).
72. Alme, C. B. et al. Place cells in the hippocampus: eleven maps for eleven rooms. *Proc. Natl Acad. Sci. USA* **111**, 18428–18435 (2014).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

Reference [55] introduced the MESH associative memory architecture, leveraging a three-layer network to store numerous independent memory states. This architecture allowed 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, the 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. In addition, vVector-HaSH includes a sensory layer that is associatively linked to the hippocampal layer, as we describe in additional detail below. Activity propagation between regions occurs in sequential order and discrete time, a simplification of the oscillations and synaptic latencies that are hypothesized to gate this information flow. Most simulations involve discrete-valued inputs and grid cell activations, which we then later show can generalize to continuous space and activations.

We represent the MEC grid cells as outlined in ref. 73, 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 $g \in \{0, 1\}^{N_{\mathrm{g}}}$, where the $N_{\mathrm{g}} = \sum_M \lambda_M^2$. The continuous attractor recurrence in the grid layer[8] is represented by a module-wise winner-take-all dynamics, which we denote as CAN. This ensures that the equilibrium states of g always correspond to a valid grid-coding state.

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

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 $N_{\mathrm{patts}} = \prod_M \lambda_M^2$ spatial locations.

This layer of grid cells projects randomly onto the hippocampal layer, through a $N_{\mathrm{h}} \times N_{\mathrm{g}}$ random matrix W_{hg}, with each element drawn independently from a Gaussian distribution with a mean of zero and s.d. of $N(0, 1)$. 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

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

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 $h_{\vec{x}}$.

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

The dynamics of the hippocampal scaffold is then set up as

$$g(t+1) = \mathrm{CAN}[W_{\mathrm{gh}} h(t)] \tag{4}$$

$$h(t+2) = \mathrm{ReLU}[W_{\mathrm{hg}} g(t+1) - \theta] \tag{5}$$

These equations maintain each $g_{\vec{x}}$, $h_{\vec{x}}$ state as a fixed point of the recurrent dynamics, as we prove in Supplementary Information, section 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 pseudoinverse learned heteroassociative weights.

$$W_{\mathrm{hs}} = HS^{+} \tag{6}$$

and

$$W_{\mathrm{sh}} = SH^{+}, \tag{7}$$

where H is a $N_{\mathrm{h}} \times N_{\mathrm{patts}}$-dimensional matrix with columns as the predetermined hippocampal states $h_{\vec{x}}$, and S is a $N_{\mathrm{s}} \times N_{\mathrm{patts}}$-dimensional matrix with columns as the encoded sensory inputs to be stored at location $\vec{x}$. As discussed in the main text, the pseudoinverse computation can be performed through a biologically plausible iterative pseudoinverse learning rule[74,75]. However, 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 bidirectional inference of sensory inputs from grid states and vice versa:

$$h(t+1) = \mathrm{ReLU}[W_{\mathrm{hs}} s(t)] \tag{8}$$

$$g(t+2) = \mathrm{CAN}[W_{\mathrm{gh}} h(t+1)] \tag{9}$$

and

$$h(t+1) = \mathrm{ReLU}[W_{\mathrm{hg}} g(t) - \theta] \tag{10}$$

$$s(t+2) = \mathrm{sgn}[W_{\mathrm{sh}} h(t+1)] \tag{11}$$

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 in which sensory states are continuous valued (as in Fig. 3b, for example) the s reconstruction equation, equation (11) is replaced with simply $s(t+2) = W_{\mathrm{sh}} h(t+2)$.

Equations (1)–(11) 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. 6. In general, across all models, we assume that the relevant synapses are plastic during input presentation for memory storage, and are frozen during testing of memory retrieval.

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 equations (6) and (7).

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, right 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 equations (8)–(11), finally generating the recalled pattern s.

In all numerical examples that we consider in the main text we either construct random binary {−1, 1} patterns, or consider images from mini-imagenet (https://www.kaggle.com/datasets/whitemoon/miniimagenet). In particular, we took 3,600 images from the first 6 classes {‘house-finch’, ‘robin’, ‘triceratops’, ‘green-mamba’, ‘harvestman’ and ‘toucan’} and centre-cropped them to consider the middle 60 × 60 image and converted them to greyscale. We refer to this set of greyscale 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 and 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,d, 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 Extended Data Fig. 1, 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 Extended Data Fig. 1 for two and three modules are listed in Table 1. Similarly, the grid module periods for the data in Fig. 2c, left are listed in Table 2.

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 (that is, 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. Furthermore, we later demonstrate in Supplementary Information, section C.3 that the basins are indeed convex. Here grid module periods $\lambda = \{3, 4, 5\}$, number of grid cells $N_{\mathrm{g}} = 50$, and $N_{\mathrm{h}} = 400$ hippocampal cells were used. Probability that a given pattern remains within its correct basin was estimated by computing the fraction of runs in which 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, that is, 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 after iteration through equations (4) and (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 Supplementary Information, section C.4 for additional details of the ordering and the freedom of possibilities in this ordering), resulting in the strongest generalization (Supplementary Information, section 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 bidirectional pseudoinverse learning between grid and hippocampal states and construct this as a scaffold. This lack of structured correlations between grid and hippocampal population vectors (PVs) results in catastrophic forgetting, with no observed fixed points remaining once all scaffold states have been used for training.

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

λ	N_{g}	N_{patts}
2, 3	13	36
3, 4	25	144
4, 5	41	400
5, 6	61	900
6, 7	85	1,764
7, 8	113	3,136
1, 2, 3	14	36
2, 3, 5	38	900
3, 4, 5	50	3,600
4, 5, 7	90	19,600
5, 6, 7	110	44,100

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

λ	N_{g}	N_{patts}
7, 8	113	3,136
3, 5, 8	98	14,400
3, 4, 5, 7	99	176,400
1, 3, 4, 5, 7	100	176,400

All curves shown in Fig. 3c–f are averaged over five 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 Fig. 3b,e,h, grid module periods $\lambda = \{3, 4, 5\}$, $N_{\mathrm{g}} = 50$, $N_{\mathrm{s}} = 3{,}600$ was used. The total capacity of the network in this case is capped by $N_{\mathrm{patts}} = \prod_M \lambda_M^2 = 3{,}600$. For the other associative memory models[55,76–81] used in Fig. 3d, all shown networks have ~5 × 10^5 synapses. Number of nodes in these networks are as follows: (1) Hopfield network of size $N = 708$, synapses = N^2. (2) Pseudoinverse Hopfield network of size $N = 708$, synapses = N^2. (3) Hopfield network with bounded synapses was trained with Hebbian learning on sequentially seen patterns. Size of the network $N = 708$, synapses = N^2. (4) Sparse Hopfield network (with sparse inputs) with a network size of $N = 708$, synapses = N^2, sparsity = $100(1 - p)$. (5) Sparse Hopfield network. Size of the network N, synapse dilution κ, synapses = $\kappa \times N^2 = 10^5$. (6) Tail-biting overparameterized autoencoder with network layer sizes 900, 275, 38, 275, 900. Vector-HaSH uses $\lambda = \{2, 3, 5\}$, and layer sizes: $N_{\mathrm{g}} = 38$, $N_{\mathrm{h}} = 275$, $N_{\mathrm{s}} = 900$.

For stored patterns of size N, recall of an independent random vector of size N would appear to have a MI of ~$1/\sqrt{N}$, which when evaluating the total MI 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: $N_{\mathrm{g}} = 50$, $N_{\mathrm{h}} = 400$, $N_{\mathrm{s}} = 3{,}600$. 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}.

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. 8, 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 time-step. 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 bidirectional recall of grid states from sensory inputs and vice versa. Here we consider Vector-HaSH with $\lambda = \{3, 4, 5\}$, $N_g = 50$, $N_h = 400$, $N_s = 3{,}600$. 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 Fig. 4, we use Vector-HaSH with grid module periods $\lambda = \{3, 4, 5, 7\}$, $N_g = 99$, $N_h = 342$, $\gamma = 0.1$, and $\theta = 2.5$. The total capacity of this grid-coding space is $176{,}400 \approx 2 \times 10^5$. 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 pre-trained memory scaffold, wherein the W_{hg} and W_{gh} matrices have already been constructed and trained corresponding to equations (2) and (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 after 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 equations (6) and (7).

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 equation (8), 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 equation (8). 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 3,600-dimensional sensory landmarks in each room in a continual learning setting. Here we begin again with simply the pre-trained grid–hippocampal scaffold. As the ith 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 ith room, vVector-HaSH is queried on the current and all previous rooms as follows: for any completed room j (that is, $0 \leq j < i$), 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 MI 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 MI 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 bidirectional 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[72]. Dot product between PVs across all combinations of the 11 test rooms were computed. To construct the PVs, we record the activations of hippocampal cells for each of the 10 × 10 positions in the simulated room. We stack these into 100 composite PVs, 1 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 2 rooms, that is, 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 colour-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 cumulative distribution function. Similarly, the cumulative distribution functions for shuffled data are computed through the same procedure, but using shuffled data to compute the PV normalized dot products. Shuffled data are 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 1,000.

Extension of Vector-HaSH to continuous space

So far we have considered the grid states to be {0,1}-valued discretely varying modular one-hot states. This leads to a finite number of grid phases per module, and hence a finite number of grid PVs that can be exactly enumerated, leading to the wealth of theoretical advancements and results described above. To bring Vector-HaSH closer to biological realism, we constructed continuous-valued grid states (Fig. 4l), as a Gaussian bump of activity on a two-dimensional lattice of neurons with periodic boundary conditions, similar to the one-hot states on a periodic lattice considered earlier (compare with Fig. 2b). Continuous attractor dynamics were approximated through a circular mean to determine mean activity location, and reinitialization of a Gaussian bump centered at the calculated mean location. Since the number of phases in each module is now infinite (the Gaussian bump need not be centred on a neuron in the lattice) it is computationally challenging to demonstrate memory capacity results similar to our analysis for the discrete model above. As a proof of concept, we demonstrated landmark reconstruction from grid phases and vice versa in Fig. 4m.

For Fig. 4m, we used 3 grid modules, consisting of 81, 144 and 225 cells each. The Gaussian bump of activity in each module was constructed to have a standard deviation of 0.5.

We also used this continuous extension of Vector-HaSH in Fig. 6b,e–f. The results presented in Fig. 6f are computed similarly to those in ref. 60. We first compute the temporal correlation between every pair of grid cells before and after hippocampal activation. We compute the correlation between these temporal correlations (shown by the vertical red dashed line). We then generate 1,000 random shuffles of the temporal correlations post hippocampal activation, and use these shuffles to generate a control distribution of the correlation between the temporal correlations (shown as the null distribution in black).

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. 8, 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).

All networks in Fig. 5j were constructed to have approximately 5×10^5 synapses, with network parameters identical to those in Fig. 3d. Figure 5i,j considers random binary patterns, and Fig. 5g,h considers bw-mini-imagenet images.

Path learning. Learning associations from the hippocampal layer directly to the velocity inputs through pseudoinverse learning would result in perfect recall for only $N_{\mathrm{seq}} \leq N_{\mathrm{h}}$, 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. 5c,d, left, for example, we use a single hidden layer in the MLP with 250 nodes. The dynamics of the network are as follows:

$$\vec{v}(t) = \mathrm{MLP}[h(t)] \tag{12}$$

$$g(t+1) = \mathrm{PI}[g(t); \vec{v}(t)] \tag{13}$$

$$h(t+2) = \mathrm{ReLU}[W_{\mathrm{hg}} g(t+1) - \theta] \tag{14}$$

$$s(t+2) = \mathrm{sgn}[W_{\mathrm{sh}} h(t+2)] \tag{15}$$

Thus, when cued with a sensory state at the start of an episode, the sensory inputs to hippocampus reconstruct the corresponding hippocampal and grid states. Then, through the MLP, the hippocampal state projects to a low-dimensional velocity vector that is used to update the grid cells via path integration. From this updated grid state, the corresponding hippocampal state is constructed, which then reconstructs the next sensory pattern of the episode. The new hippocampal state also maps to the next velocity vector, that continues the iteration by updating the grid state. In this way, the memory scaffold along with the MLP successively construct grid and hippocampal states, and the heteroassociative weights to the sensory layer successively construct the memorized patterns of the episode.

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

$$W_{vs} = VS_r^+, \tag{16}$$

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

$$\vec{v}(t) = \mathrm{WTA}[W_{vs} s(t)] \tag{17}$$

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

$$h(t+2) = \mathrm{ReLU}[W_{\mathrm{hg}} g(t+1) - \theta] \tag{19}$$

$$s(t+2) = \mathrm{sgn}[W_{\mathrm{sh}} h(t+2)] \tag{20}$$

As argued in Supplementary Information, section D.9, this results in perfect sequence recall for $N_{\mathrm{seq}} \leq N_{\mathrm{s}}$, which can scale as the exponentially large capacity of the grid-coding space. Note that the results in Supplementary Information, section D.9 rely on S_{r} being a rank-ordered matrix. While this holds for random binary patterns through equation (20) applying a sign nonlinearity, this does not directly hold for continuous-valued sensory states, where no nonlinearity is necessary. In this case, we take the input sensory patterns, and apply an inverse sigmoid function to them before storage in the W_{sh} matrix. Then, we use equation (20) with the sign nonlinearity replaced with the sigmoid nonlinearity. This application of the inverse sigmoid and then the sigmoid ensures that the final recovered states correspond to the inital patterns, but the sensory states are recovered through a nonlinear readout.

Reproducing entorhinal–hippocampal phenomenology. For Fig. 6e,f we used the continuous extension of Vector-HaSH (see details in 'Extension of Vector-HaSH to continuous space'). We describe the methods details for other panels below.

Grid–hippocampal correlations. We follow in Fig. 6d a similar analysis to ref. 45. We consider two 10 × 10 rooms. Then, we choose a hippocampal cell that is active in both rooms at some location. Then, we calculate the fraction of grid cells that are active at both of these locations, shown in red. Then, we generate 100 shuffles of all place cell PVs, and generate a control distribution of the fraction of grid cells that are co-active with these shuffled place cells shown as the black histogram.

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), that is, 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, that is, W_{hs} and W_{sh}, via equations (6) and (7). 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 thus 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[62], route-dependent place cells[63], directional place fields in one-dimensional environments[64] and on directed routes in two-dimensional environments[82] in Fig. 6g–j and Supplementary Fig. 19; and of directional place fields in a radial eight-arm maze[82] in Supplementary Fig. 21. 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 contextual cues, 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. 6g–j and Supplementary Figs. 19 and Fig. 21, Vector-HaSH with $\lambda = \{3, 4, 5, 7\}$, $N_h = 500$, $N_g = 99$, $\theta = 2.5$ 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. 6h, we follow an analysis method similar to the analysis done on the experimental data[62]. The central stem is divided into four equal regions (Supplementary Fig. 22b), and the mean activation of every hippocampal cell is computed in each of the four regions. Supplementary Fig. 22c 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. 6h, 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 ~3.896%, and the percentage of hippocampal cells with differential firing was found to be ~96.103% in Vector-HaSH (using a threshold of 2 on the activation ratio).

Route encoding: In Supplementary Fig. 19a,c we employed an ensemble analysis approach mirroring that used in ref. 63 to validate if hippocampal cells demonstrate route-dependent activity. Our simulated session comprised 4 blocks, each representing one of 4 routes (0–3), with 11 trials per block. We performed ensemble analysis on the maze region common to all routes.

We compared the 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 PVs (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 Supplementary Fig. 20a, 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. Supplementary Fig. 20a,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 Supplementary Fig. 20b. We then estimated the 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 Supplementary Fig. 20a.

Supplementary Fig. 19c 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 Fig. 6j and Supplementary Figs. 19d and 21, the directionality index is defined similar to that defined for the experimental data analysis[63,64]. 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 Supplementary Fig. 24.

Multiple traces theory. In Fig. 6l–o, we consider Vector-HaSH with $\lambda = \{3, 4, 5\}$, $N_g = 50$, $N_h = 400$, $N_s = 3{,}600$, $\gamma = 0.6$, and $\theta = 0.5$. We use random binary patterns in Fig. 6n, right, o, and bw-mini-imagenet patterns in Fig. 6m,n, left. The results are averaged over 20 runs. For sensory inputs presented multiple times, the sensory-hippocampal weights are reinforced multiple times using online pseudoinverse learning rule[74], and the grid–hippocampal weights are reinforced multiple times using Hebbian learning (Fig. 6l). The W_{hs} weights are invariant to reinforcement due to the iterative pseudoinverse causing perfect hippocampal reconstruction from sensory inputs. 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 follow equations (8)–(11). Additional results from each layer of Vector-HaSH while testing the Multiple-Trace Theory are shown in Supplementary Fig. 16, right. Furthermore, Supplementary Fig. 16, 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. 6n, right and Fig. 6o.

Parameter values. Figure 2: across all panels: $\gamma = 0.6$, $\theta = 0.5$. Stable states counted if they can correct noise of magnitude 20% of the typical hippocampal state magnitude, and requiring dynamics to return to within 0.6% of the original hippocampal state. Figure 2e,f use three modules with $\lambda = \{3, 4, 5\}$ and $N_h = 400$.

Figure 3: across all panels: $\gamma = 0.6$, $\theta = 0.5$. In Fig. 3b,e,i, $\lambda = \{3, 4, 5\}$, $N_g = 50$, $N_s = 3{,}600$. In Fig. 3c,e,f, right,i, $N_h = 400$. Capacity is computed in Fig. 3f through injecting 2.5% noise in the sensory inputs, and demanding perfect (0 error) recall. In Fig. 3d, all shown networks have ~5×10^5 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: (1) Hopfield network of size $N = 708$, synapses = N^2. (2) Pseudoinverse Hopfield network of size $N = 708$, synapses = N^2. (3) Hopfield

network with bounded synapses was trained with Hebbian learning on sequentially seen patterns. Size of the network $N = 708$, synapses = N^2. (4) Sparse Hopfield network (with sparse inputs) with a network size of $N = 708$, synapses = N^2, sparsity = $100(1 - p)$. (5) Sparse Hopfield network. Size of the network N, synapse dilution κ, synapses = $\kappa \times N^2 = 10^5$. (6) Tail-biting overparameterized autoencoder[55] with network layer sizes 900, 275, 38, 275, 900.

Figure 4: across all panels: $\gamma = 0.1$, $\theta = 2.5$. Figure 4a,c used $\lambda = \{3, 4, 5\}$, $N_h = 400$ and $N_s = 3{,}600$. In Fig. 4m, we used 3 grid modules, consisting of 81, 144 and 225 cells each, and $N_h = 1{,}000$. The Gaussian bump of activity had standard deviation of 0.5. All other panels in Fig. 4 used $\lambda = \{3, 4, 5, 7\}$, $N_h = 342$. In Fig. 4c, we show the model 596 other landmarks before observing the 4 shown landmarks.

Figure 5a, left,b–e used $\lambda = \{5, 9, 13\}$, $\gamma = 0.6$, $\theta = 0.5$, $N_h = 500$. Figure 5a, right,g used $\lambda = \{3, 4, 5\}$, $N_h = 400$, $N_s = 36{,}00$. Hopfield network in Fig. 5h used 3,600 nodes. Figure 5g,h stored a sequence of length 1,000. Figure 5i also used $N_h = 400$. Figure 5a,c,d, left,e used 250 MLP nodes. Parameters used in Fig. 5j were identical to those used in Fig. 3d.

Figure 6: all panels used $\gamma = 0.1$, $\theta = 2.5$. Figure 6b,e,f used the continuous version of Vector-HaSH, using the same parameters as Fig. 4k,m. All other panels in Fig. 6 used $\lambda = \{3, 4, 5\}$, $N_h = 400$, $N_s = 3{,}600$.

Figure 7: all panels used $\lambda = \{4, 5, 7\}$, $N_h = 400$, $\gamma = 0.6$, $\theta = 0.5$. The mnemonic input layer had 3,600 nodes.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Data were collecting by running the codes available at https://github.com/FieteLab.

Code availability

Codes used to run the model and analyse data are available at https://github.com/FieteLab.

73. Yim, ManYi, Sadun, L. A., Fiete, I. R. & Taillefumier, T. Place-cell capacity and volatility with grid-like inputs. *eLife* **10**, e62702 (2021).
74. Tapson, J. & van Schaik, A. Learning the pseudoinverse solution to network weights. *Neural Netw.* **45**, 94–100 (2013).
75. O'Reilly, R. C. Six principles for biologically based computational models of cortical cognition. *Trends Cogn. Sci.* **2**, 455–462 (1998).
76. Personnaz, L., Guyon, I. & Dreyfus, G. Information storage and retrieval in spin-glass like neural networks. *J. Physique Lettres* **46**, 359–365 (1985).
77. Personnaz, L., Guyon, I. & Dreyfus, G. Collective computational properties of neural networks: new learning mechanisms. *Phys. Rev. A* **34**, 4217 (1986).
78. Parisi, G. A memory which forgets. *J. Phys. A* **19**, L617 (1986).
79. Fusi, S. & Abbott, L. F. Limits on the memory storage capacity of bounded synapses. *Nat. Neurosci.* **10**, 485–493 (2007).
80. Tsodyks, M. V. & Feigel'man, M. V. The enhanced storage capacity in neural networks with low activity level. *Europhysics Lett.* **6**, 101 (1988).
81. Dominguez, D., Koroutchev, K., Serrano, E. & Rodríguez, F. B. Information and topology in attractor neural networks. *Neural Comput.* **19**, 956–973 (2007).
82. 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).

Acknowledgements This work was supported by ONR award N00014-19-1-2584, by NSF-CISE award IIS-2151077 under the Robust Intelligence programme, by the ARO-MURI award W911NF-23-1-0277, by the Simons Foundation SCGB programme 1181110, and the K. Lisa Yang ICoN Center.

Author contributions Conceptualization: S.C., S.S., R.C. and I.F. Funding acquisition: I.F. Writing: S.C., S.S. and I.F. Coding and analysis: S.C. and S.S.

Competing interests The authors declare no competing interests

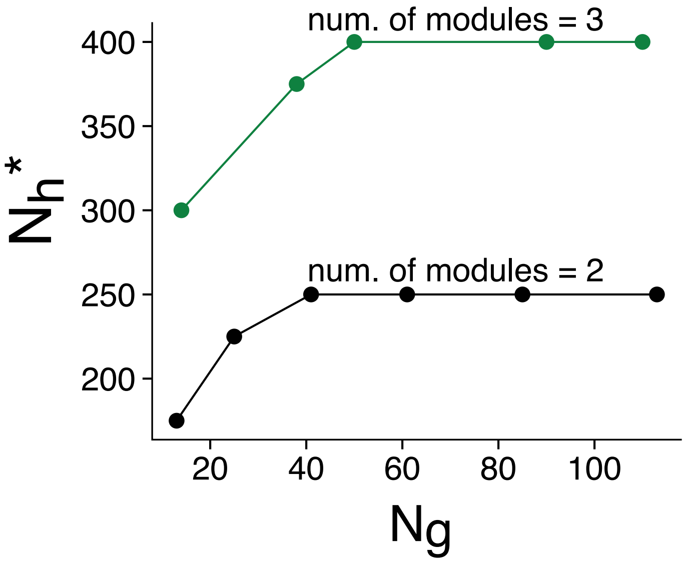
Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-024-08392-y.

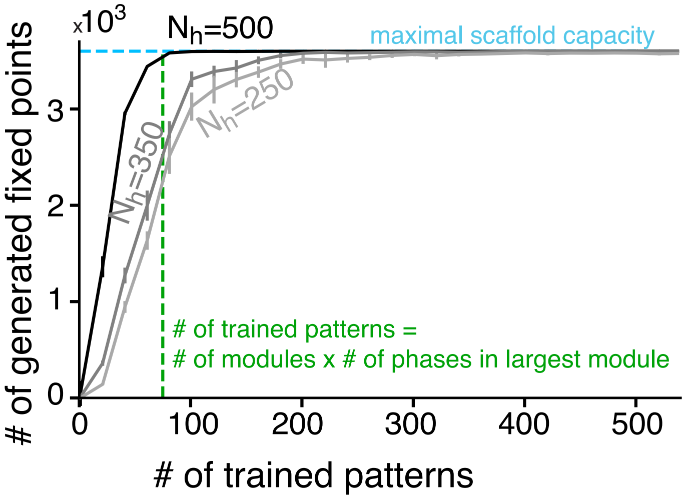
Correspondence and requests for materials should be addressed to Ila Fiete.

Peer review information *Nature* thanks Dori Derdikman, Christian Machens and Cristina Savin for their contribution to the peer review of this work. Peer review reports are available.

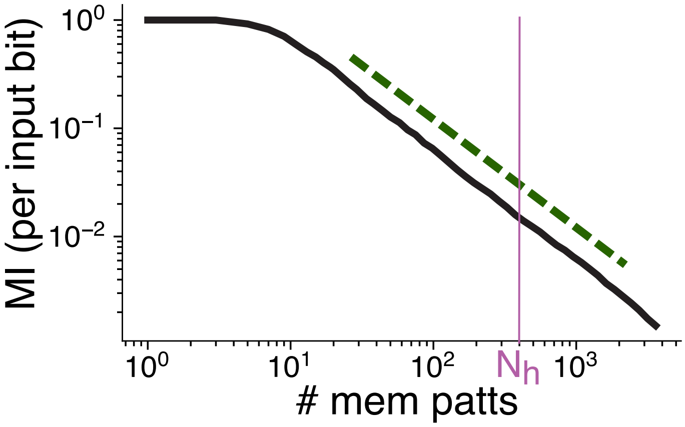
Reprints and permissions information is available at http://www.nature.com/reprints.



Extended Data Fig. 1 | 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.



Extended Data Fig. 2 | 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 expectation of $M \times K_{max}$, as proved in SI Sec. C.4.



Extended Data Fig. 3 | 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 $\lambda = \{3, 4, 5\}$.

nature portfolio

Corresponding author(s): Ila Fiete

Last updated by author(s): Sep 9, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
□	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	□	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	□	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	□	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	□	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	We wrote custom Python codes to perform numerical simulations of the constructed model. The code will be available at https://github.com/FieteLab
Data analysis	We wrote custom Python codes to analyze the results obtained by the numerical simulations. The code will be available at https://github.com/FieteLab

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

We did not collect any new data beyond the conducted numerical simulations. The code and data used will be made available at https://github.com/FieteLab

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	n/a
Reporting on race, ethnicity, or other socially relevant groupings	n/a
Population characteristics	n/a
Recruitment	n/a
Ethics oversight	n/a

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences □ Behavioural & social sciences □ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We have not collected or analyzed data beyond the performed numerical simulations. Sample sizes of numerical simulations were chosen to be as large as possible given computational feasibility
Data exclusions	No data was excluded from the analyses
Replication	We have not collected or analyzed data beyond the performed numerical simulations. n/a
Randomization	We have not collected or analyzed data beyond the performed numerical simulations. n/a
Blinding	We have not collected or analyzed data beyond the performed numerical simulations.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study	
☒	□	Antibodies
☒	□	Eukaryotic cell lines
☒	□	Palaeontology and archaeology
☒	□	Animals and other organisms
☒	□	Clinical data
☒	□	Dual use research of concern
☒	□	Plants

Methods

n/a	Involved in the study	
☒	□	ChIP-seq
☒	□	Flow cytometry
☒	□	MRI-based neuroimaging

Plants

Seed stocks

Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.

Novel plant genotypes

Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.

Authentication

Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.