nature communications

Article

https://doi.org/10.1038/s41467-024-44877-0

Distinguishing examples while building concepts in hippocampal and artificial networks

Received: 9 March 2023

Accepted: 3 January 2024

Published online: 20 January 2024

Check for updates

Louis Kang[1,2] & Taro Toyoizumi[3,4]

The hippocampal subfield CA3 is thought to function as an auto-associative network that stores experiences as memories. Information from these experiences arrives directly from the entorhinal cortex as well as indirectly through the dentate gyrus, which performs sparsification and decorrelation. The computational purpose for these dual input pathways has not been firmly established. We model CA3 as a Hopfield-like network that stores both dense, correlated encodings and sparse, decorrelated encodings. As more memories are stored, the former merge along shared features while the latter remain distinct. We verify our model's prediction in rat CA3 place cells, which exhibit more distinct tuning during theta phases with sparser activity. Finally, we find that neural networks trained in multitask learning benefit from a loss term that promotes both correlated and decorrelated representations. Thus, the complementary encodings we have found in CA3 can provide broad computational advantages for solving complex tasks.

The hippocampus underlies our ability to form episodic memories, through which we can recount personally experienced events from our daily lives[1]. In particular, the subfield CA3 is believed to provide this capability as an autoassociative network[2–4]. Its pyramidal cells contain abundant recurrent connections exhibiting spike-timing-dependent plasticity[5,6]. These features allow networks to perform pattern completion and recover stored patterns of neural activity from noisy cues. Sensory information to be stored as memories arrives to CA3 via the entorhinal cortex (EC), which serves as the major gateway between hippocampus and neocortex (Fig. 1A). Neurons from layer II of EC project to CA3 via two different pathways[7]. First, they synapse directly onto the distal dendrites of CA3 pyramidal cells through the perforant path (PP). Second, before reaching CA3, perforant path axons branch towards the dentate gyrus (DG) and synapse onto granule cells. Granule cell axons form the mossy fibers (MF) that also synapse onto CA3 pyramidal cells, though at more proximal dendrites.

Along these pathways, information is transformed by each projection in addition to being simply relayed. DG sparsifies encodings from EC by maintaining high inhibitory tone across its numerous neurons[8]. Sparsification in feedforward networks generally decorrelates activity patterns as well[9–13]. The sparse, decorrelated nature of DG encodings is preserved by the MF pathway because its connectivity is also sparse; each CA3 pyramidal cell receives input from only ≈50 granule cells[14]. In contrast, PP connectivity is dense with each CA3 pyramidal cell receiving input from ≈4000 EC neurons[14], so natural correlations between similar sensory stimuli should be preserved. Thus, CA3 appears to receive two encodings of the same sensory information with different properties: one sparse and decorrelated through MF and the other dense and correlated through PP. What is the computational purpose of this dual-input architecture? Previous theories have proposed that the MF pathway is crucial for pattern separation during memory storage, but retrieval is predominantly mediated by the PP pathway and can even be hindered by MF inputs[15–17]. In these models, MF and PP encodings merge during storage and one hybrid pattern per memory is recovered during retrieval.

[1]Neural Circuits and Computations Unit, RIKEN Center for Brain Science, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan. [2]Graduate School of Informatics, Kyoto University, 36-1 Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan. [3]Laboratory for Neural Computation and Adaptation, RIKEN Center for Brain Science, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan. [4]Graduate School of Information Science and Technology, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. e-mail: louis.kang@riken.jp

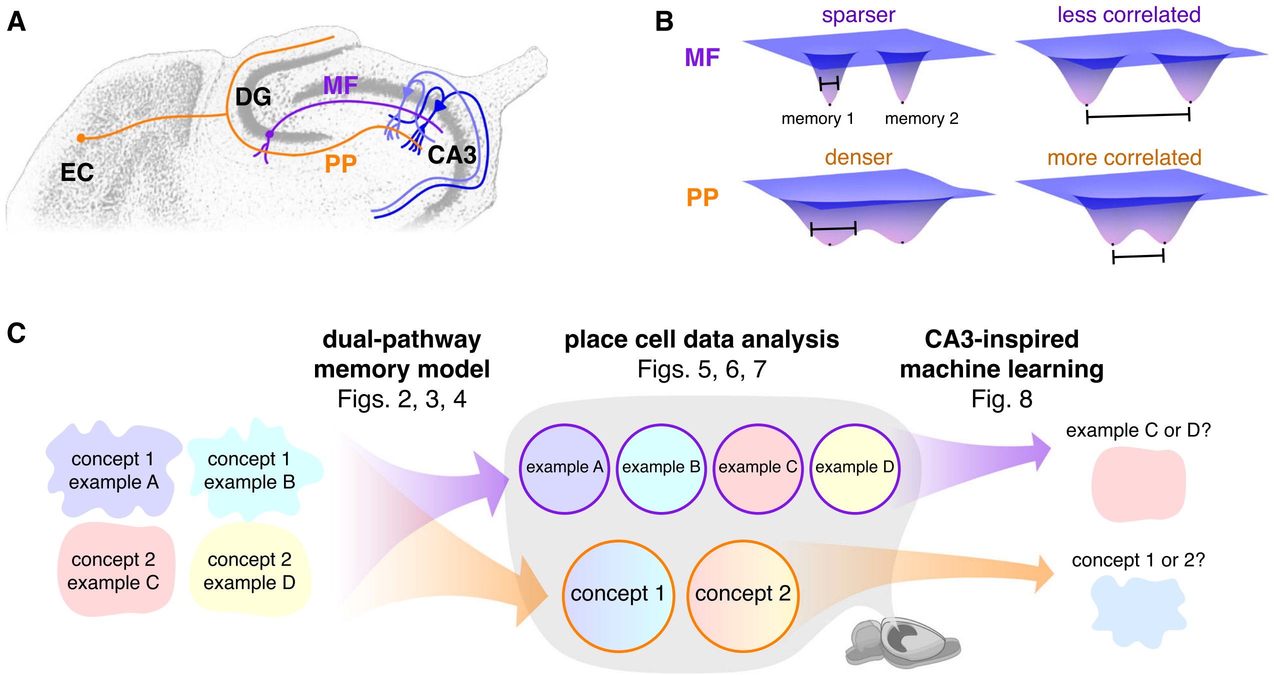


Fig. 1 | Overview and motivation. A Entorhinal cortex (EC) projects to CA3 directly via the perforant path (PP, orange) as well as indirectly through the dentate gyrus (DG) via mossy fibers (MF, purple). Adapted from Rosen GD, Williams AG, Capra JA, Connolly MT, Cruz B, Lu L, Airey DC, Kulkarni K, Williams RW (2000) The Mouse Brain Library @ https://www.mbl.org. Int Mouse Genome Conference 14: 166. https://www.mbl.org. **B** MF memory encodings are believed to be sparser and less correlated compared to PP encodings. In an autoassociative network, attractor basins of the former tend to remain separate and those of the latter tend to merge. **C** By modeling hippocampal networks, we first predict that MF and PP encodings in CA3 can respectively maintain distinctions between memories and generalize across them (Figs. 2–4). By analyzing publicly available neural recordings, we then detect signatures of these encoding properties in rat CA3 place cells (Figs. 5–7). By training artificial neural networks, we finally demonstrate that these encoding types are suited to perform the complementary tasks of example discrimination and concept classification (Fig. 8).

Instead, we consider the possibility that CA3 can store both MF and PP encodings for each memory and retrieve either of them. Inhibitory tone selects between the two; with a higher activity threshold, sparser MF patterns are more likely to be recovered, and the opposite holds for denser PP patterns. By encoding the same memory in two different ways, each can be leveraged for a different computational purpose. Conceptually, in terms of energy landscapes, sparser patterns have narrower attractor basins than denser patterns because fewer neurons actively participate (Fig. 1B). Moreover, less correlated patterns are located farther apart compared to more correlated patterns. Thus, MF energy basins tend to remain separate with barriers between them, a property called pattern separation that maintains distinctions between similar memories and is known to exist in DG[18–20]. In contrast, PP energy basins tend to merge, which enables the clustering of similar memories into concepts. This proposed ability for CA3 to recall both individual experiences and generalizations across them would explain observed features of hippocampal function. For instance, remembering the details of a recent visit with an acquaintance is an example of hippocampus-dependent episodic memory[1,21]. Meanwhile, hippocampal neurons can also generalize over your visits and respond to many different representations of your acquaintance, including previously unseen photographs or her name in spoken or written form[22,23].

To instantiate these ideas, we constructed a model for EC, DG, and CA3 in which CA3 stores both MF and PP encodings of each memory (Fig. 1C). We observe that MF encodings remain distinct, whereas PP encodings perform concept learning by merging similar memories. Our model predicts relationships between coding properties and network sparsity across phases of the theta oscillation, which modulates inhibitory tone in the hippocampal region. We tested these predictions across two publicly available datasets[24,25], and each analysis reveals that tuning of CA3 neurons is sharper during sparse theta phases and broader during dense phases. This supports our model and enriches our understanding of phase coding in hippocampus. While our model does not include CA1, we present comparative experimental analyses for this subfield in various Supplementary Figures. Beyond asserting the presence of complementary encodings in CA3, we demonstrate that they can offer functional advantages. Applying inspiration from our CA3 model and data analysis toward machine learning, we introduced a plug-and-play loss function that endows artificial neural networks with both correlated, PP-like and decorrelated, MF-like representations. These networks can perform better in multitask learning compared to networks with single representation types, which suggests a promising strategy for helping neural networks to solve complex tasks. While the essential components of our networks are explained in the Results section, their full descriptions and justifications are provided in the Methods section with parameter values in Table 1.

Results

MF encodings remain distinct while PP encodings build concepts in our model for CA3

We model how representations of memories are transformed along the two pathways from EC to CA3 and then how the resultant encodings are stored and retrieved in CA3. First, we focus on the transformations between memories and their CA3 encodings. The sensory inputs whose encodings serve as memories in our model are FashionMNIST images[26], each of which is an example belonging to one of three concepts: sneakers, trousers, and coats (Fig. 2A). They are converted to neural activity patterns along each projection from EC to CA3 (Fig. 2B). Our neurons are binary with activity values of 0 or 1. Each image $\mathbf{i}_{\mu\nu}$ representing example ν in concept μ is first encoded by EC using a binary autoencoder, whose middle hidden layer activations represent the patterns $\mathbf{x}_{\mu\nu}^{\text{EC}}$ (Fig. 2C). Only 10% of the neurons are allowed to be active, so the representation is sparse, and there are more EC neurons than image pixels, so the representation is overcomplete; sparse, overcomplete encoding models and autoencoder neural networks are common unsupervised models for natural image processing[27–30].

Table 1 | Key hippocampus model parameters and their values unless otherwise noted

Parameter	Value	See also
number of concepts	3	Fig. 2A
examples stored per concept s	1–100	Fig. 3
EC network size N_{EC}	1024	Eqs. (5) and (6), Fig. 2C, D
DG network size N_{DG}	8192	Eq. (6), Fig. 2D
CA3 network size N_{CA3}	2048	Eqs. (6) and (8), Figs. 2D and 3
EC pattern density a_{EC}	0.1	Eqs. (5) and (6), Fig. 2C, D
DG pattern density a_{DG}	0.005	Eq. (6), Fig. 2D
MF pattern density a_{MF}	0.02	Eq. (6), Fig. 2D
PP pattern density a_{PP}	0.2	Eq. (6), Fig. 2D
EC pattern correlation ρ_{EC}	0.15	Eqs. (5) and (6), Fig. 2C, D
DG pattern correlation ρ_{DG}	0.02	Eq. (6), Fig. 2D
MF pattern correlation ρ_{MF}	0.01	Eq. (6), Fig. 2D
PP pattern correlation ρ_{PP}	0.09	Eq. (6), Fig. 2D
PP pattern strength ζ	0.1	Eq. (8), Fig. 3A
fraction of neurons flipped to form cue	0.01	Fig. 3B
rescaled threshold θ'	0–0.5	Eq. (11), Fig. 3C
rescaled inverse temperature β'	100	Eq. (11)

From EC, we produce DG, MF, and PP encodings with random, binary, and sparse connectivity matrices between presynaptic and postsynaptic regions (Fig. 2D), i.e., from EC to DG, from DG to CA3 via MF, and from EC to CA3 via PP. Each matrix transforms presynaptic patterns $\mathbf{x}_{\mu\nu}^{\text{pre}}$ into postsynaptic inputs, which are converted into postsynaptic patterns $\mathbf{x}_{\mu\nu}^{\text{post}}$ at a desired density using a winners-take-all approach. That is, the postsynaptic neurons receiving the largest inputs are set to 1 and the others are set to 0. We define *density* to be the fraction of active neurons, so lower values correspond to sparser patterns. Enforcing a desired postsynaptic pattern density is equivalent to adjusting an activity threshold. At CA3, two encodings for each image converge: $\mathbf{x}_{\mu\nu}^{\text{MF}}$ with density 0.02 and $\mathbf{x}_{\mu\nu}^{\text{PP}}$ with density 0.2. Not only are MF patterns sparser, they are less correlated with average correlation 0.01, compared to a corresponding value of 0.09 for PP patterns. Such an association between sparsification and decorrelation has been widely reported across many theoretical models and brain regions[10–13], and it is also captured by our model. Decreasing postsynaptic pattern density (sparsification) correspondingly decreases the postsynaptic correlation (decorrelation) for any presynaptic statistics (Fig. 2E). We contribute further insight by deriving an explicit mathematical formula that connects densities and correlations of patterns in presynaptic and postsynaptic networks:

$$\rho_{\text{post}} = \frac{\Gamma\left[\sqrt{2}\operatorname{erfc}^{-1}(2a_{\text{post}}), a_{\text{pre}} + \rho_{\text{pre}} - a_{\text{pre}}\rho_{\text{pre}}\right] - a_{\text{post}}^2}{a_{\text{post}}(1 - a_{\text{post}})},$$

$$\text{where} \quad \Gamma[\phi,\sigma] \equiv \frac{1}{2\pi}\int_{\arccos\sigma}^{\pi} d\psi \, \exp\left[-\frac{\phi^2}{1+\cos\psi}\right] \qquad (1)$$

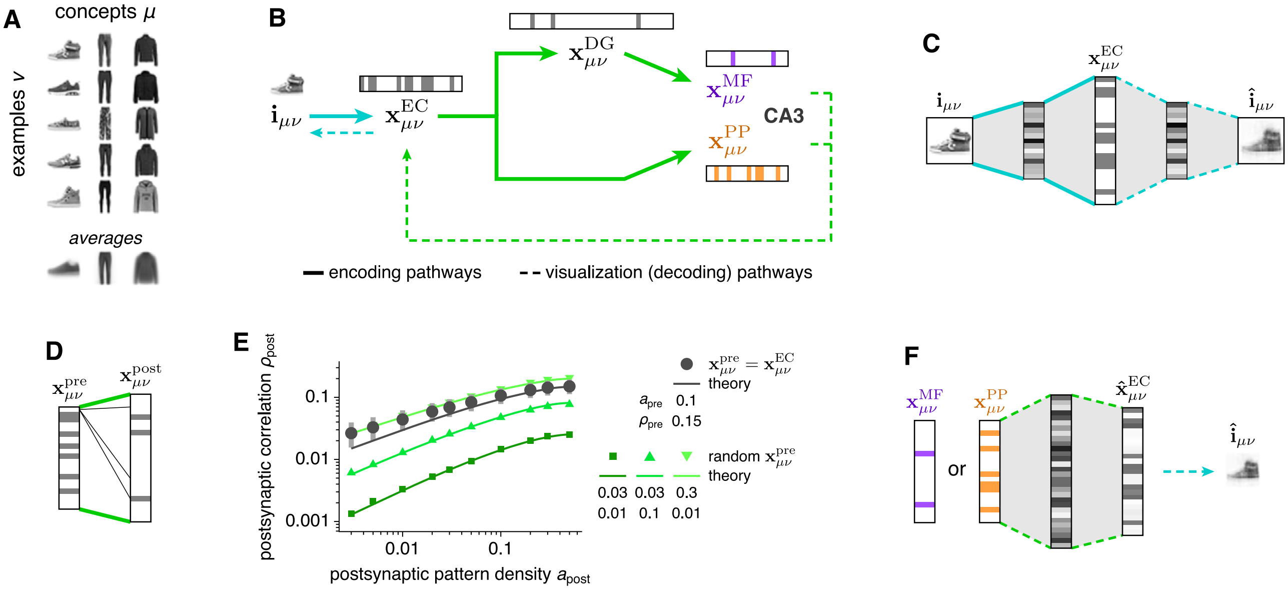

A
concepts μ
examples ν
averages
B
$\mathbf{i}_{\mu\nu}$
$\mathbf{x}_{\mu\nu}^{\text{EC}}$
$\mathbf{x}_{\mu\nu}^{\text{DG}}$
$\mathbf{x}_{\mu\nu}^{\text{MF}}$
$\mathbf{x}_{\mu\nu}^{\text{PP}}$
CA3
encoding pathways
visualization (decoding) pathways
C
$\mathbf{i}_{\mu\nu}$
$\mathbf{x}_{\mu\nu}^{\text{EC}}$
$\hat{\mathbf{i}}_{\mu\nu}$
D
$\mathbf{x}_{\mu\nu}^{\text{pre}}$
$\mathbf{x}_{\mu\nu}^{\text{post}}$
E
postsynaptic correlation ρ_{post}
0.1
0.01
0.001
0.01
0.1
postsynaptic pattern density a_{post}
$\mathbf{x}_{\mu\nu}^{\text{pre}} = \mathbf{x}_{\mu\nu}^{\text{EC}}$
theory
a_{pre} 0.1
ρ_{pre} 0.15
random $\mathbf{x}_{\mu\nu}^{\text{pre}}$
theory
0.03 0.03 0.3
0.01 0.1 0.01
F
$\mathbf{x}_{\mu\nu}^{\text{MF}}$
or
$\mathbf{x}_{\mu\nu}^{\text{PP}}$
$\hat{\mathbf{x}}_{\mu\nu}^{\text{EC}}$
$\hat{\mathbf{i}}_{\mu\nu}$

Fig. 2 | We model the transformation of memory representations along hippocampal pathways; MF and PP encodings of the same memories converge at CA3. A Memories are FashionMNIST images, each of which is an example of a concept. **B** Overview of model pathways. Encoding pathways correspond to the biological architecture in Fig. 1A. Decoding pathways are used to visualize CA3 activity and are not intended to have biological significance. **C** We use an autoencoder with a binary middle layer to transform each memory $\mathbf{i}_{\mu\nu}$ into an EC pattern $\mathbf{x}_{\mu\nu}^{\text{EC}}$. **D** From EC to CA3, we use random binary connectivity matrices to transform each presynaptic pattern $\mathbf{x}_{\mu\nu}^{\text{pre}}$ to a postsynaptic pattern $\mathbf{x}_{\mu\nu}^{\text{post}}$. **E** Enforcing sparser postsynaptic patterns in **D** promotes decorrelation. Dark gray indicates use of $\mathbf{x}_{\mu\nu}^{\text{EC}}$ as presynaptic patterns. Points indicate means and bars indicate standard deviations over 8 random connectivity matrices. Green indicates randomly generated presynaptic patterns at various densities a_{pre} and correlations ρ_{pre}. Theoretical curves depict Eq. (1). **F** To visualize CA3 encodings, we pass them through a feed-forward network trained to produce the corresponding $\mathbf{x}_{\mu\nu}^{\text{EC}}$ for each $\mathbf{x}_{\mu\nu}^{\text{MF}}$ and $\mathbf{x}_{\mu\nu}^{\text{PP}}$. Images are then decoded using the autoencoder in **C**. Source data are provided as a Source Data file.

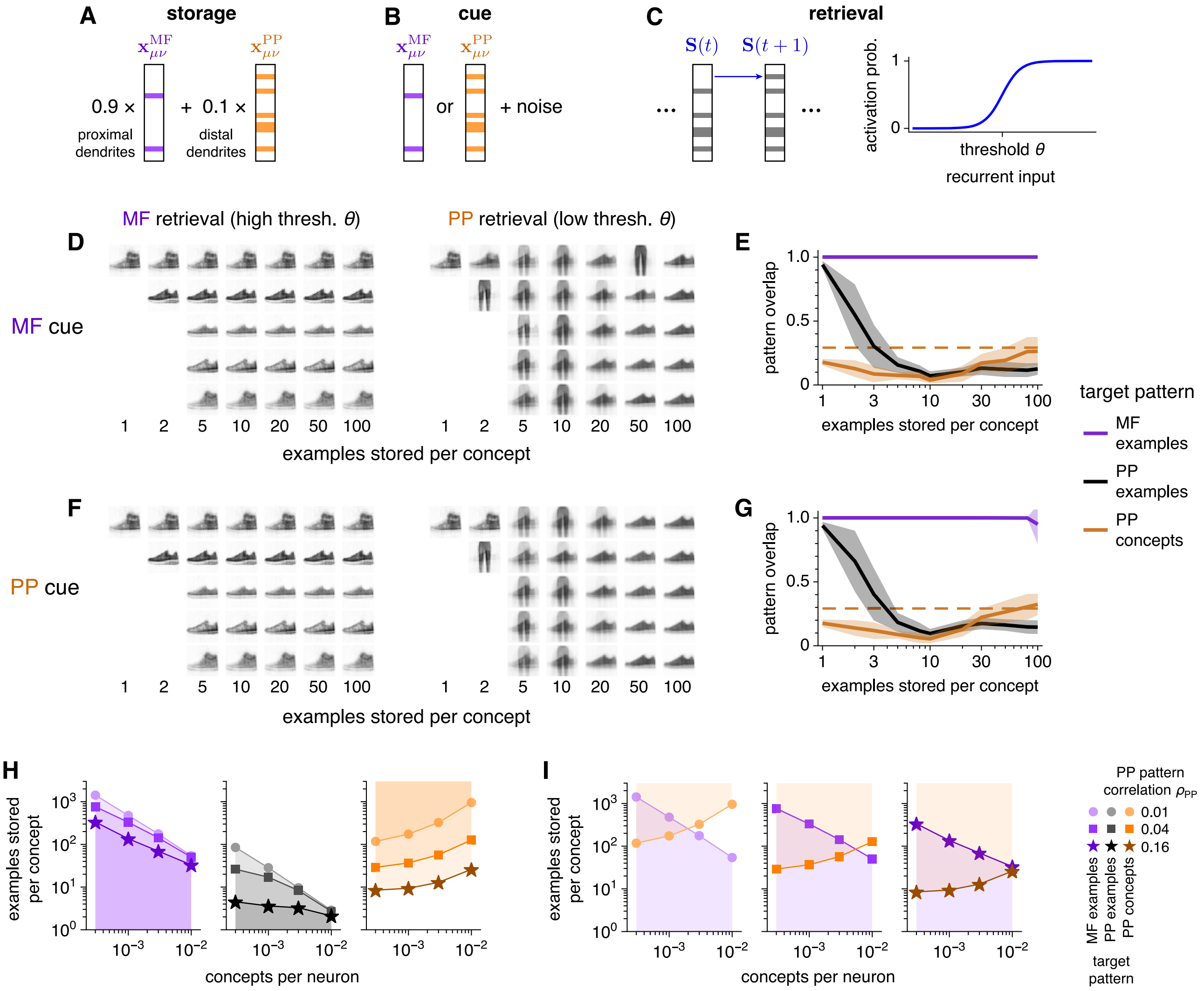


Fig. 3 | We model CA3 to store both MF and PP encodings of the same memories; MF examples remain distinct while PP examples build concept representations. **A**–**C** Overview of the Hopfield-like model for CA3. **A** We store linear combinations of MF and PP encodings, with greater weight on the former because MF inputs are stronger. **B** Retrieval begins by initializing the network to a stored pattern corrupted by flipping the activity of randomly chosen neurons. **C** During retrieval, the network is asynchronously updated with a threshold θ that controls the desired sparsity of the recalled pattern. **D**, **E** Retrieval behavior using MF cues. Examples from the three concepts depicted in Fig. 2A are stored. **D** Visualizations of retrieved patterns. MF encodings, retrieved at high θ, maintain distinct representations of examples. PP encodings, retrieved at low θ, merge into concept representations as more examples are stored (compare with average image in Fig. 2A). **E** Overlap of retrieved patterns with target patterns: MF examples, PP examples, or PP concepts defined by averaging over PP examples and binarizing (Methods). Solid lines indicate means, shaded regions indicate standard deviations, and the dashed orange line indicates the theoretically estimated maximum value for concept retrieval (Methods). In all networks, up to 30 cues are tested. **F**, **G** Similar to **D**, **E**, but using PP cues. **H** Network capacities computed using random MF and PP patterns instead of FashionMNIST encodings. Shaded regions indicate regimes of high overlap between retrieved patterns and target patterns (Supplementary Information). MF patterns have density 0.01 and correlation 0. PP patterns have density 0.5. **I** Similar to **H**, but overlaying capacities for MF examples and PP concepts to highlight the existence of regimes in which both can be recovered. Source data are provided as a Source Data file.

and erfc^{-1} is the inverse complementary error function. In other words, given the density a_{pre} and correlation ρ_{pre} of the presynaptic patterns and the desired density a_{post} of the postsynaptic patterns, the postsynaptic correlation ρ_{post} is determined. Equation (1) is remarkable in that only these four quantities are involved, revealing that at least in some classes of feedforward networks, other parameters such as network sizes, synaptic density, and absolute threshold values do not contribute to decorrelation. It is derived in Supplementary Information, and its behavior is further depicted in Supplementary Fig. 2B, C.

Ultimately, the encoding pathways in Fig. 2C–E provide CA3 with a sparse, decorrelated $\mathbf{x}_{\mu\nu}^{\mathrm{MF}}$ and a dense, correlated $\mathbf{x}_{\mu\nu}^{\mathrm{PP}}$ for each memory, in accordance with our biological understanding (Fig. 1A, B). Next, we aim to store these patterns in an autoassociative model of CA3. Before doing so, we develop visualization pathways that decode CA3 representations back into images, so memory retrieval can be intuitively evaluated. This is accomplished by training a continuous-valued feedforward network to associate each MF and PP pattern with its corresponding EC pattern (Fig. 2F). From there, the reconstructed EC pattern can be fed into the decoding half of the autoencoder in Fig. 2C to recover the image encoded by CA3. These decoding pathways are for visualization only and are not designed to mimic biology, although there may be parallels with the neocortical output pathway from CA3 to CA1 and deep layers of EC[7]. The neuroanatomical connectivity of CA1 is more complex and includes temporoammonic inputs from EC as well as strong secondary outputs through the subiculum, which also reciprocally connects with EC.

Now, we model memory storage in the CA3 autoassociative network. For each example ν in concept μ, its MF encoding $\mathbf{x}_{\mu\nu}^{\mathrm{MF}}$ arrives at the proximal dendrites and its PP encoding $\mathbf{x}_{\mu\nu}^{\mathrm{PP}}$ arrives at the distal dentrites of CA3 pyramidal cells (Fig. 3A). The relative strength of PP inputs is weaker because PP synapses are located more distally and are observed to be much weaker than MF synapses, which are even called detonator synapses[7,31,32]. The inputs are linearly summed and stored in a Hopfield-like network[33], with connectivity

$$W_{ij} \sim \sum_{\mu\nu} \left(0.9\,x_{\mu\nu i}^{\mathrm{MF}} + 0.1\,x_{\mu\nu i}^{\mathrm{PP}}\right)\left(0.9\,x_{\mu\nu j}^{\mathrm{MF}} + 0.1\,x_{\mu\nu j}^{\mathrm{PP}}\right), \qquad (2)$$

where i and j are respectively postsynaptic and presynaptic neurons. Equation (2) captures the most crucial terms in W_{ij}; see Methods for the full expression. While we assume linear summation between $\mathbf{x}_{\mu\nu}^{\mathrm{MF}}$ and $\mathbf{x}_{\mu\nu}^{\mathrm{PP}}$ for simplicity, integration of inputs across CA3 dendritic compartments is known to be nonlinear[17,34,35]. Moreover, sublinear summation can also arise from a temporal offset between MF and PP inputs, in which case changes in synaptic weights across pathways could be weaker than those within the same pathway according to spike-timing-dependent plasticity[5,6]. In Supplementary Fig. 3F, we show that network behavior can be maintained when nonlinearity is introduced.

In previous models, CA3 would retrieve only MF encodings, only PP encodings, or only the activity common between MF–PP pairs[15–17]. We assess the ability of the network to retrieve either $\mathbf{x}_{\mu\nu}^{\mathrm{MF}}$ or $\mathbf{x}_{\mu\nu}^{\mathrm{PP}}$ using either encoding as a cue (Fig. 3B). Each cue is corrupted by flipping randomly chosen neurons between active and inactive and is set as the initial network activity. During retrieval, the network is asynchronously updated via Glauber dynamics[36]. That is, at each simulation timestep, one neuron is randomly selected to be updated (Fig. 3C). If its total input from other neurons exceeds a threshold θ, then it is more likely to become active; conversely, subthreshold total input makes silence more likely. The width of the sigmoid function in Fig. 3C determines the softness of the threshold. A large width implies that activation and silence are almost equally likely for recurrent input near threshold. A small width implies that activation is almost guaranteed for recurrent input above threshold and almost impossible for input below threshold. See Methods for the full expression of this update rule.

The threshold θ represents the general inhibitory tone of CA3 and plays a key role in retrieval. At high θ, neural activity is disfavored, so we expect the network to retrieve the sparser, more strongly stored MF encoding of the cue. Upon lowering θ, more neurons are permitted to activate, so those participating in the denser, more weakly stored PP encoding should become active as well. Because our neurons are binary, active neurons in either the MF or the PP encoding would have the same activity level of 1, even though their connectivity strengths differ. This combined activity of both encodings is almost the same as the PP encoding alone, which contains many more active neurons. Thus, we expect the network to approximately retrieve the PP encoding at low θ.

Figure 3 D–G illustrates the central behavior of our CA3 model; see Supplementary Fig. 3A, B for trouser and coat visualizations, which behave similarly to the sneaker visualizations shown here. First using MF encodings as cues, we seek to retrieve either MF or PP encodings by respectively setting a high or low threshold. As we load the network with increasingly more stored examples, distinct MF examples can consistently be retrieved with high threshold (Fig. 3D). Meanwhile, retrieval of PP examples with low threshold fails above 1–2 examples stored per concept. At large example loads, the network again retrieves a sneaker memory when cued with sneaker examples. However, this memory is the same for all sneaker cues and appears similar to the average image over all sneaker examples (Fig. 2A), which captures common sneaker features (Supplementary Fig. 2A). Thus, the network is retrieving a representation of the sneaker *concept*. Notably, concepts are never directly presented to the network; instead, the network builds them through the unsupervised accumulation of correlated examples. The retrieval properties visualized in Fig. 3D are quantified in Fig. 3E by computing the overlap between retrieved and target patterns. Across all example loads shown, retrieved MF patterns overlap with target examples. As example load increases, retrieved PP patterns transition from encoding examples to representing concepts. We define the target pattern $\mathbf{x}_{\mu}^{\mathrm{PP}}$ for a PP concept μ by activating the most active neurons across PP examples within that concept until the PP density is reached, and the dashed line in Fig. 3E coarsely estimates the largest overlap achievable (Methods).

The network capabilities observed for MF cues are preserved when we instead use PP cues (Fig. 3F, G) or cues combining the neurons active in either encoding (Supplementary Fig. 3C–E); again, these latter two are similar because MF encodings are sparse. Thus, retrieval behavior is driven largely by the level of inhibition rather than the encoding type of cues. This feature implies that our model is agnostic to whether memory retrieval in the hippocampus is mediated by the MF pathway, the PP pathway, or both. Computationally, it implies that our model can not only retrieve two encodings for each memory but also perform heteroassociation between them. Autoassociation and heteroassociation are preserved over large ranges in model parameters (Supplementary Fig. 3F).

To show that concept target patterns $\mathbf{x}_{\mu}^{\mathrm{PP}}$ and average images within concepts are indeed valid representations of concepts for our image dataset, we plot them in image space after transforming $\mathbf{x}_{\mu}^{\mathrm{PP}}$ through the visualization pathway (Supplementary Fig. 2A). We observe that these two representation types appear similar to each other and lie near the centers of well-separated clusters of examples for each concept. In machine learning, clustering around central points is a common paradigm for unsupervised category learning, with k-means clustering as an example. In cognitive science as well, clustering has been used as model for unsupervised category learning[37,38], and central representations called prototypes can be used for category assignment[39]. Thus, we conclude that $\mathbf{x}_{\mu}^{\mathrm{PP}}$ and average images can indeed serve as concept representations. With more complex image datasets, such as CIFAR10[40], examples may not be clustered in image space or in encoding space with our model's simple autoencoder. To learn concepts, nonlinear decision boundaries can identified using supervised algorithms, but these complicated partitions of space may not admit central prototypes that accurately represent concepts. Alternatively, we can employ more sophisticated feature extraction techniques to map examples into an encoding space that exhibits clusters with simple boundaries between concepts. If that is achieved, then central features such as averages within clusters in that space can again serve as concept representations. More powerful feature extraction can be incorporated in our model by substituting our simple autoencoder with, for example, an unsupervised variational autoencoder or a supervised deep classifier.

We investigate retrieval more comprehensively by randomly generating MF and PP patterns across a broader range of statistics instead of propagating images along the hippocampal pathways in Fig. 2B (Methods). For simplicity, we take MF examples to be uncorrelated. In Fig. 3H, I, we show regimes for successful retrieval of MF examples, PP examples, and PP concepts. For MF and PP examples, the network has a capacity for stored patterns above which they can no longer be retrieved (Fig. 3H). For PP concepts, the network requires storage of a minimum number of examples below which concepts cannot be built. As expected intuitively, fewer examples are needed if they are more correlated, since common features can be more easily deduced. Figure 3I overlays retrieval regimes for MF examples and PP concepts. When the number of concepts is low, there exists a regime at intermediate numbers of stored examples in which both examples and concepts can be retrieved. This multiscale retrieval regime corresponds to the network behavior observed in Fig. 3D–G, and it is larger for more correlated PP

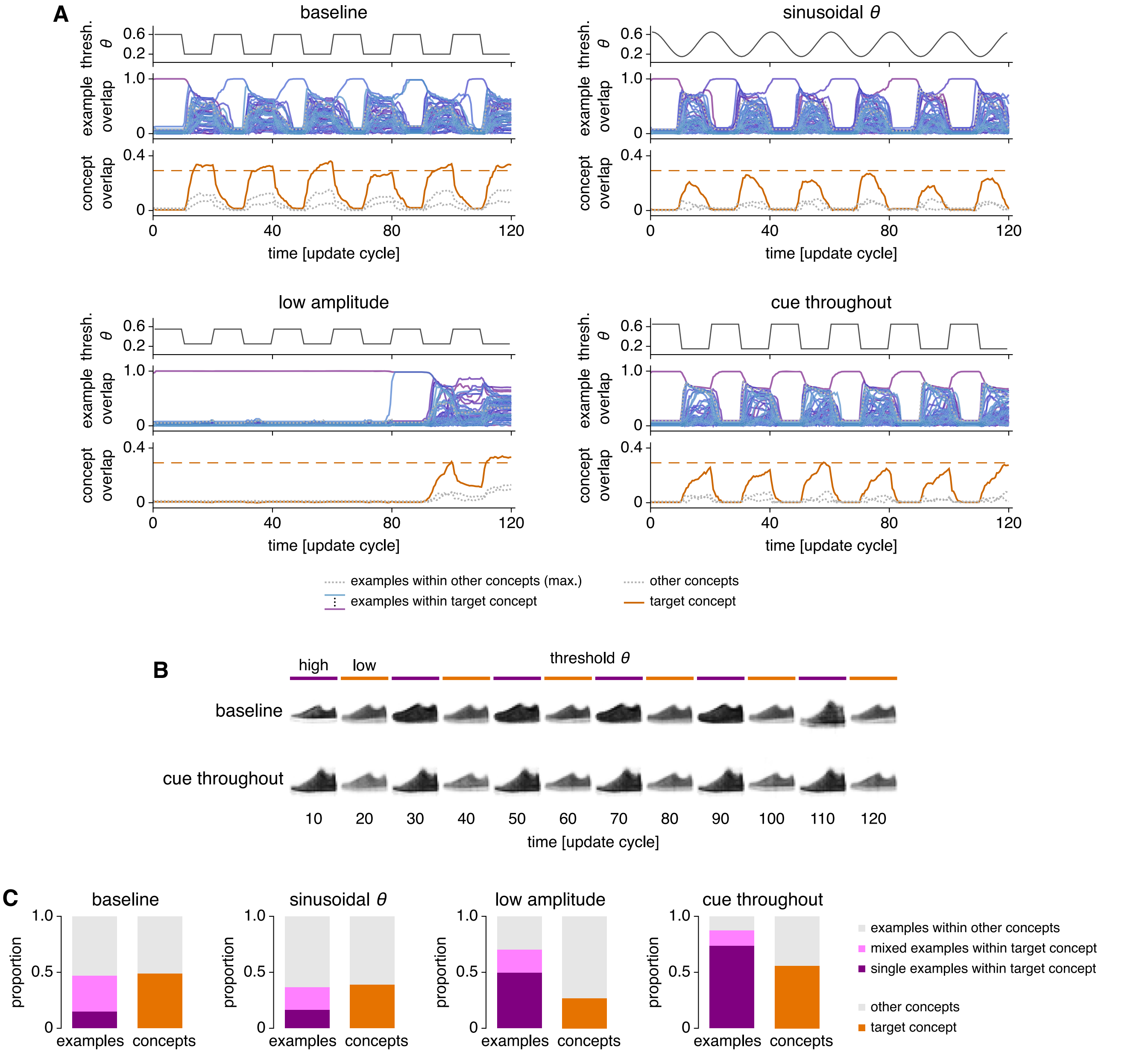


Fig. 4 | The CA3 model can alternate between MF example and PP concept representations under an oscillating threshold. Four scenarios are considered: a baseline condition with abrupt threshold changes, sinusoidal threshold changes, threshold values of 0.55 and 0.25 instead of 0.6 and 0.2, and the weak input of an MF cue throughout the simulation instead of only at the beginning. **A** Pattern overlap dynamics. Each panel shows, from top to bottom, the threshold, overlaps with MF examples, and overlaps with PP concepts. The dashed orange line indicates the theoretically estimated maximum value for concept retrieval (Methods). **B** Visualizations of retrieved patterns show alternation between examples and concepts. In the baseline case, various examples are explored; in the cue-throughout case, the same cued example persists. **C** Summary of retrieval behavior between update cycles 60 to 120. For each scenario, 20 cues are tested in each of 20 networks. Each panel depicts the fractions of simulations demonstrating various example (left) and concept (right) behaviors. In all networks, 50 randomly chosen examples from each of the 3 concepts depicted in Fig. 2A are stored. One update cycle corresponds to the updating of every neuron in the network (Methods). Source data are provided as a Source Data file.

encodings. On the other hand, its size does not substantially change with the sparsity of MF patterns (Supplementary Fig. 3G, H). Our capacity values agree with theoretical formulas calculated using techniques from statistical physics[41]. In all, our networks with randomly generated patterns demonstrate that our results generalize to larger networks that store more examples in more concepts and are not idiosyncratic to the pattern generation process in Fig. 2.

To further explore the heteroassociative capability of our network, we cue the network with an MF pattern and apply a time-varying threshold during retrieval. The network representation can then alternate between the PP concept of the original cue during oscillation phases with low threshold and various MF examples of that concept during phases with high threshold (Fig. 4A, B). Sharply and sinusoidally varying threshold values both produce this behavior. From one oscillation cycle to the next, the MF encoding can hop among different examples because concept information is preferentially preserved over example information during low-threshold phases. If we weakly apply the MF cue as additional neural input throughout the simulation (Methods), the network will only alternate between the target MF example and the target PP concept. This condition can represent memory retrieval with ongoing sensory input. If we decrease the amplitude of the oscillation,

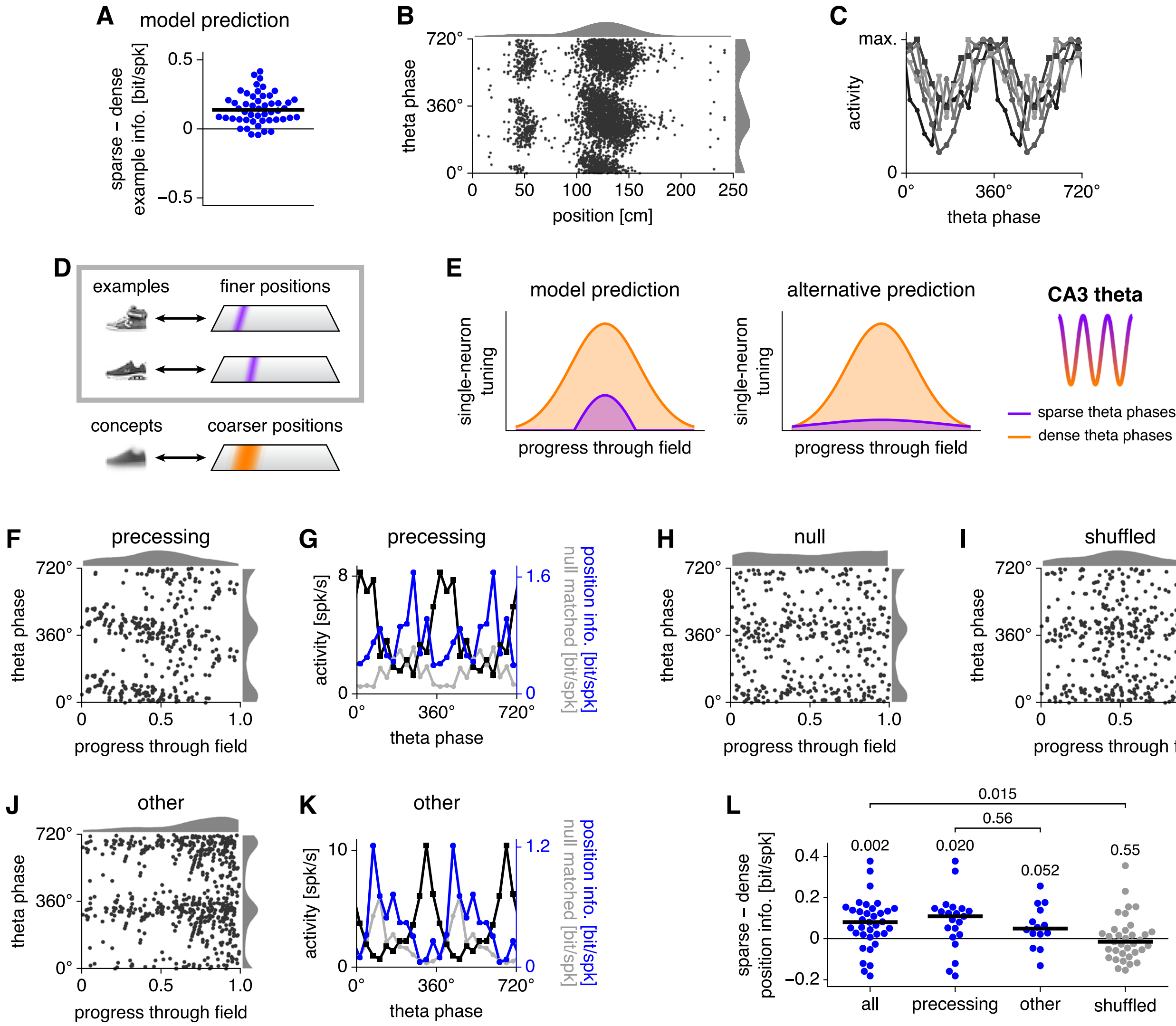


Fig. 5 | Place field data support the model prediction that sparser theta phases should preferentially encode finer, example-like positions. **A** Our CA3 model predicts that single neurons convey more information per spike about example identity during sparse regimes. Each point represents a neuron, $n = 50$. **B** Example CA3 place cell activity along a linear track. Each spike is represented by two points at equivalent phases with histograms over position (top) and over theta phase (right). **C** Activity by theta phase for 5 CA3 place cells. **D** To test our model, we construe CA3 place cells to store fine positions as examples, which can combine into coarser regions as concepts. Here, we focus on example encoding. **E** Our model predicts that CA3 place fields are more sharply tuned during sparse theta phases. An alternative hypothesis is sharper tuning during dense phases. **F** Example phase-precessing place field. **G** Activity (black), raw position information per spike (blue), and mean null-matched position information (gray) by theta phase for the field in **F**. Sparsity-corrected position information is the difference between the raw and mean null-matched values. **H** Null-matched place field obtained by replacing spike positions, but not phases, with uniformly distributed random values. **I** Shuffled place field obtained by permuting spike phases and positions. **J**, **K** Similar to **F**, **G**, but for a place field that is not precessing. **L** Average difference in position information between the sparsest and densest halves of theta phases. For all cell populations, sparse phases convey more position information per spike. Each point represents a field. All and shuffled $n = 35$, precessing $n = 21$, and other $n = 14$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests except for the comparison between precessing and other, which is calculated by the two-tailed Mann-Whitney U test. For all results, spikes during each traveling direction are separately analyzed. In **A** and **L**, information is sparsity-corrected with horizontal lines indicating medians. Source data are provided as a Source Data file.

alternation between examples and concepts is disrupted and the network favors one encoding type over the other. We quantify the distribution of network behaviors during high- and low-threshold phases in Fig. 4C. The proportion of simulations in which single MF patterns are retrieved, the persistence of the target PP concept, and other retrieval properties vary with network parameters. In Supplementary Fig. 4A, B, we present analogous results for randomly generated MF and PP patterns demonstrating that these retrieval properties also depend on MF pattern sparsity. All in all, while our network can represent either examples or concepts at each moment in time, an oscillating threshold provides access to a range of representations over every oscillation cycle.

Place cell data reveals predicted relationships between encoding properties and theta phase

The central feature of our CA3 model is that an activity threshold determines whether the network retrieves example or concept encodings. We claim that the theta oscillation in CA3 physiologically implements this threshold and drives changes in memory scale. To be specific, our model predicts that single neurons should convey more information per spike about example identity during epochs of sparser activity (Fig. 5A). This single-neuron prediction can be tested by analyzing publicly available datasets of CA3 place cells. Figure 5B shows one example place cell recorded while a rat traverses a linear track[24,42]. During locomotion, single-neuron activity in CA3 is strongly modulated by the theta oscillation (Fig. 5C); we use this activity as an indicator of network sparsity since a relationship between the two has been observed (Fig. 3 in Skaggs et al.[43]). We assume an equivalence between the encoding of images by our CA3 model and the encoding of spatial positions by CA3 place cells (Fig. 5D). Examples are equivalent to fine positions along the linear track. Just as similar examples merge into concepts, nearby positions can aggregate into coarser regions of space. Through this equivalence, we can translate the prediction about example information per spike (Fig. 5A) into a prediction about spatial tuning (Fig. 5E). During denser theta phases, place fields should be broader, which corresponds to lower position information per spike. This prediction relies on our claim that the theta oscillation in CA3 acts as the inhibitory threshold of our model. A priori, the alternative prediction that place fields are sharper during dense theta phases is an equally valid hypothesis. Higher activity may result from strong drive by external stimuli that the neuron serves to encode, while lower activity may reflect noise unrelated to neural tuning. The sharpening of visual tuning curves by attention is an example of this alternative prediction[44]. From a more general perspective, the model and alternative predictions in Fig. 5E roughly correspond to subtractive and divisive modulation of firing rates, respectively. Both kinds of inhibitory effects are found in cortical circuits[45–47]. We will now test whether experimental data reflect our model prediction of sharper place field tuning with higher spatial information during sparser theta phases, which would support a subtractive role of theta as an oscillating inhibitory threshold over a divisive one.

First, we investigate the encoding of fine, example-like positions by analyzing phase-dependent tuning within single place fields. We use the Collaborative Research in Computational Neuroscience (CRCNS) hc-3 dataset contributed by György Buzsáki and colleagues[24,42]. Figure 5F shows one extracted field that exhibits phase precession (for others, see Supplementary Fig. 5A). At each phase, we compute the total activity as well as the information per spike conveyed about position within the field (Fig. 5G and Supplementary Fig. 5B). It is well known that the estimation of information per spike is strongly biased by sparsity. Consider the null data in Fig. 5H that is matched in spike phases; spike positions, however, are randomly chosen from a uniform distribution. In the large spike count limit, uniformly distributed activity should not convey any information. Yet, the null data show more position information per spike during sparser phases (Fig. 5G). To correct for this bias, we follow previous protocols and subtract averages over many null-matched samples from position information[48]. In all of our comparisons of information between sparse and dense phases, including the model prediction in Fig. 5A, we report sparsity-corrected information. For further validation, we generate a shuffled dataset that disrupts any relationship between spike positions and phases found in the original data (Fig. 5I). Figure 5J, K illustrates a second place field whose tuning also depends on theta phase but does not exhibit precession. For each theta-modulated CA3 place field, we partition phases into sparse and dense halves based on activity, and we average the sparsity-corrected position information per spike across each partition. CA3 place fields convey significantly more information during sparse phases than dense phases (Fig. 5L). This relationship is present in both phase-precessing and other fields (although slightly non-significantly in the latter) and is absent in the shuffled data. Thus, experimental data support our model's prediction that CA3 encodes information in a finer, example-like manner during sparse theta phases. Notably, CA1 place fields do not convey more information per spike during sparse phases, which helps to show that our prediction is nontrivial and demonstrates that the phase behavior in CA3 is not just simply propagated forward to CA1 (Supplementary Fig. 5C).

To characterize the relationship between information and theta phase more precisely, we aggregate spikes over phase-precessing fields in CA3 and in CA1 (Supplementary Fig. 5D–G). This process implicitly assumes that each phase-precessing field is a sample of a general distribution characteristic to each region. These aggregate fields recapitulate the single-neuron results that CA3 spikes are uniquely more informative during sparse phases (Supplementary Fig. 5H). They also reveal how position information varies with other field properties over theta phases (Supplementary Fig. 5I, J). For example, information is negatively correlated with field width, confirming the interpretation that more informative phases have sharper tuning curves (Fig. 5E). In CA3, information is greatest during early progression through the field, which corresponds to future locations, with a smaller peak during late progression, which corresponds to past locations. In contrast, past locations are more sharply tuned in CA1. Thus, different hippocampal subfields may differentially encode past and future positions across the theta cycle; we will return to this topic in the Discussion.

Next, we turn our attention to the representation of concepts instead of examples. Our model predicts that single neurons exhibit more concept information per spike during dense activity regimes (Fig. 6A). To test this prediction using the same CRCNS hc-3 dataset, we invoke the aforementioned equivalence between concepts in our model and coarser positions along a linear track (Fig. 6B). Thus, single CA3 neurons should encode more information per spike about coarse positions during dense theta phases. Previously, to test for finer position encoding in Fig. 5, we divided single place fields into multiple position bins during the computation of information. Here, we analyze encoding of coarser positions by choosing large position bins across the whole track (Fig. 6C, D). We consider different bin sizes to characterize at which scale the merging of examples into concepts occurs. When we again compute the average difference in sparsity-corrected position information per spike between sparse and dense theta phases, we find that dense phases are the most preferentially informative at the coarsest scales (Fig. 6E). CA1 place cells also exhibit this property (Supplementary Fig. 6B). Crucially, differences between sparse and dense phases are not seen in shuffled data, which supports the validity of our analysis methods (Fig. 6F). Our results are further bolstered by their preservation under a different binning procedure (Supplementary Fig. 6C–E). Thus, coarse positions along a linear track can be best distinguished during dense theta phases, in agreement with our model. Note that we always consider 4 bins at a time even for track scales smaller than 1/4, because changing the number of bins across scales introduces a bias in the shuffled data (Supplementary Fig. 6F–H). At finer scales, this process sometimes fails to capture entire fields and artificially splits them (Fig. 6C, left). Here, we adopt a neutral approach and do not adjust our partitions to avoid these cases; the opposite approach was adopted in Fig. 5 where intact fields were explicitly extracted. These different approaches may explain why at the finest scales in Fig. 6E, sparse phases do not convey more information like they do in Fig. 5L.

In our model, concepts are formed by merging examples across all correlated features. While track position can be one such feature, we now assess whether our predictions also apply to another one. In the CRCNS hc-6 dataset contributed by Loren Frank and colleagues, CA3 place cells are recorded during a W-maze alternation task in which mice must alternately visit left and right arms between runs along the center arm[25]. It is known that place cells along the center arm can encode the turn direction upon entering or leaving the center arm in addition to position[49,50]. Again, our model predicts that sparse theta phases preferentially encode specific information (Fig. 7A), so they should be more tuned to a particular turn direction (Fig. 7B). During dense phases, they should generalize over turn directions and solely encode position.

Figure 7C shows spikes from one CA3 place cell accumulated over outward runs along the center arm followed by either left or right turns (Supplementary Fig. 7A). For each theta phase, we compute the total

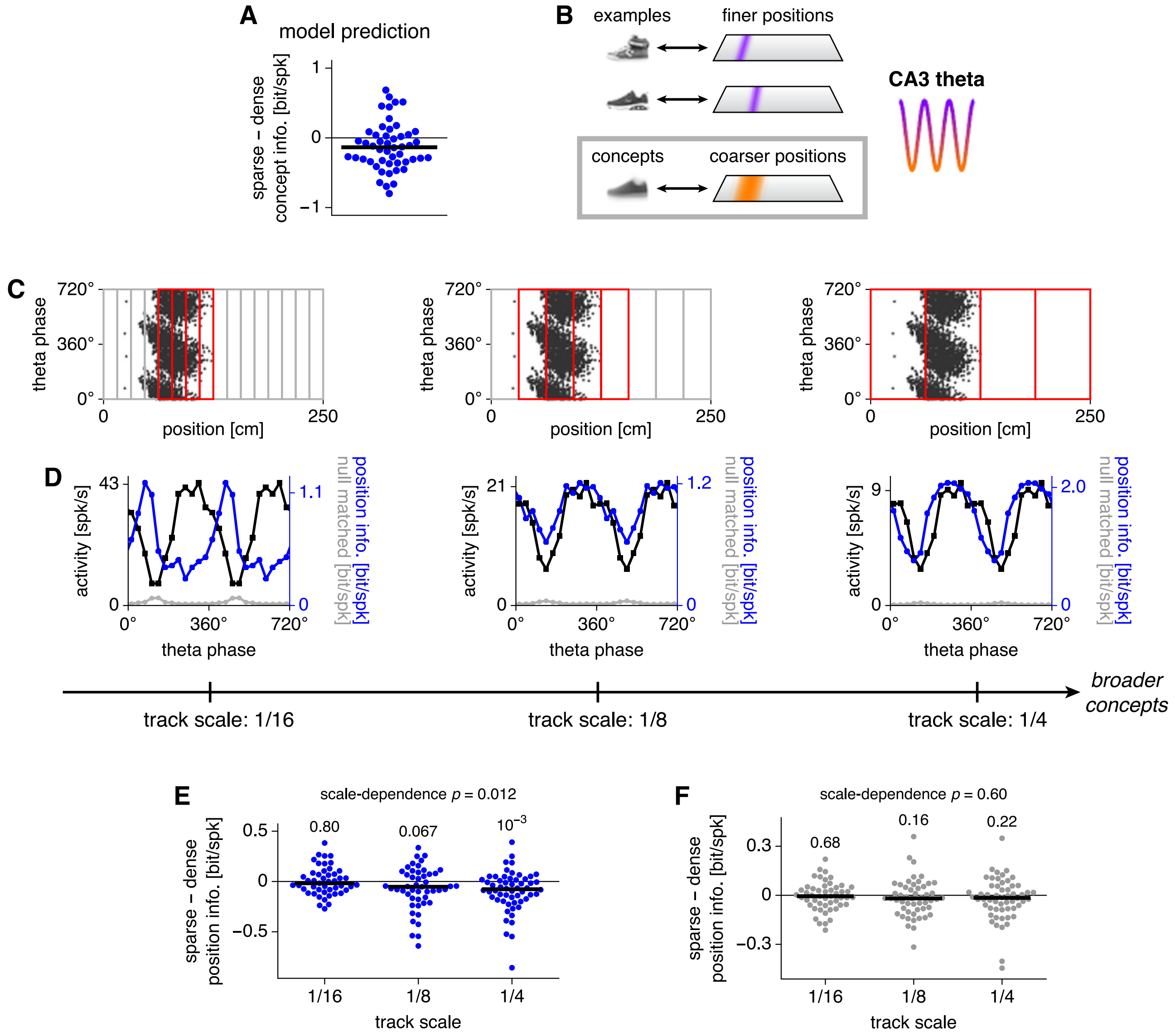


Fig. 6 | Place cell data support the model prediction that denser theta phases should preferentially encode coarser, concept-like positions. **A** Our CA3 model predicts that single neurons convey more information per spike about concept identity during dense regimes. Each point represents a neuron, $n = 50$. **B** To test our model, we construe CA3 place cells to store fine positions as examples, which can combine into coarser regions as concepts. Here, we focus on concept encoding. **C** We calculate position information at various track scales over windows of 4 contiguous bins. **D** Activity (black), raw position information per spike (blue), and mean null-matched position information (gray) by theta phase for the red windows in **C.** Sparsity-corrected position information is the difference between the raw and mean null-matched values. **E** Average difference in position information between the sparsest and densest halves of theta phases. For coarser scales, dense phases convey more position information per spike. Each point represents values from a place cell averaged over all windows. Track scale 1/16 $n = 47$, 1/8 $n = 49$, and 1/4 $n = 56$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests for each scale and by Spearman's ρ for the trend across scales. **F** Similar to **E**, but for shuffled data whose spike phases and positions are permuted. For all results, spikes during each traveling direction are separately analyzed. In **A**, **E**, and **F**, information is sparsity-corrected with horizontal lines indicating medians. Source data are provided as a Source Data file.

activity, the turn information per spike (ignoring position), and the mean information of null-matched samples used for sparsity correction (Fig. 7D). Figure 7E, F shows similar results for inward runs (for others, see Supplementary Fig. 7B, C). For both outward and inward runs, sparsity-corrected turn information per spike is greater during sparse theta phases compared to dense phases (Fig. 7G). This finding is not observed in data in which theta phase and turn direction are shuffled (Fig. 7G, H). Not only do these results support our model, they also reveal that in addition to splitter cells that encode turn direction over all theta phases[51], CA3 contains many more place cells that encode it only at certain phases (Supplementary Fig. 7D). The difference between sparse and dense phases is significantly greater in CA3 than it is in CA1 (Supplementary Fig. 7E, F). Thus, our subfield-specific results for example encoding are consistent across position and turn direction. Aggregate neurons, formed by combining spikes from more active turn directions and those from less active turn directions, demonstrate similar tuning properties to individual neurons (Supplementary Fig. 7G–I).

Beyond the single-neuron results presented above, we seek to test our predictions at the population level. To do so, we perform phase-dependent Bayesian population decoding of turn direction during runs along the center arm (Fig. 7I). This analysis requires multiple neurons with sufficiently sharp tuning to be simultaneously active across all theta phases; it can be used to decode left versus right turns, whereas an analogous decoding of track position, which spans a much broader range of values, is intractable with our datasets. We find that

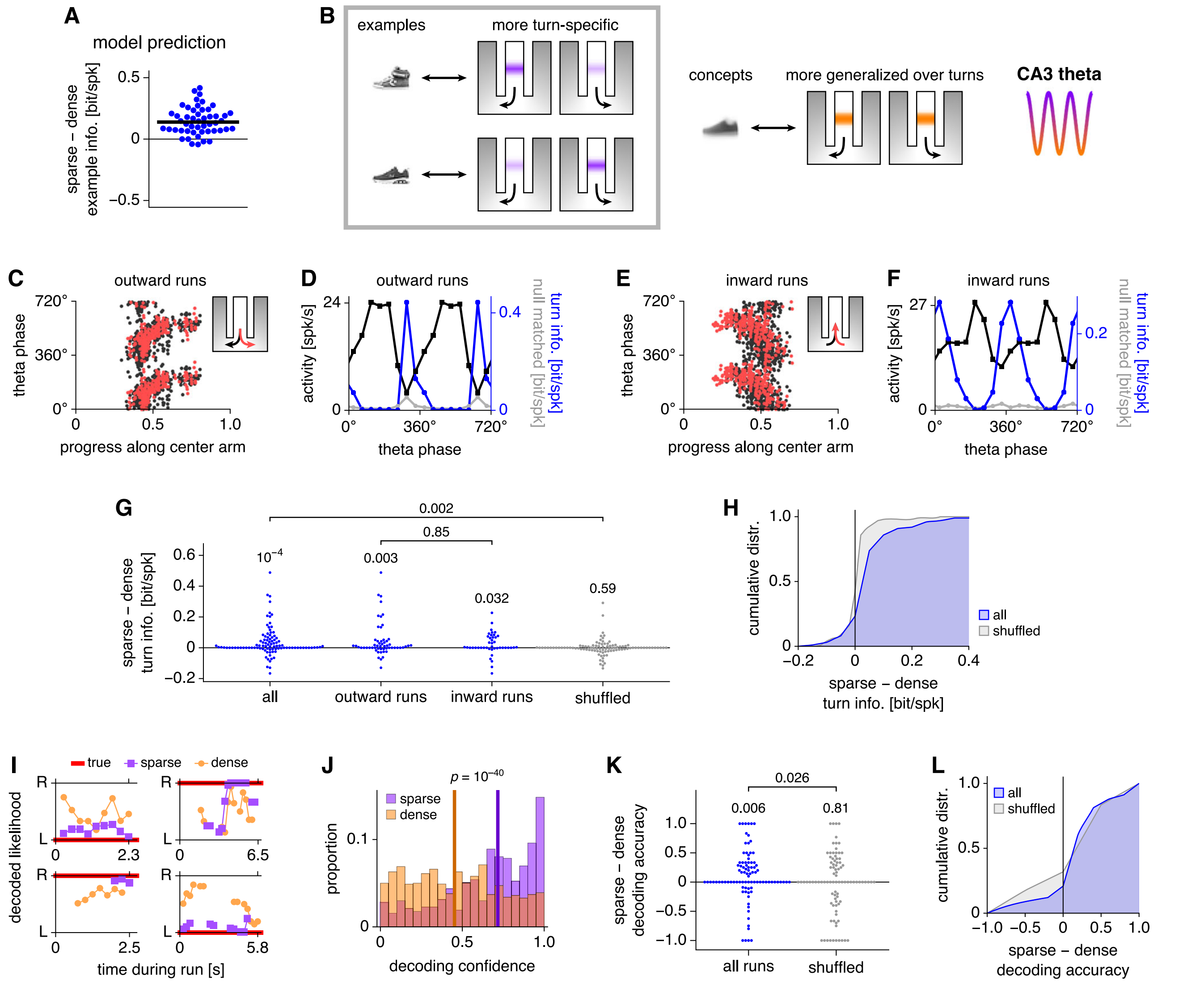


Fig. 7 | W-maze data support the model prediction that sparser theta phases should preferentially encode turn direction in addition to position. **A** Same as Fig. 5A. **B** To test our model, we construe CA3 place cells to store turn directions during the central arm of a W-maze alternation task as examples. By combining examples, concepts that generalize over turns to solely encode position can be formed. **C**–**H** Single-neuron information results. **C** Example place cell that is active during outward runs. Each spike is represented by two points at equivalent phases with different colors representing different future turn directions. **D** Activity (black), raw turn information (blue), and mean null-matched turn information (gray) by theta phase for the neuron in **C**. Sparsity-corrected turn information is the difference between the raw and mean null-matched values. **E**, **F** Similar to **C**, **D**, but for a place cell that is active during inward runs with colors representing past turn directions. **G** Average difference in turn information between the sparsest and densest halves of theta phases. For all cell populations, sparse phases convey more turn information per spike. Each point represents a place cell. All and shuffled $n = 99$, outward runs $n = 56$, and inward runs $n = 43$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests except for the comparison between outward and inward runs, which is calculated by the two-tailed Mann-Whitney U test. **H** Cumulative distribution functions for values in **G**. **I**–**L** Bayesian population decoding results. **I** Likelihood of left (L) or right (R) turns during four runs along the center arm using spikes from either the sparsest or densest halves of theta phases. **J** Sparse encodings exhibit greater confidence about turn direction. Vertical lines indicate medians with p-value calculated by the two-tailed Mann-Whitney U test. **K** Average difference in maximum likelihood estimation accuracy between the sparsest and densest halves of theta phases. Sparse phases encode turn direction more accurately. Each point represents one run averaged over decoded timepoints. All runs and shuffled $n = 91$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests. **L** Cumulative distribution functions for values in **K**. For all results, spikes during each traveling direction are separately analyzed. In **A**, **G**, and **H**, information is sparsity-corrected. Source data are provided as a Source Data file.

the CA3 population likelihood exhibits greater confidence during sparse phases (Fig. 7J). From a Bayesian perspective, the population expresses stronger beliefs about turn direction during sparse phases and is more agnostic during dense phases. If pressed to choose the direction with a higher likelihood as its estimate, CA3 is also more accurate during sparse phases (Fig. 7K, L). These results match our predictions in Fig. 7A, B and bolster our single-neuron results. Moreover, they are specific to CA3, as similar conclusions cannot be made about the CA1 place cell population (Supplementary Fig. 7J–L).

In summary, extensive data analysis reveals experimental support for our CA3 model over two datasets collected by different research groups, across two encoding modalities, for both example and concept representations, and at both the single-neuron and population level.

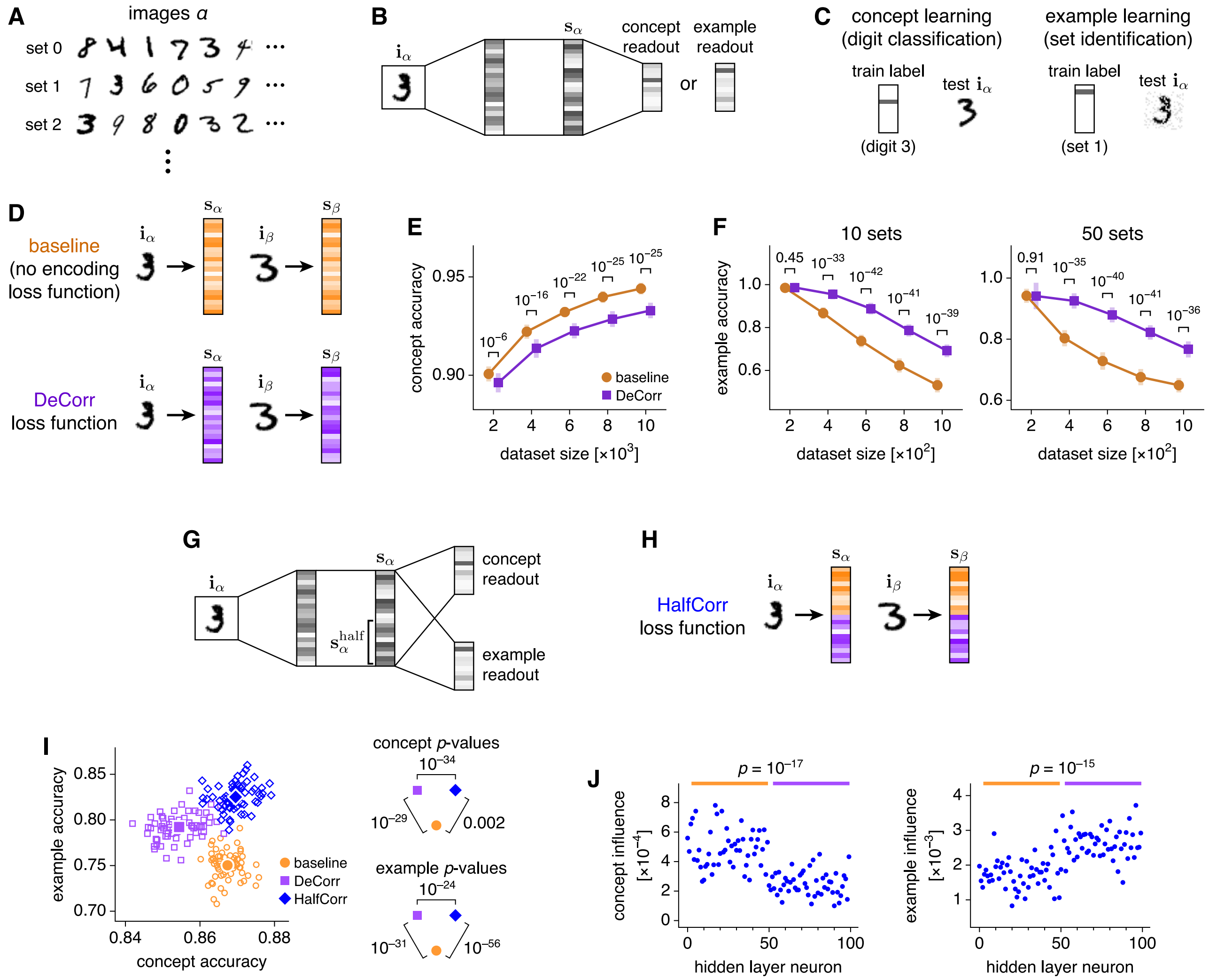


Fig. 8 | Complementary encodings inspired by CA3 can improve machine learning performance in a complex task. A We extend the MNIST dataset by randomly assigning an additional set label to each image. **B**–**F** We train a multilayer perceptron to either classify digits or identify sets. **B** Network architecture. Each hidden layer contains 50 neurons. **C** Task structures. Digit classification requires building concepts and is tested with held-out test images. Set identification requires distinguishing examples and is tested with noisy train images. **D** We apply the DeCorr loss function (Eq. (3)) to decorrelate encodings in the final hidden layer, in analogy with MF patterns in CA3. Without an encoding loss function, image correlations are preserved, in analogy with PP patterns. **E**, **F** DeCorr decreases concept learning performance and increases example learning performance. Points indicate means and bars indicate standard deviations over 32 networks. **G**–**J** We train a multilayer perceptron to simultaneously classify digits and identify sets. **G** Network architecture. Each hidden layer contains 100 neurons. The train dataset contains 1000 images and 10 sets. **H** We apply the HalfCorr loss function (Eq. (4)) to decorrelate encodings only among the second half of the final hidden layer. Correlated and decorrelated encodings are both present, in analogy with MF and PP patterns across the theta cycle in CA3. **I** DeCorr networks generally perform better at example learning but worse at concept learning compared to baseline. HalfCorr networks exhibit high performance in both tasks. Open symbols represent individual networks and filled symbols represent means over 64 networks. **J** Influence of each neuron in HalfCorr networks on concept and example learning, defined as the average decrease in accuracy upon clamping its activation to 0. Correlated neurons (orange bars) are more influential in concept learning, and decorrelated neurons (purple bars) are more influential in example learning. For all results, *p*-values are computed using unpaired two-tailed *t*-tests. Source data are provided as a Source Data file.

CA3-like complementary encodings improve neural network performance in multitask machine learning

We have observed how CA3 encodes behaviorally relevant information at different scales across theta phases. Can these different types of encodings be useful for solving different types of tasks? Can they even benefit neural networks designed for machine learning, abstracting away from the hippocampus? To address these questions, we turn to a classic paradigm in machine learning: a multilayer perceptron trained on MNIST handwritten digit images[52]. First, we augment the MNIST dataset by randomly assigning an additional label to each image: a set number (Fig. 8A). We train the fully connected feedforward network to perform one of two tasks: classification of the written digit or identification of the assigned set (Fig. 8B). The former requires clustering of images based on common features, which resembles concept learning in our CA3 model, and the latter requires discerning differences between similar images, which resembles example learning in our CA3 model (Fig. 8C). We use a held-out test dataset to evaluate digit classification performance and corrupted images from the train dataset to evaluate set identification performance.

In our CA3 model, we found that examples were preferentially encoded by the decorrelated MF pathway and concepts by the correlated PP pathway (Fig. 3). In an analogous fashion, we seek to manipulate the correlation properties within the final hidden layer of our perception, whose activations $\mathbf{s}_\alpha$ serve as encodings of the input images $\mathbf{i}_\alpha$. In particular, we apply a DeCorr loss function, which penalizes correlations in $\mathbf{s}_\alpha$ between every pair of items α, β in a training batch (Fig. 8D):

$$\mathcal{L}_{\mathrm{DeCorr}} \approx \frac{1}{2}\sum_{\substack{\alpha,\beta\in \\ \mathrm{batch}}} \mathrm{Pearson}(\mathbf{s}_\alpha,\mathbf{s}_\beta)^2. \tag{3}$$

DeCorr mimics the MF pathway; the equation is approximate due to a slight modification of the Pearson correlation formula to aid numerical convergence (Methods). Alternatively, we consider the baseline condition with no loss function on hidden layer activations, which preserves natural correlations between similar images and mimics the PP pathway. Indeed, we observe that different encoding properties are suited for different tasks. Baseline networks perform better in concept learning (Fig. 8E) while DeCorr networks perform better in example learning (Fig. 8F), and these effects vary consistently with the strength of the DeCorr loss function (Supplementary Fig. 8A, B). Thus, DeCorr allows us to tune encoding correlations in neural networks to highlight input features at either broader or finer scales. Tasks can be solved more effectively by matching their computational requirements with the appropriate encoding scale. Note that DeCorr is different from the DeCov loss function previously developed to reduce overfitting[53]. DeCorr decorrelates pairs of inputs across all neurons in the specified layer, whereas DeCov decorrelates pairs of neurons across all inputs. As a regularizer that promotes generalization, DeCov improves digit classification and does not substantially improve set identification, which contrasts with the effect of DeCorr (Supplementary Fig. 8C, D).

Complex tasks, including those performed by biological systems, may require information to be processed at different scales of correlation. In CA3, a spectrum of encodings is available during each theta cycle. Can neural networks take advantage of multiple encodings? We tackle this question by asking a perceptron to simultaneously perform digit classification and set identification (Fig. 8G). In addition to the baseline and DeCorr networks, we define a HalfCorr loss function (Fig. 8H):

$$\mathcal{L}_{\mathrm{HalfCorr}} \approx \frac{1}{2}\sum_{\substack{\alpha,\beta\in \\ \mathrm{batch}}} \mathrm{Pearson}(\mathbf{s}_\alpha^{\mathrm{half}},\mathbf{s}_\beta^{\mathrm{half}})^2, \tag{4}$$

where $\mathbf{s}_\alpha^{\mathrm{half}}$ represents the second half of neurons in the final hidden layer. After training with this loss function, the neural representation consists of both a correlated, PP-like component in the first half and a decorrelated, MF-like component in the second half. When we evaluate these networks on both digit classification and set identification, we see that baseline and DeCorr networks behave similarly to how they did on single tasks. Compared to baseline, DeCorr networks perform better in example learning at the cost of poorer concept learning (Fig. 8I). However, HalfCorr networks do not suffer from this tradeoff and perform well at both tasks. Their superior performance is maintained over a variety of network and dataset parameters (Supplementary Fig. 8E). Moreover, HalfCorr networks learn to preferentially use each type of encoding for the task to which it is better suited. We use the decrease in task accuracy upon silencing a neuron as a metric for its influence on the task. Correlated neurons are more influential in concept learning and decorrelated neurons in example learning (Fig. 8J).

Note that we do not manipulate pattern sparsity in these artificial networks. Sparsification can be useful in the hippocampus because it provides a biologically tractable means of achieving decorrelation. It also allows biological networks to access both less and more correlated representations by changing the level of inhibition. Instead, we can directly manipulate correlation through the DeCorr and HalfCorr loss functions. Under some conditions, the decorrelated half of the final hidden layer in HalfCorr networks indeed exhibits sparser activation than the correlated half (Supplementary Fig. 8F). It is possible that directly diversifying sparsity can also improve machine learning performance, especially since sparse coding is known to offer certain computational advantages as well as greater energy efficiency[27,54,55].

Discussion

The hippocampus is widely known to produce our ability to recall specific vignettes as episodic memories. This process has been described as indexing every sensory experience with a unique neural barcode so that separate memories can be independently recovered[56,57]. Recently, research has shown that the hippocampus is also important in perceiving commonalities and regularities across individual experiences, which contribute to cognitive functions such as statistical learning[58,59], category learning[60–63], and semantic memory[64–66]. Evidence for this has been obtained largely through human studies, which can present and probe memories in controlled settings. However, the detailed circuit mechanisms used by the hippocampus to generalize across experiences while also indexing them separately are not known.

Our analysis of rodent place cell recordings reveals that single CA3 neurons alternate between finer, example-like representations and broader, concept-like representations of space across the theta cycle (Figs. 5–7). These single-neuron results extend to the network level, which alternatively encodes more specific and more general spatial features in a corresponding manner (Fig. 7). If we accept that place cells store these features as spatial memories, then our experimental analysis reveals that CA3 can access memories of different scales at different theta phases. We propose that the computational mechanism underlying these observations is the multiplexed encoding of each memory at different levels of correlation (Figs. 2 and 8). We show that CA3 can biologically implement this mechanism through the storage of both sparse, decorrelated MF inputs and dense, correlated PP inputs and their alternating retrieval by the theta oscillation, which acts as an activity threshold and subtractively modulates neural activity (Fig. 2, 3, and 4). Our model performs successful pattern completion of both types of encodings, suggesting that patterns across the theta cycle can truly function as memories that can be recovered from partial cues.

Alone, our secondary analyses of experimental data contribute to a large set of observations on how coding properties vary with theta phase in the hippocampus. Of note is phase precession, in which different phases preferentially encode different segments within a firing field as it is traversed, with later phases tuned to earlier segments[67]. Phase precession is most widely reported for place cells and traversals of physical space, but it also appears during the experience of other sequences, such as images and tasks[68–70]. Our analysis implies that the sharpness of tuning is not constant throughout traversals. In particular, CA3 neurons are more sharply tuned at early positions in place fields, while CA1 neurons are more sharply tuned at late positions (Supplementary Fig. 5I, J). Transforming these conclusions about position into those about time through the concept of theta sequences, CA3 represents the future more precisely, while CA1 represents the future more broadly. The latter is consistent with the idea that CA1 may participate in the exploration of multiple possible future scenarios[71]. Furthermore, our W-maze analysis reveals that certain hippocampal neurons which do not obviously encode an external modality across all theta phases, such as turn direction, may do so only during sparse phases (Supplementary Fig. 7D). This observation adds to the subtleties with which the hippocampus represents the external world.

Other groups have investigated the variation of place field sharpness with theta phase in CA1, not CA3, and their results are largely in agreement with our CA1 analyses. Skaggs et al.[43] partitioned theta phases into halves, one of which with higher activity than the other, and found more information per spike during the less active half. We observe no difference at the single-neuron level, though our W-maze results are only slightly non-significant (Supplementary Figs. 5C and 7E). Their partitions differ from ours by 30° and they employ a different binning technique, both of which can influence the results. The less informative phases in their work correspond to future positions, which we also observe (Supplementary Fig. 5J). Note that their computation of sparsity is performed along a different axis compared to ours; using terms from Willmore and Tolhurst[72], they use the *lifetime* density while we compute the *population* density, which fundamentally differ. Ujfalussy and Orbán[73] also found that phases with larger field sizes correspond to future positions. Mehta et al.[74] considered phase-dependent tuning within CA1 place fields, but they calculate field width over theta phase as a function of field progress, whereas we do the opposite. Similarly, Souza and Tort[75] considered phase tuning at various field progresses. Both sets of results appear to be compatible with ours, but a direct comparison cannot be made. Overall, our work offers original insights into hippocampal phase coding not only by focusing on CA3, which behaves differently from CA1, but also by elucidating a connection between tuning width and network sparsity. Intriguingly, Pfeiffer and Foster[76] found a relationship between CA1 replay speed and slow gamma oscillation phase, which modulates network activity during quiescence. This observation opens the possibility that other oscillations can leverage the connection between tuning and sparsity when the dominant theta rhythm is absent.

Better memory performance has been associated with greater theta power during both encoding[77–82] and retrieval[79,82–84]. Our model cannot explain the former because it does not contain a theta oscillation during encoding. It does, however, offer a possible explanation for the latter observation. We simulated retrieval under an oscillating threshold with lower amplitude and observed that the network stalls on MF example encodings instead of alternating with PP concept encodings (Fig. 4A). Thus, the biological processes that produce larger theta amplitudes, such as stronger medial septum inputs or changes in neuromodulator concentrations[85], may promote memory recall by granting access to wider ranges of pattern sparsities and, consequently, representational scales.

The temporal coordination between memory storage and retrieval is also biologically significant. We make the major simplification of separately simulating memory storage and retrieval. These two operating regimes can represent different tones of a neuromodulator such as acetylcholine, which is thought to bias the network towards storage[86]. Another proposal is that storage and retrieval preferentially occur at different theta phases, motivated by the variation in long-term potentiation (LTP) strength at CA1 synapses across the theta cycle[87–89]. Although this idea focuses on plasticity in CA1, it is possible that storage and retrieval also occur at different phases in CA3. Note that our experimental analysis reveals a sharp dip in position information around the sparsest theta phase in both CA3 and CA1 (Supplementary Fig. 5E, G). This phase may coincide with the storage of new inputs, during which the representation of existing memories is momentarily disrupted; the rest of the theta cycle may correspond to retrieval. This interpretation could motivate excluding the sparsest theta phase from further analysis, since our model predictions only regard memory retrieval. However, we take a conservative approach and include all phases. Interestingly, recent work reported that the strength of LTP in CA1 peaks twice per theta cycle[90], suggesting for our model that MF and PP patterns could have their own storage and retrieval intervals during each theta cycle.

Our work connects hippocampal anatomy and physiology with foundational attractor theory. Among others, Tsodyks and Feigel'man[91] observed that sparse, decorrelated patterns can be stored at high capacity, and Fontanari[92] found that dense, correlated patterns can merge into representations of common features. We demonstrate that both types of representations can be stored and retrieved in the same network, using a threshold to select between them. This capability can be given solid theoretical underpinnings using techniques from statistical mechanics[41]. The convergence of MF and PP pathways in CA3 has also been the subject of previous computational investigations[15–17]. In these models, CA3 stores and retrieves one encoding per memory, while our model asserts that multiple encodings for the same memory alternate across the theta cycle. Another series of models proposes, like we do, that the hippocampus can simultaneously maintain both decorrelated, example-like encodings and correlated, concept-like encodings[93,94]. These encodings converge at CA1 and each type is not independently retrieved there, which differs from our model. In a related hippocampal model, the PP pathway was shown to be crucial for learning cue-target associations in the presence of additional context inputs to EC that drift over time[95]. Successful learning requires the network to perform decontextualization and abstract away the slowly varying context inputs, which is, like concept learning, a form of generalization. EC has also been hypothesized to differentially encode inputs upstream of CA3, with specific sensory information conveyed by lateral EC and common structural representations by medial EC[96]. Further experimental investigation into the contributions of various subregions would help to clarify how the hippocampus participates in memory generalization.

Finally, our work addresses how CA3-like complementary encodings can be computationally leveraged by neural networks to solve complex tasks. We conceptually extend our results about CA3 and introduce a HalfCorr loss function that diversifies hidden layer representations to include both correlated and decorrelated components (Fig. 8). HalfCorr networks can better learn tasks that involve both distinction between similar inputs and generalization across them. They are simultaneously capable of pattern separation and categorization even based on small datasets, demonstrating a possible advantage of brain computation over conventional deep learning. Yet, we deliberately chose a neural architecture that differs from that of the CA3 network to test the scope over which complementary encodings can improve learning. Instead of a recurrent neural network storing patterns of different sparsities through unsupervised Hopfield learning rules, we implemented a feedforward multilayer perceptron, a workhorse of supervised machine learning. The success of HalfCorr networks in this scenario supports the possibility that HalfCorr can be broadly applied as a plug-and-play loss function to improve computational flexibility.

Functional heterogeneity is commonly invoked in the design of modern neural networks. It can be implemented in the form of deep or modular neural networks in which different subnetworks perform different computations in series or parallel, respectively[97,98]. Of note, Kowadlo et al.[99] constructed a deep, modular network inspired by the hippocampus for one-shot machine learning of both concepts and examples. As an aside, their architecture shares similarities with our hippocampus model in Fig. 2, but their Hopfield-like network for CA3 only stores MF patterns and inactivating these recurrent connections does not affect network performance. In contrast to these networks with specialized subnetworks, we propose a different paradigm in which a loss function applied differentially across neurons promotes heterogeneity within a single layer. This idea can be extended from the two components of HalfCorr networks, correlated and decorrelated, by assigning a different decorrelation strength to each neuron and thereby producing a true spectrum of representations. Furthermore, heterogeneity in other encoding properties such as mean activation, variance, and sparsity may also improve performance in tasks with varying or unclear computational requirements. Such tasks are not limited to multitask learning, but also include continual learning[100], reinforcement learning[101], and natural learning by biological brains.

Methods

Transformation of memories along hippocampal pathways

Binary autoencoder from images to EC. Our memories are 256 images from each of the *sneaker*, *trouser*, and *coat* classes in the FashionMNIST dataset[26]. We train a fully connected linear autoencoder on these images with three hidden layers of sizes 128, 1024, and 128. Batch normalization is applied to each hidden layer, followed by a rectified linear unit (ReLU) nonlinearity for the first and third hidden layers and a sigmoid nonlinearity for the output layer. Activations in the middle hidden layer are binarized by a Heaviside step function with gradients backpropagated by the straight-through estimator[102]. The loss function is

$$\mathcal{L} = \sum_{\substack{\mu\nu\in\\ \text{batch}}} ||\hat{\mathbf{i}}_{\mu\nu} - \mathbf{i}_{\mu\nu}||^2 + \lambda \sum_{\substack{\mu\nu\in\\ \text{batch}}} \mathrm{KL}\left(\frac{1}{N_{\mathrm{EC}}}\sum_i x^{\mathrm{EC}}_{\mu\nu i} \,\Big|\Big|\, a_{\mathrm{EC}}\right), \tag{5}$$

where $\mathbf{i}_{\mu\nu}$ is the image with pixel values between 0 and 1, $\hat{\mathbf{i}}_{\mu\nu}$ is its reconstruction, $\mathbf{x}^{\mathrm{EC}}_{\mu\nu}$ represents the binary activations of the middle hidden layer with $N_{\mathrm{EC}} = 1024$ units indexed by i, and $a_{\mathrm{EC}} = 0.1$ is its desired density (Fig. 2C). Sparsification with strength $\lambda = 10$ is achieved by computing the Kullback-Leibler (KL) divergence between the hidden layer density and a_{EC}[103]. Training is performed over 150 epochs with batch size 64 using the Adam optimizer with learning rate 10^{-3} and weight decay 10^{-5}.

Binary feedforward networks from EC to CA3. To propagate patterns from EC to DG, from DG to MF inputs, and from EC to PP inputs, we compute

$$x^{\mathrm{post}}_{\mu\nu i} = \Theta\left[\sum_j W_{ij} x^{\mathrm{pre}}_{\mu\nu j} - \theta\right], \tag{6}$$

where $\mathbf{x}^{\mathrm{pre}}_{\mu\nu}$ and $\mathbf{x}^{\mathrm{post}}_{\mu\nu}$ are presynaptic and postsynaptic patterns, W_{ij} is the connectivity matrix, and θ is a threshold. Each postsynaptic neuron receives l excitatory synapses of equal strength from randomly chosen presynaptic neurons. θ is implicitly set through a winners-take-all process that enforces a desired postsynaptic pattern density a_{post}. Θ is the Heaviside step function, and N is the network size.

EC patterns have $N_{\mathrm{EC}} = 1024$ and $a_{\mathrm{EC}} = 0.1$. To determine N, a, and l for each subsequent region, we turn to estimated biological values and loosely follow their trends. Rodents have approximately 5–10 times more DG granule cells and 2–3 times more CA3 pyramidal cells compared to medial EC layer II principal neurons[14,104,105]. Thus, we choose $N_{\mathrm{DG}} = 8192$ and $N_{\mathrm{CA3}} = 2048$. During locomotion, DG place cells are approximately 10 times less active than medial EC grid cells[106], and MF inputs are expected to be much sparser than PP inputs[15]. Thus, we choose $a_{\mathrm{DG}} = 0.005$, $a_{\mathrm{MF}} = 0.02$, and $a_{\mathrm{PP}} = 0.2$. We do not directly enforce correlation within concepts, which take values $\rho_{\mathrm{EC}} = 0.15$, $\rho_{\mathrm{DG}} = 0.02$, $\rho_{\mathrm{MF}} = 0.01$, and $\rho_{\mathrm{PP}} = 0.09$. Each DG neuron receives approximately 4000 synapses from EC and each CA3 neuron receives approximately 50 MF and 4000 PP synapses[14]. Thus, we choose $l_{\mathrm{DG}} = 205$, $l_{\mathrm{MF}} = 8$, and $l_{\mathrm{PP}} = 205$. Note that for each feedforward projection, postsynaptic statistics a_{post} and ρ_{post} are not expected to depend on l (Eq. (1)).

In Fig. 2E, for the case of $\mathbf{x}^{\mathrm{pre}}_{\mu\nu} = \mathbf{x}^{\mathrm{EC}}_{\mu\nu}$, we use $N_{\mathrm{post}} = 2048$ and $l = 205$. ρ_{post} is obtained by computing correlations between examples within the same concept and averaging over 3 concepts and 8 connectivity matrices. For the case of randomly generated $\mathbf{x}^{\mathrm{pre}}_{\mu\nu}$, we use $N_{\mathrm{pre}} = N_{\mathrm{post}} = 10000$, a single concept, and a single connectivity matrix with $l = 2000$. See Supplementary Information for further details, including the derivation of Eq. (1).

Visualization pathway from CA3 to EC. We train a fully connected linear feedforward network with one hidden layer of size 4096 to map inputs $\mathbf{x}^{\mathrm{MF}}_{\mu\nu}$ to targets $\mathbf{x}^{\mathrm{EC}}_{\mu\nu}$ and inputs $\mathbf{x}^{\mathrm{PP}}_{\mu\nu}$ also to targets $\mathbf{x}^{\mathrm{EC}}_{\mu\nu}$. Batch normalization and a ReLU nonlinearity is applied to the hidden layer and a sigmoid nonlinearity is applied to the output layer. The loss function is

$$\mathcal{L} = \sum_{\substack{\mu\nu\in\\ \text{batch}}} ||\hat{\mathbf{x}}^{\mathrm{EC}}_{\mu\nu} - \mathbf{x}^{\mathrm{EC}}_{\mu\nu}||^2. \tag{7}$$

Training is performed over 100 epochs with batch size 128 using the Adam optimizer with learning rate 10^{-4} and weight decay 10^{-5}.

Hopfield-like model for CA3

Pattern storage. Our Hopfield-like model for CA3 stores linear combinations $\mathbf{q}_{\mu\nu}$ of MF and PP patterns:

$$q_{\mu\nu i} = (1-\zeta)\cdot(x^{\mathrm{MF}}_{\mu\nu i} - a_{\mathrm{MF}}) + \zeta\cdot(x^{\mathrm{PP}}_{\mu\nu i} - a_{\mathrm{PP}}), \tag{8}$$

where $\zeta = 0.1$ is the relative strength of the PP patterns (Fig. 3A). The subtraction of densities from each pattern is typical of Hopfield networks with neural states 0 and 1[91]. The synaptic connectivity matrix is

$$W_{ij} = \frac{1}{N_{\mathrm{CA3}}}\sum_{\mu\nu} q_{\mu\nu i} q_{\mu\nu j}. \tag{9}$$

Pattern retrieval. Cues are formed from target patterns by randomly flipping the activity of a fraction 0.01 of all neurons (Fig. 3B). This quantity is termed *cue inaccuracy*; in Supplementary Fig. 3F, we also consider *cue incompleteness*, which is the fraction of active neurons that are randomly inactivated to form the cue. During retrieval, neurons are asynchronously updated in cycles during which every neuron is updated once in random order (Fig. 3C). The total synaptic input to neuron i at time t is

$$g_i(t) = \sum_j W_{ij} S_j(t) + h_i(t), \tag{10}$$

where $S_j(t)$ is the activity of presynaptic neuron j and $h_i(t)$ is an external input. The external input is zero except for the cue-throughout condition in Fig. 4, in which $\mathbf{h}(t) = \sigma\mathbf{x}$ for noisy MF cue $\mathbf{x}$ and strength $\sigma = 0.2$.

The activity of neuron i at time t is probabilistically updated via the Glauber dynamics

$$P[S_i(t+1) = 1] = \frac{1}{1 + \mathrm{e}^{-\beta[g_i(t) - \theta(t)]}}, \tag{11}$$

where θ is the threshold and β is inverse temperature, with higher β implying a harder threshold. Motivated by theoretical arguments, we define rescaled variables θ' and β' such that $\theta = \theta'\cdot(1-\zeta)^2 a_{\mathrm{MF}}$ and $\beta = \beta'/(1-\zeta)^2 a_{\mathrm{MF}}$[41]. Unless otherwise indicated, we run simulations for 10 update cycles, use $\beta' = 100$, and use $\theta' = 0.5$ to retrieve MF patterns and $\theta' = 0$ to retrieve PP patterns. The rescaled θ' is the threshold value illustrated in Fig. 4A and Supplementary Fig. 4A.

Retrieval evaluation. The overlap between the network activity $\mathbf{S}$ and a target pattern $\mathbf{x}$ is

$$m = \frac{1}{N_{\mathrm{CA3}} a(1-a)}\sum_i S_i(x_i - a), \tag{12}$$

where a is the density of the target pattern. This definition is also motivated by theory[41]. The target pattern $\mathbf{x}^{\mathrm{PP}}_{\mu}$ for PP concept μ is

$$x^{\mathrm{PP}}_{\mu i} = \Theta\left[\sum_{\nu} x^{\mathrm{PP}}_{\mu\nu i} - \phi\right], \tag{13}$$

where ϕ is a threshold implicitly set by using winners-take-all to enforce that $\mathbf{x}^{\mathrm{PP}}_{\mu}$ has density a_{PP}. The theoretical maximum overlap between the network and $\mathbf{x}_\mu^{\mathrm{PP}}$ is the square root of the correlation $\sqrt{\rho_{\mathrm{PP}}}$[41]. Because this estimate is derived for random binary patterns in the large network limit, it can be exceeded in our simulations.

To visualize $\mathbf{S}$, we first recall that the inverse of a dense stored pattern $\mathbf{x}$ with every neuron flipped can also be an equivalent stable state[33]. Thus, we invert $\mathbf{S}$ if we are retrieving PP patterns at low θ and if $m < 0$. Then, we decode its EC representation by passing $\mathbf{S}$ through the feedforward visualization network and binarizing the output with threshold 0.5. Finally, we pass the EC representation through the decoding layers of the image autoencoder.

See Supplementary Information for the determination of network capacity with random binary patterns (Fig. 3H, I and Supplementary Fig. 3G, H) and the definition of oscillation behaviors (Fig. 4C and Supplementary Fig. 4B).

Experimental data analysis

General considerations. To calculate activity, we tabulate spike counts $c(r, \phi)$ over spatial bins r (position or turn direction) and theta phase bins ϕ, and we tabulate trajectory occupancy $u(r, \phi)$ over the same r and distribute them evenly across ϕ. Activity as a function of theta phase, the spatial variable, and both variables are respectively

$$f(\phi)=\frac{\sum_r c(r,\phi)}{\sum_r u(r,\phi)},\quad f(r)=\frac{\sum_\phi c(r,\phi)}{\sum_\phi u(r,\phi)},\quad \text{and}\quad f(r,\phi)=\frac{c(r,\phi)}{u(r,\phi)}. \tag{14}$$

Information per spike as a function of theta phase is calculated by

$$I(\phi)=\sum_r \frac{c(r,\phi)}{c(\phi)}\log_2\frac{f(r,\phi)}{f(\phi)}, \tag{15}$$

where $c(\phi)=\sum_r c(r,\phi)$[107]. To perform sparsity correction for each neuron, we generate 100 null-matched neurons in which the spatial bin of each spike is replaced by a random value uniformly distributed across spatial bins. We subtract the mean $I(\phi)$ over the null matches from the $I(\phi)$ for the true data. To calculate the average difference in information between sparse and dense phases, we first use $f(\phi)$ to partition ϕ into sparse and dense halves. We then average the sparsity-corrected $I(\phi)$ over each half, apply a ReLU function to each half to prevent negative information values, and compute the difference between halves.

See Supplementary Information for dataset preprocessing details.

Model prediction. For the example prediction in Fig. 5A, we choose one concept from Fig. 2A and find 50 neurons that are active in at least one MF example and one PP example within it. For each neuron, we convert each active response to one spike and assign equal occupancies across all examples. We calculate the information per spike across MF examples and across PP examples using example identity ν as the spatial bin r. These values are sparsity-corrected with 50 null-matched neurons, and their difference becomes our example prediction, associating MF encodings with sparse phases and PP with dense.

For the concept prediction in Fig. 6A, we find 50 neurons that are active in at least one MF example and one PP example within any concept. For each neuron, we convert each active response to one spike and collect MF and PP concept responses by summing spikes within each concept. We assign equal occupancies across all concepts. We calculate the information per spike across MF concepts and across PP concepts using concept identity μ as the spatial bin r. We then proceed as in the example case to produce our concept prediction.

Linear track data. Single neurons in Fig. 6 are preprocessed from the CRCNS hc-3 dataset as described in Supplementary Information[24]. To identify place cells, we compute the phase-independent position information per spike using 1 cm-bins across all theta phases, and we select neurons with values greater than 0.5. For each place cell, we bin spikes into various position bins as illustrated and phase bins of width 30°. Since our prediction compares sparse and dense information conveyed by the same neurons, we require at least 8 spikes within each phase value to allow for accurate estimation of position information across all theta phases. To ensure theta modulation, we also require the most active phase to contain at least twice the number of spikes as the least active phase.

Place fields in Fig. 5 are extracted as described in Supplementary Information. Processing occurs similarly to the whole-track case above, except we do not enforce a phase-independent information constraint, we use 5 progress bins, and we require at least 5 spikes within each phase value. Phase precession is detected by performing circular–linear regression between spike progresses and phases[108,109]. The precession score and precession slope are respectively defined to be the mean resultant length and regression slope. Precessing neurons have score greater than 0.3 and negative slope steeper than −72°/field.

W-maze data. Single neurons in Fig. 7A–H are preprocessed from the CRCNS hc-6 dataset as described in Supplementary Information[25]. For each neuron, we bin spikes into 2 turn directions and phase bins of width 45°. Since our prediction compares sparse and dense information conveyed by the same neurons, we require at least 5 spikes within each phase value to allow for accurate estimation of position information across all theta phases. To ensure theta modulation, we also require the most active phase to contain at least twice the number of spikes as the least active phase.

Bayesian population decoding in Fig. 7I–L involves the same binning as in the single-neuron case above, and we enforce a minimum spike count of 30 across all phases instead of a minimum for each phase value. We do not ensure theta modulation on a single-neuron basis. We consider all sessions in which at least 5 neurons are simultaneously recorded; there are 8 valid CA3 sessions and 25 valid CA1 sessions. For each session, we compute the total activity across neurons and turn directions as a function of theta phase to determine the sparsest and densest half of phases (similarly to Eq. (14)). We then compute activities $f_i(r, \psi)$ over each half, indexed by $\psi \in \{\text{sparse, dense}\}$, for neurons i and turn directions r. For each neuron, we rectify all activity values below 0.02 times its maximum.

We decode turn direction during runs along the center arm using sliding windows of width $\Delta t = 0.5$ s and stride 0.25 s. In each window at time t, we tabulate the population spike count $\mathbf{c}(t, \psi)$ over sparse and dense phases ψ. The likelihood that it arose from turn direction r is

$$p(\mathbf{c}(t,\psi)|r)=\prod_i p(c_i(t,\psi)|r)\propto\left(\prod_i f_i(r,\psi)^{c_i(t,\psi)}\right)\exp\left(-\Delta t\sum_i f_i(r,\psi)\right). \tag{16}$$

This formula assumes that spikes are independent across neurons and time and obey Poisson statistics[110]. We only decode with at least 2 spikes. By Bayes's formula and assuming a uniform prior, the likelihood is proportional to the posterior probability $p(r|\mathbf{c}(t,\psi))$ of turn direction r decoded from spikes $\mathbf{c}(t, \psi)$. Consider one decoding that yields $p(R)$ as the probability of a right turn. Its confidence is $|2p(R)-1|$. Its accuracy is 1 if $p(R) > 0.5$ and the true turn direction is right or if $p(R) < 0.5$ and the true direction is left; otherwise, its accuracy is 0.

Machine learning with multilayer perceptrons

Dataset. We use the MNIST dataset of handwritten digits[52]. Each image $\mathbf{i}_\alpha$ is normalized by subtracting the mean value and dividing by the standard deviation across all images and pixels. In addition to its digit class label, we randomly assign a set number. We train networks on a subset of images from the train dataset. To test concept learning through digit classification, we use all held-out images from the test dataset. To test example learning through set identification, we use all train images corrupted by randomly setting 20% of normalized pixel values to 0.

Single-task learning. We train a fully-connected two-layer perceptron with a hyperbolic tangent (tanh) activation function applied to each hidden layer and a softmax activation function applied to the output layer. Each hidden layer contains 50 neurons, and the output layer contains 10 neurons for digit classification and as many neurons as sets for set identification.

Let $\mathbf{s}_\alpha$ be the activations of the final hidden layer for image α. The loss is composed of a cross-entropy loss function between reconstructed labels $\hat{\mathbf{y}}_\alpha$ and true labels $\mathbf{y}_\alpha$, which are one-hot encodings of either digit class or set number, and the DeCorr loss function:

$$\mathcal{L} = -\sum_{\substack{\alpha\in\\ \text{batch}}}\sum_{i=0}^{N-1}\left[y_{\alpha i}\log\hat{y}_{\alpha i} + (1-y_{\alpha i})\log(1-\hat{y}_{\alpha i})\right] + \lambda\mathcal{L}_{\text{DeCorr}}, \tag{17}$$

where

$$\mathcal{L}_{\text{DeCorr}} = \frac{1}{2}\sum_{\substack{\alpha\neq\beta\in\\ \text{batch}}}\frac{\left[\sum_{i=0}^{N-1}(s_{\alpha i}-\bar{s}_\alpha)(s_{\beta i}-\bar{s}_\beta)\right]^2}{\left[\sum_{i=0}^{N-1}(s_{\alpha i}-\bar{s}_\alpha)^2+N\epsilon\right]\left[\sum_{i=0}^{N-1}(s_{\beta i}-\bar{s}_\beta)^2+N\epsilon\right]}. \tag{18}$$

We introduce $\epsilon = 0.001$, which is scaled by the number of hidden layer neurons N, to aid numerical convergence. Mean activations are $\bar{s}_\alpha = (1/N)\sum_{i=0}^{N-1}s_{\alpha i}$. The DeCorr strength is λ; except for Supplementary Fig. 8A, B, we use $\lambda = 0$ for the baseline case and $\lambda = 1$ for the DeCorr case.

We train the network using stochastic gradient descent with batch size 50, learning rate 10^{-4}, and momentum 0.9. In general, we train until the network reaches > 99.9% accuracy with the train dataset. For example, we use 40, 100, and 200 epochs respectively for digit classification and set identification with 10 and 50 sets.

In contrast to DeCorr, the DeCov loss function formulated to reduce overfitting is

$$\mathcal{L}_{\text{DeCov}} = \frac{1}{2}\sum_{i\neq j=0}^{N-1}\left[\sum_{\substack{\alpha\in\\ \text{batch}}}(s_{\alpha i}-\bar{s}_i)(s_{\alpha j}-\bar{s}_j)\right]^2, \tag{19}$$

where mean activations are now taken over batch items: $\bar{s}_i = (1/N_{\text{batch}})\sum_{\alpha\in\text{batch}}s_{\alpha i}$[53].

Multitask learning. We train a fully-connected two-layer perceptron with a hyperbolic tangent (tanh) activation function applied to each hidden layer. In Supplementary Fig. 8E, F, we also consider applying a ReLU activation function to each hidden layer, or a ReLU to the first hidden layer and no nonlinearity to the second. The final hidden layer is fully connected to two output layers, one for digit classification and the other for set identification. A softmax activation function is applied to each layer. Each hidden layer contains 100 neurons, the concept output layer contains 10 neurons, and the example output layer contains as many neurons as sets.

The loss is composed of a cross-entropy loss function between reconstructed $\hat{\mathbf{y}}_\alpha$ and true $\mathbf{y}_\alpha$ digit labels, a cross-entropy loss function between reconstructed $\hat{\mathbf{z}}_\alpha$ and true $\mathbf{z}_\alpha$ set labels, and either the DeCorr or HalfCorr loss function:

$$\begin{aligned}\mathcal{L} = &-\sum_{\substack{\alpha\in\\ \text{batch}}}\sum_{i=0}^{N-1}\left[y_{\alpha i}\log\hat{y}_{\alpha i} + (1-y_{\alpha i})\log(1-\hat{y}_{\alpha i})\right]\\ &-\sum_{\substack{\alpha\in\\ \text{batch}}}\sum_{i=0}^{N-1}\left[z_{\alpha i}\log\hat{z}_{\alpha i} + (1-z_{\alpha i})\log(1-\hat{z}_{\alpha i})\right] + \lambda\mathcal{L}_{\text{DeCorr/HalfCorr}},\end{aligned} \tag{20}$$

where

$$\mathcal{L}_{\text{HalfCorr}} = \frac{1}{2}\sum_{\substack{\alpha\neq\beta\in\\ \text{batch}}}\frac{\left[\sum_{i=N/2}^{N-1}(s_{\alpha i}-\bar{s}_\alpha)(s_{\beta i}-\bar{s}_\beta)\right]^2}{\left[\sum_{i=N/2}^{N-1}(s_{\alpha i}-\bar{s}_\alpha)^2+N\epsilon/2\right]\left[\sum_{i=N/2}^{N-1}(s_{\beta i}-\bar{s}_\beta)^2+N\epsilon/2\right]}. \tag{21}$$

Mean activations are $\bar{s}_\alpha = (2/N)\sum_{i=N/2}^{N-1}s_{\alpha i}$. The DeCorr/HalfCorr strength is λ; we use $\lambda = 1$ with a tanh activation function, $\lambda = 0.04$ with a ReLU, $\lambda = 2$ with no nonlinearity, and $\lambda = 0$ for the baseline case with any nonlinearity.

We train the network using stochastic gradient descent with batch size 50 and learning rate 10^{-4}. In general, we train until the network reaches > 99.9% accuracy in both tasks with the train dataset. For example, we use 100 epochs for the results in Fig. 8I, J.

Reporting summary

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

Data availability

All experimental data used in this study are taken from the Collaborative Research in Computational Neuroscience (CRCNS) hc-3 dataset contributed by György Buzsáki and colleagues[24] and the hc-6 dataset contributed by Loren Frank and colleagues[25]. They are publicly available at https://crcns.org/data-sets/hc. The MNIST dataset[52] used in this study is available at http://yann.lecun.com/exdb/mnist. The FashionMNIST dataset[26] used in this study can be found at https://github.com/zalandoresearch/fashion-mnist. Source data are provided with this paper.

Code availability

All network training and simulation code is available at https://github.com/louiskang-group/kang-2024-examples-concepts.

References

1. Scoville, W. B. & Milner, B. Loss of recent memory after bilateral hippocampal lesions. *J. Neurol. Neurosurg. Psychiatry* **20**, 11 (1957).
2. McNaughton, B. & Morris, R. Hippocampal synaptic enhancement and information storage within a distributed memory system. *Trends Neurosci.* **10**, 408–415 (1987).
3. O'Reilly, R. C. & Rudy, J. W. Conjunctive representations in learning and memory: Principles of cortical and hippocampal function. *Psychol. Rev.* **108**, 311–345 (2001).
4. Rolls, E. T. & Kesner, R. P. A computational theory of hippocampal function, and empirical tests of the theory. *Prog. Neurobiol.* **79**, 1–48 (2006).
5. Bi, G.-q & Poo, M.-m Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **18**, 10464–10472 (1998).
6. Mishra, R. K., Kim, S., Guzman, S. J. & Jonas, P. Symmetric spike timing-dependent plasticity at CA3-CA3 synapses optimizes storage and recall in autoassociative networks. *Nat. Commun.* **7**, 11552 (2016).

7. Amaral, D. & Pierre, L. Hippocampal neuroanatomy. In *The Hippocampus Book*. (eds Andersen, P. et al.). 37–114 (Oxford University Press, 2006).
8. Engin, E. et al. Tonic inhibitory control of dentate gyrus granule cells by α5-containing GABAA receptors reduces memory interference. *J. Neurosci.* **35**, 13698–13712 (2015).
9. Marr, D. Simple memory: a theory for archicortex. *Philos. Trans. R. Soc. B* **262**, 23–81 (1971).
10. O'Reilly, R. C. & McClelland, J. L. Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off. *Hippocampus* **4**, 661–682 (1994).
11. Vinje, W. E. & Gallant, J. L. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* **287**, 1273–1276 (2000).
12. Pitkow, X. & Meister, M. Decorrelation and efficient coding by retinal ganglion cells. *Nat. Neurosci.* **15**, 628–635 (2012).
13. Cayco-Gajic, N. A., Clopath, C. & Silver, R. A. Sparse synaptic connectivity is required for decorrelation and pattern separation in feedforward networks. *Nat. Commun.* **8**, 1116 (2017).
14. Amaral, D. G., Ishizuka, N. & Claiborne, B. Neurons, numbers and the hippocampal network. *Prog. Brain Res.* **83**, 1–11 (1990).
15. Treves, A. & Rolls, E. T. Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network. *Hippocampus* **2**, 189–199 (1992).
16. McClelland, J. L. & Goddard, N. H. Considerations arising from a complementary learning systems perspective on hippocampus and neocortex. *Hippocampus* **6**, 654–665 (1996).
17. Kaifosh, P. & Losonczy, A. Mnemonic functions for nonlinear dendritic integration in hippocampal pyramidal circuits. *Neuron* **90**, 622–634 (2016).
18. Leutgeb, J. K., Leutgeb, S., Moser, M.-B. & Moser, E. I. Pattern separation in the dentate gyrus and CA3 of the hippocampus. *Science* **315**, 961–966 (2007).
19. Aimone, J. B., Deng, W. & Gage, F. H. Resolving new memories: A critical look at the dentate gyrus, adult neurogenesis, and pattern separation. *Neuron* **70**, 589–596 (2011).
20. Borzello, M. et al. Assessments of dentate gyrus function: discoveries and debates. *Nat. Rev. Neurosci.* **24**, 502–517 (2023).
21. Squire, L. R. Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychol. Rev.* **99**, 195–231 (1992).
22. Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C. & Fried, I. Invariant visual representation by single neurons in the human brain. *Nature* **435**, 1102–1107 (2005).
23. Quian Quiroga, R., Kraskov, A., Koch, C. & Fried, I. Explicit encoding of multimodal percepts by single neurons in the human brain. *Curr. Biol.* **19**, 1308–1313 (2009).
24. Mizuseki, K., Sirota, A., Pastalkova, E., Diba, K., & Buzsáki, G. Multiple single unit recordings from different rat hippocampal and entorhinal regions while the animals were performing multiple behavioral tasks. *CRCNS.org*. https://doi.org/10.6080/K09G5JRZ (2013).
25. Karlsson, M., Carr, M., & Frank, L. M. Simultaneous extracellular recordings from hippocampal areas CA1 and CA3 (or MEC and CA1) from rats performing an alternation task in two W-shaped tracks that are geometrically identically but visually distinct. *CRCNS.org*. https://doi.org/10.6080/K0NK3BZJ (2015).
26. Xiao, H., Rasul, K., & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv* 1708.07747. https://doi.org/10.48550/arXiv.1708.07747 (2017).
27. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
28. Lee, H., Battle, A., Raina, R., & Ng, A. Y. Efficient sparse coding algorithms. *Adv. Neural Inf. Process. Syst*. **19**, 801–808 (2006).
29. Makhzani, A. & Frey, B. k-sparse autoencoders. *arXiv* 1312.5663. https://doi.org/10.48550/arXiv.1312.5663 (2014).
30. Chen, Y., Paiton, D., & Olshausen, B. The Sparse Manifold Transform. Advances in Neural Information Processing Systems, pages 10513–10524. Curran Associates, Inc., (2018).
31. Henze, D. A., Wittner, L. & Buzsáki, G. Single granule cells reliably discharge targets in the hippocampal CA3 network in vivo. *Nat. Neurosci.* **5**, 790–795 (2002).
32. Vyleta, N. P., Borges-Merjane, C. & Jonas, P. Plasticity-dependent, full detonation at hippocampal mossy fiber-CA3 pyramidal neuron synapses. *eLife* **5**, 3386 (2016).
33. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554–2558 (1982).
34. Kim, S., Guzman, S. J., Hu, H. & Jonas, P. Active dendrites support efficient initiation of dendritic spikes in hippocampal CA3 pyramidal neurons. *Nat. Neurosci.* **15**, 600–606 (2012).
35. Makara, J. & Magee, J. Variable dendritic integration in hippocampal CA3 pyramidal neurons. *Neuron* **80**, 1438–1450 (2013).
36. Amit, D. J., Gutfreund, H. & Sompolinsky, H. Spin-glass models of neural networks. *Phys. Rev. A* **32**, 1007–1018 (1985).
37. Anderson, J. R. The adaptive nature of human categorization. *Psychol. Rev.* **98**, 409–429 (1991).
38. Love, B. C., Medin, D. L. & Gureckis, T. M. SUSTAIN: A network model of category learning. *Psychol. Rev.* **111**, 309–332 (2004).
39. Ashby, F. G. & Maddox, W. T. Human category learning. *Annu. Rev. Psychol.* **56**, 149–178 (2005).
40. Krizhevsky, A. & Hinton, G. *Learning Multiple Layers of Features from Tiny Images*. Technical Report 0 (University of Toronto, 2009).
41. Kang, L. & Toyoizumi, T. Hopfield-like network with complementary encodings of memories. *Phys. Rev. E* **108**, 054410 (2023).
42. Mizuseki, K. et al. Neurosharing: large-scale data sets (spike, LFP) recorded from the hippocampal-entorhinal system in behaving rats. *F1000Research* **3**, 98 (2014).
43. Skaggs, W. E., McNaughton, B. L., Wilson, M. A. & Barnes, C. A. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* **6**, 149–172 (1996).
44. McAdams, C. J. & Maunsell, J. H. R. Effects of attention on orientation-tuning functions of single neurons in Macaque cortical area V4. *J. Neurosci.* **19**, 431–441 (1999).
45. Isaacson, J. & Scanziani, M. How inhibition shapes cortical activity. *Neuron* **72**, 231–243 (2011).
46. Carandini, M. & Heeger, D. J. Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* **13**, 51–62 (2012).
47. Ferguson, K. A. & Cardin, J. A. Mechanisms underlying gain modulation in the cortex. *Nat. Rev. Neurosci.* **21**, 80–92 (2020).
48. Dotson, N. M. & Yartsev, M. M. Nonlocal spatiotemporal representation in the hippocampus of freely flying bats. *Science* **373**, 242–247 (2021).
49. Frank, L. M., Brown, E. N. & Wilson, M. Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron* **27**, 169–178 (2000).
50. Wood, E. R., Dudchenko, P. A., Robitsek, R. & Eichenbaum, H. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron* **27**, 623–633 (2000).
51. Duvelle, É., Grieves, R. M. & van der Meer, M. A. Temporal context and latent state inference in the hippocampal splitter signal. *eLife* **12**, e82357 (2023).

52. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
53. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *arXiv* 1511.06068. https://doi.org/10.48550/arXiv.1511.06068 (2015).
54. Olshausen, B. A. & Field, D. J. Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* **14**, 481–487 (2004).
55. Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **105**, 2295–2329 (2017).
56. Teyler, T. J. & DiScenna, P. The hippocampal memory indexing theory. *Behav. Neurosci.* **100**, 147–154 (1986).
57. Teyler, T. J. & Rudy, J. W. The hippocampal indexing theory and episodic memory: Updating the index. *Hippocampus* **17**, 1158–1169 (2007).
58. Schapiro, A. C., Gregory, E., Landau, B., McCloskey, M. & Turk-Browne, N. B. The necessity of the medial temporal lobe for statistical learning. *J. Cognit. Neurosci.* **26**, 1736–1747 (2014).
59. Covington, N. V., Brown-Schmidt, S. & Duff, M. C. The necessity of the hippocampus for statistical learning. *J. Cognit. Neurosci.* **30**, 680–697 (2018).
60. Knowlton, B. J. & Squire, L. R. The learning of categories: Parallel brain systems for item memory and category knowledge. *Science* **262**, 1747–1749 (1993).
61. Zeithamova, D., Maddox, W. T. & Schnyer, D. M. Dissociable prototype learning systems: Evidence from brain imaging and behavior. *J. Neurosci.* **28**, 13194–13201 (2008).
62. Mack, M. L., Love, B. C. & Preston, A. R. Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 13203–13208 (2016).
63. Bowman, C. R. & Zeithamova, D. Abstract memory representations in the ventromedial prefrontal cortex and hippocampus support concept generalization. *J. Neurosci.* **38**, 2605–2614 (2018).
64. Manns, J. R., Hopkins, R. O. & Squire, L. R. Semantic memory and the human hippocampus. *Neuron* **38**, 127–133 (2003).
65. Duff, M. C., Covington, N. V., Hilverman, C. & Cohen, N. J. Semantic memory and the hippocampus: Revisiting, reaffirming, and extending the reach of their critical relationship. *Front. Hum. Neurosci.* **13**, 471 (2020).
66. Norman, Y., Raccah, O., Liu, S., Parvizi, J. & Malach, R. Hippocampal ripples and their coordinated dialogue with the default mode network during recent and remote recollection. *Neuron* **109**, 2767–2780 (2021).
67. O'Keefe, J. & Recce, M. L. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* **3**, 317–330 (1993).
68. Terada, S., Sakurai, Y., Nakahara, H. & Fujisawa, S. Temporal and rate coding for discrete event sequences in the hippocampus. *Neuron* **94**, 1248–1262 (2017).
69. Qasim, S. E., Fried, I. & Jacobs, J. Phase precession in the human hippocampus and entorhinal cortex. *Cell* **184**, 3242–3255 (2021).
70. Reddy, L. et al. Theta-phase dependent neuronal coding during sequence learning in human single neurons. *Nat. Commun.* **12**, 4839 (2021).
71. Kay, K. et al. Constant sub-second cycling between representations of possible futures in the hippocampus. *Cell* **180**, 552–567 (2020).
72. Willmore, B. & Tolhurst, D. J. Characterizing the sparseness of neural codes. *Netw. Comput. Neural Syst.* **12**, 255–270 (2001).
73. Ujfalussy, B. B. & Orbán, G. Sampling motion trajectories during hippocampal theta sequences. *eLife* **11**, e74058 (2022).
74. Mehta, M. R., Lee, A. K. & Wilson, M. A. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature* **417**, 741–746 (2002).
75. Souza, B. C. & Tort, A. B. L. Asymmetry of the temporal code for space by hippocampal place cells. *Sci. Rep.* **7**, 8507 (2017).
76. Pfeiffer, B. E. & Foster, D. J. Autoassociative dynamics in the generation of sequences of hippocampal place cells. *Science* **349**, 180–183 (2015).
77. Berry, S. D. & Thompson, R. F. Prediction of learning rate from the hippocampal electroencephalogram. *Science* **200**, 1298–1300 (1978).
78. Seager, M. A., Johnson, L. D., Chabot, E. S., Asaka, Y. & Berry, S. D. Oscillatory brain states and learning: Impact of hippocampal theta-contingent training. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 1616–1620 (2002).
79. Lega, B. C., Jacobs, J. & Kahana, M. Human hippocampal theta oscillations and the formation of episodic memories. *Hippocampus* **22**, 748–761 (2012).
80. Penley, S. C. et al. Novel space alters theta and gamma synchrony across the longitudinal axis of the hippocampus. *Front. Syst. Neurosci.* **7**, 20 (2013).
81. Backus, A. R., Schoffelen, J.-M., Szebényi, S., Hanslmayr, S. & Doeller, C. F. Hippocampal-prefrontal theta oscillations support memory integration. *Curr. Biol.* **26**, 450–457 (2016).
82. Herweg, N. A., Solomon, E. A. & Kahana, M. J. Theta oscillations in human memory. *Trends in Cognitive Sciences* **24**, 208–227 (2020).
83. Jacobs, J., Hwang, G., Curran, T. & Kahana, M. J. EEG oscillations and recognition memory: Theta correlates of memory retrieval and decision making. *Neuroimage* **32**, 978–987 (2006).
84. Zheng, J. et al. Multiplexing of theta and alpha rhythms in the amygdala-hippocampal circuit supports pattern separation of emotional information. *Neuron* **102**, 887–898 (2019).
85. Colgin, L. L. Mechanisms and functions of theta rhythms. *Annu. Rev. Neurosci.* **36**, 295–312 (2013).
86. Hasselmo, M. E. The role of acetylcholine in learning and memory. *Curr. Opin. Neurobiol.* **16**, 710–715 (2006).
87. Hasselmo, M. E., Bodeln, C. & Wyble, B. P. A proposed function for hippocampal theta rhythm: Separate phases of encoding and retrieval enhance reversal of prior learning. *Neural Comput.* **14**, 793–817 (2002).
88. Kunec, S., Hasselmo, M. E. & Kopell, N. Encoding and retrieval in the CA3 region of the hippocampus: A model of theta-phase separation. *J. Neurophysiol.* **94**, 70–82 (2005).
89. Siegle, J. H. & Wilson, M. A. Enhancement of encoding and retrieval functions through theta phase-specific manipulation of hippocampus. *eLife* **3**, e03061 (2014).
90. Leung, L. S. & Law, C. S. H. Phasic modulation of hippocampal synaptic plasticity by theta rhythm. *Behav. Neurosci.* **134**, 595–612 (2020).
91. Tsodyks, M. V. & Feigel'man, M. V. The enhanced storage capacity in neural networks with low activity level. *Europhys. Lett.* **6**, 101–105 (1988).
92. Fontanari, J. F. Generalization in a Hopfield network. *J. Phys.* **51**, 2421–2430 (1990).
93. Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M. & Norman, K. A. Complementary learning systems within the hippocampus: a neural network modelling approach to reconciling episodic memory with statistical learning. *Philos. Trans. R. Soc. B* **372**, 20160049 (2017).
94. Sučević, J. & Schapiro, A. C. A neural network model of hippocampal contributions to category learning. *eLife* **12**, e77185 (2023).

95. Antony, J., Liu, X. L., Zheng, Y., Ranganath, C., & O'Reilly, R. C. Memory out of context: Spacing effects and decontextualization in a computational model of the medial temporal lobe. *bioRxiv* 2022.12.01.518703. https://doi.org/10.1101/2022.12.01.518703 (2023).
96. Whittington, J. C. et al. The Tolman-Eichenbaum Machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263 (2020).
97. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
98. Amer, M. & Maul, T. A review of modularization techniques in artificial neural networks. *Artif. Intell. Rev.* **52**, 527–561 (2019).
99. Kowadlo, G., Ahmed, A., & Rawlinson, D. AHA! an 'Artificial Hippocampal Algorithm' for episodic machine learning. *arXiv* 909.10340. https://doi.org/10.48550/arXiv.1909.10340 (2020).
100. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C. & Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **113**, 54–71 (2019).
101. Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. Deep reinforcement learning. *IEEE Signal Process. Mag.* **34**, 26–38 (2017).
102. Bengio, Y., Léonard, N., & Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv* 1308.3432. https://doi.org/10.48550/arXiv.1308.3432 (2013).
103. Le, Q. V., Karpenko, A., Ngiam, J., & Ng, A. Y. ICA with reconstruction cost for efficient overcomplete feature learning. *Adv. Neural Inf. Process. Syst.* **24**, 1017–1025 (2011).
104. Murakami, T. C. et al. A three-dimensional single-cell-resolution whole-brain atlas using CUBIC-X expansion microscopy and tissue clearing. *Nat. Neurosci.* **21**, 625–637 (2018).
105. Attili, S. M., Silva, M. F. M., Nguyen, T.-v & Ascoli, G. A. Cell numbers, distribution, shape, and regional variation throughout the murine hippocampal formation from the adult brain Allen Reference Atlas. *Brain Struct. Funct.* **224**, 2883–2897 (2019).
106. Mizuseki, K. & Buzsáki, G. Preconfigured, skewed distribution of firing rates in the hippocampus and entorhinal cortex. *Cell Rep.* **4**, 1010–1021 (2013).
107. Skaggs, W. E., McNaughton, B. L., Gothard, K. M., & Markus, E. J. An information-theoretic approach to deciphering the hippocampal code. *Adv. Neural Inf. Process. Syst.* **5**, 1030–1037 (1993).
108. Kempter, R., Leibold, C., Buzsáki, G., Diba, K. & Schmidt, R. Quantifying circular-linear associations: Hippocampal phase precession. *J. Neurosci. Methods* **207**, 113–124 (2012).
109. Kang, L. & DeWeese, M. R. Replay as wavefronts and theta sequences as bump oscillations in a grid cell attractor network. *eLife* **8**, e46351 (2019).
110. Zhang, K., Ginzburg, I., McNaughton, B. L. & Sejnowski, T. J. Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *J. Neurophysiol.* **79**, 1017–1044 (1998).

Acknowledgements

We thank Tom McHugh and Łukasz Kuśmierz for helpful ideas. LK is supported by JSPS KAKENHI for Early-Career Scientists (22K15209) and has been supported by the Miller Institute for Basic Research in Science and a Burroughs Wellcome Fund Collaborative Research Travel Grant. TT is supported by Brain/MINDS from AMED (JP19dm0207001) and JSPS KAKENHI (JP18H05432).

Author contributions

L.K. and T.T. conceptualized the study, analyzed the results, and wrote the manuscript. L.K. trained and simulated the neural networks, performed the mathematical derivations, and analyzed the experimental data.

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/s41467-024-44877-0.

Correspondence and requests for materials should be addressed to Louis Kang.

Peer review information *Nature Communications* thanks Zilong Ji, Joseph Monaco and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Supplementary Figures and Supplementary Methods for “Distinguishing examples while building concepts in hippocampal and artificial networks”

Louis Kang*[1] and Taro Toyoizumi[2]

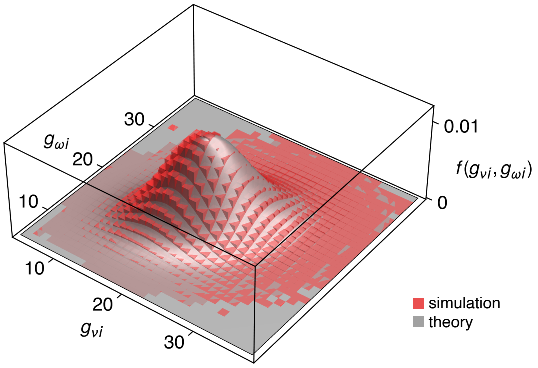
[1]Neural Circuits and Computations Unit, RIKEN Center for Brain Science
[2]Laboratory for Neural Computation and Adaptation, RIKEN Center for Brain Science

Contents

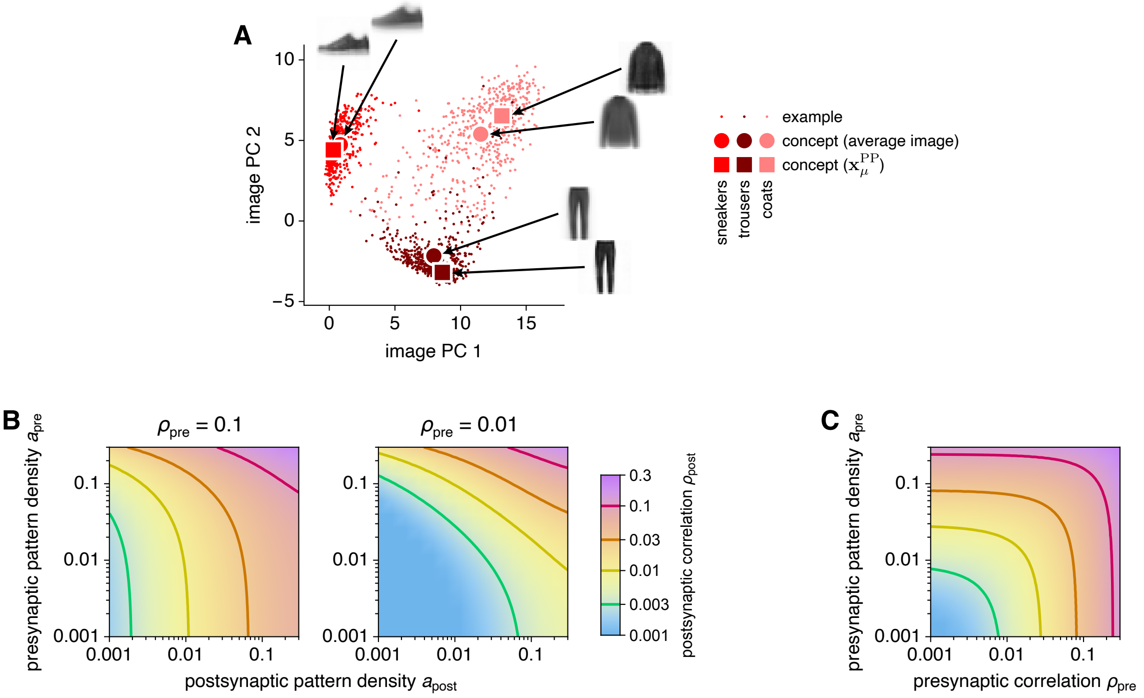
N.B.: All equation numbers below refer to this document and not the main text unless explicitly stated.

*louis.kang@riken.jp

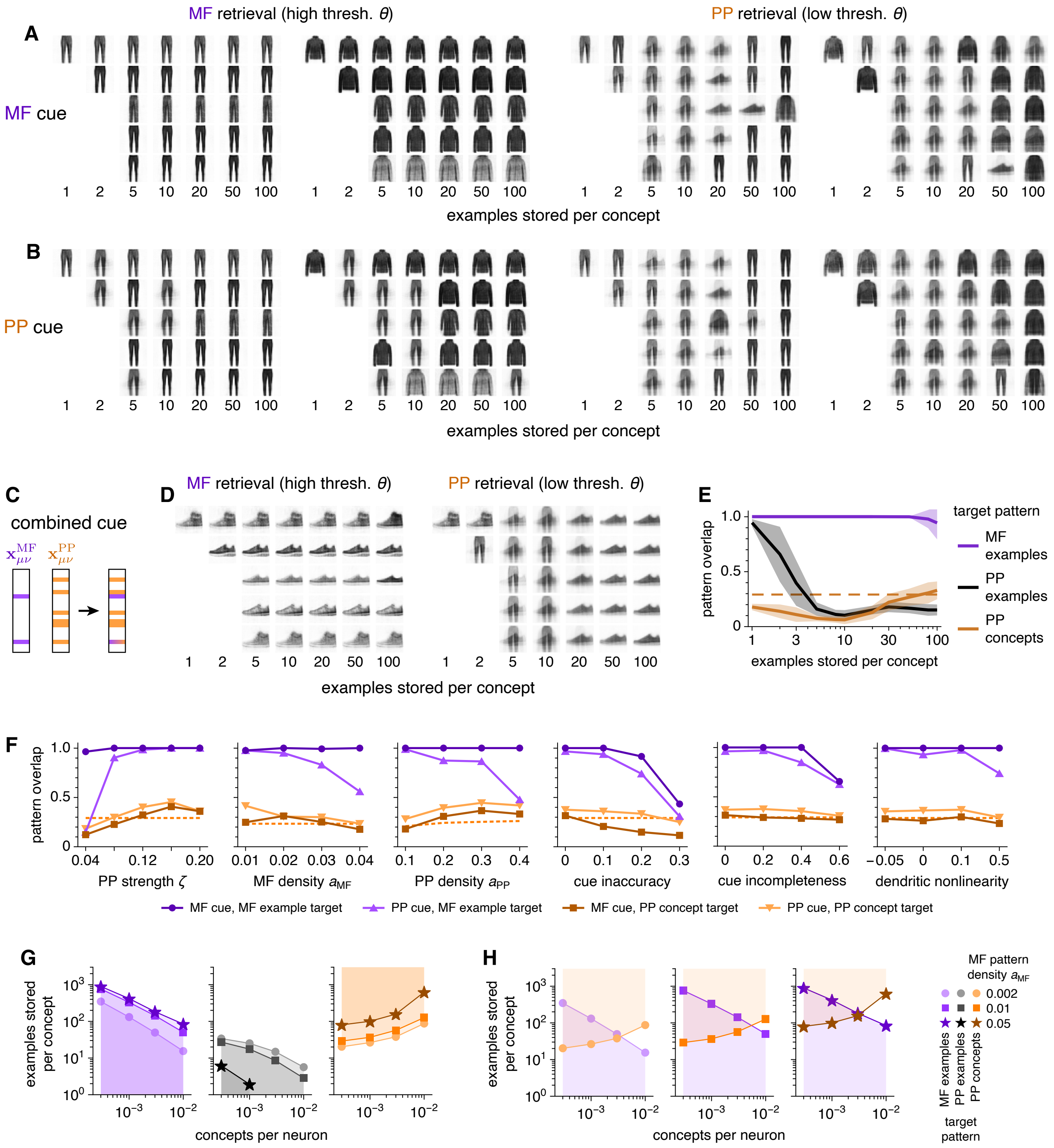
Supplementary Figures



Supplementary Figure 1: Extended results on decorrelation in binary feedforward networks. Joint probability distribution of total inputs $G_{\nu i}$ and $G_{\omega i}$ on postsynaptic neuron i for two patterns $\nu \neq \omega$ (Eq. 12). The theoretically derived probability density function $f(g_{\nu i}, g_{\omega i})$ (Eq. 25) agrees with simulation results. Each g is a sample of the corresponding random variable G. Simulation parameters are 1000 presynaptic neurons, 1000 postsynaptic neurons, synaptic connection probability 0.1, 100 random activity patterns, presynaptic pattern density 0.1, and presynaptic correlation 0.3.



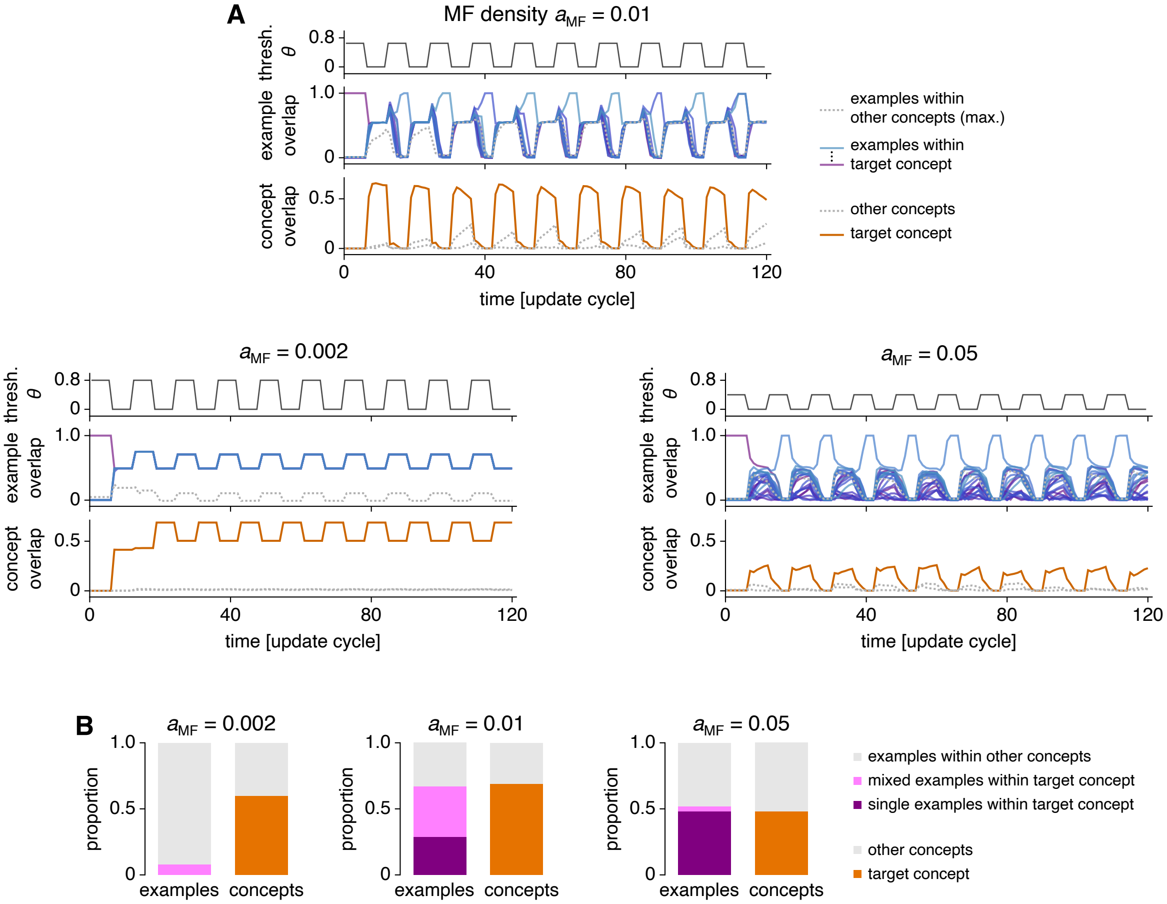
Supplementary Figure 2: Extended results for Fig. 2 of the main text. (**A**) Example images and two definitions of concepts projected along the first two principal components (PCs) of the example dataset. Each concept contains 512 examples. Averaged images are formed by averaging pixel intensities. PP concept target patterns $\mathbf{x}_\mu^{\mathrm{PP}}$ are formed according to Eq. 13 of the main text and then passed through the visualization pathways in Fig. 2F of the main text. (**B**, **C**) Postsynaptic correlation ρ_{post} in binary feedforward networks in Fig. 2D, E. (**B**) ρ_{post} as a function of presynaptic pattern density a_{pre} and postsynaptic pattern density a_{post} for two values of presynaptic correlation ρ_{pre}. On the left, $\rho_{\mathrm{pre}} = 0.1$ corresponds to the red line, and on the right, $\rho_{\mathrm{pre}} = 0.01$ corresponds to the yellow line. Decorrelation occurs in the regions below and to the left of these lines. (**C**) ρ_{post} as a function of presynaptic pattern density a_{pre} and presynaptic correlation ρ_{pre} for postsynaptic pattern density $a_{\mathrm{post}} = 0.1$. Source data are provided as a Source Data file.



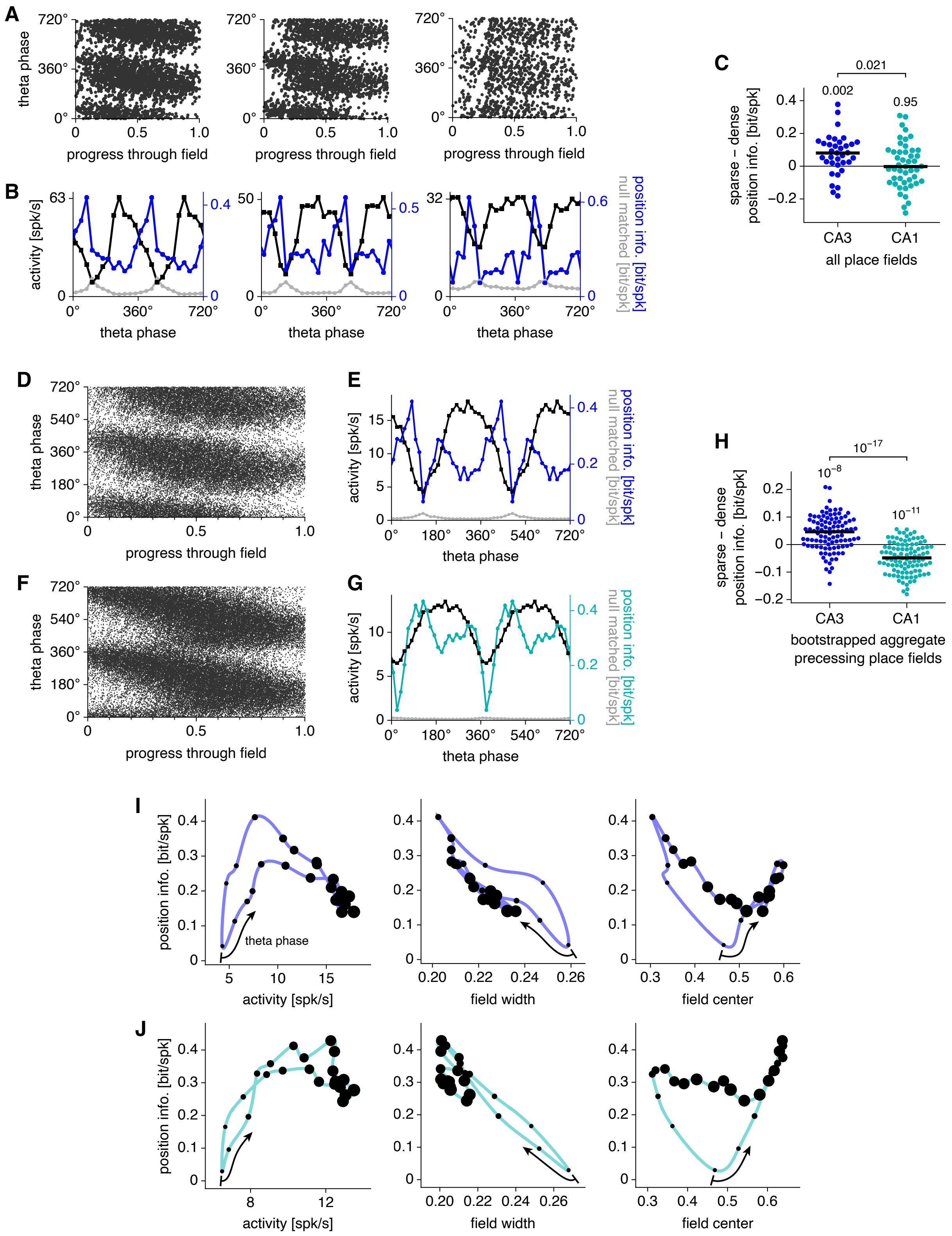
Supplementary Figure 3: Extended results for Fig. 3 of the main text. (**A**) Similar to Fig. 3D, but for trouser and coat concepts. (**B**) Similar to Fig. 3F, but for trouser and coat concepts. (**C**–**E**) Results for cues that combine active neurons from both MF and PP encodings. (**C**) Combined cues formed by the neuron-wise or operation. (**D**) Similar to Fig. 3D, F, but for combined cues. (**E**) Similar to Fig. 3E, G, but for combined cues.

(Continued on the next page.)

Supplementary Figure 3: *(Continued from the previous page.)*
(**F**) Overlaps of retrieved patterns over a wide range of network parameters. MF examples and PP concepts are retrieved at high and low threshold, respectively, optimized by grid search. Cue inaccuracy is the fraction of randomly chosen neurons in the target pattern whose activity is flipped to form the cue. Cue incompleteness is the fraction of randomly chosen active neurons in the target pattern which are inactivated to form the cue. Dendritic nonlinearity η introduces nonlinear summation between MF patterns $\mathbf{x}_{\mu\nu}^{\mathrm{MF}}$ and PP patterns $\mathbf{x}_{\mu\nu}^{\mathrm{PP}}$ by adding a term $\eta x_{\mu\nu i}^{\mathrm{MF}} x_{\mu\nu i}^{\mathrm{PP}}$ to Eq. 8 of the main text. Negative and positive η correspond to sublinear and superlinear regimes, respectively. For PP pattern strength $\gamma = 0.04$, 0.08, 0.12, 0.16, and 0.20, we respectively use example loads $s = 400$, 120, 80, 50, and 20; otherwise, $s = 100$. Points represent means over 4 networks with 15 cues tested in each. (**G**, **H**) Similar to Fig. 3H, I, but for different MF pattern densities a_{MF}. PP patterns have correlation $\rho_{\mathrm{PP}} = 0.04$. Source data are provided as a Source Data file.

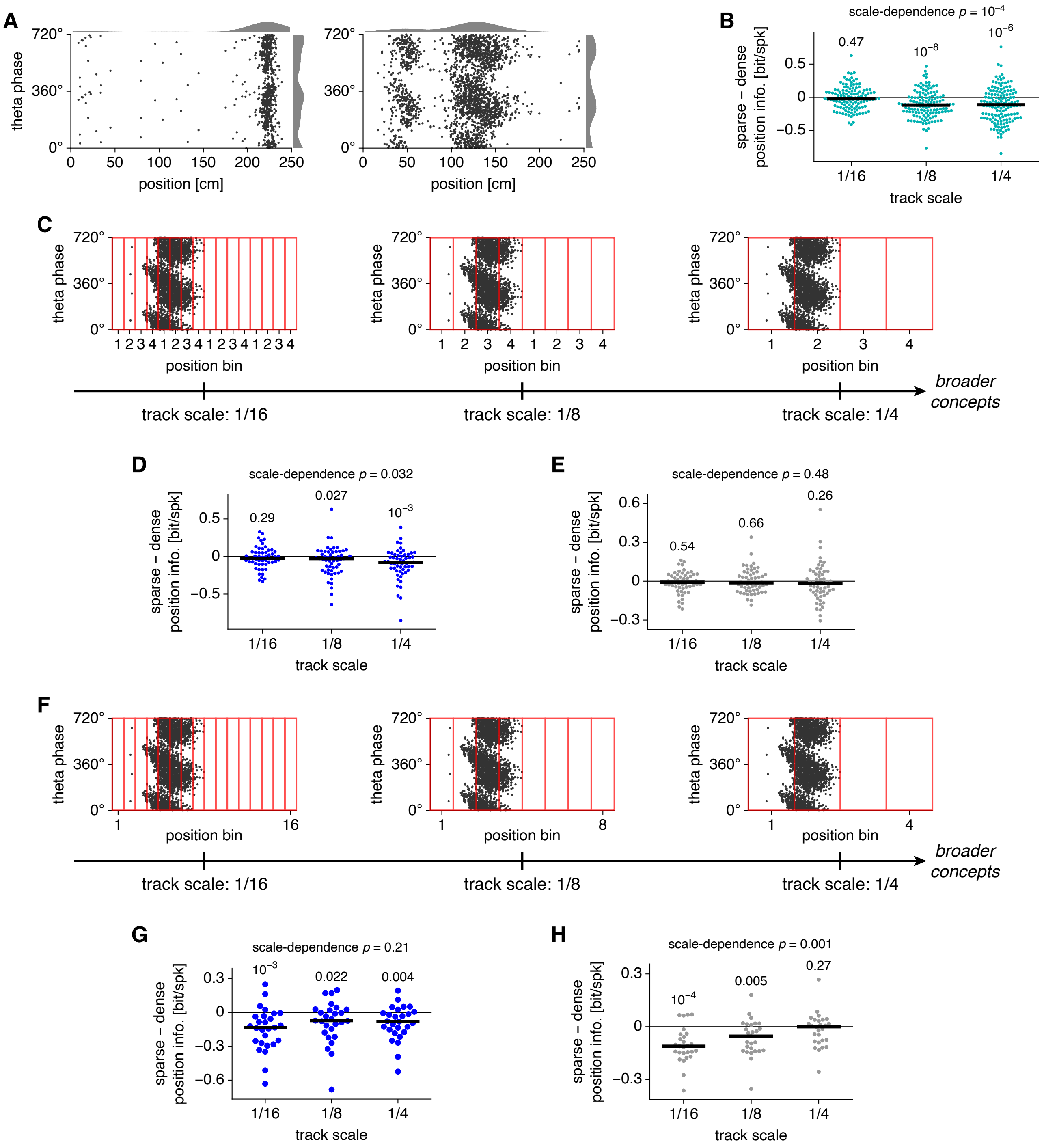


Supplementary Figure 4: Extended results for Fig. 4 of the main text. We use random MF and PP patterns instead of FashionMNIST encodings. (**A**) Similar to Fig. 4A. (**B**) Similar to Fig. 4C. For each scenario in **B**, 10 cues are tested in each of 10 networks. In all networks, 3 concepts are used, MF patterns have correlation 0, and PP patterns have density 0.5 and correlation 0.16. For MF densities $a_{\mathrm{MF}} = 0.002$, 0.01, and 0.05, we store 5, 10, and 20 patterns per concept, respectively. Source data are provided as a Source Data file.



Supplementary Figure 5: Extended results for Fig. 5 of the main text. (**A**–**C**) Additional single-neuron results. (**A**) Similar to Fig. 5F, but for three additional precessing fields from CA3. (**B**) Similar to Fig. 5G, but for the fields in **A**.
(Continued on the next page.)

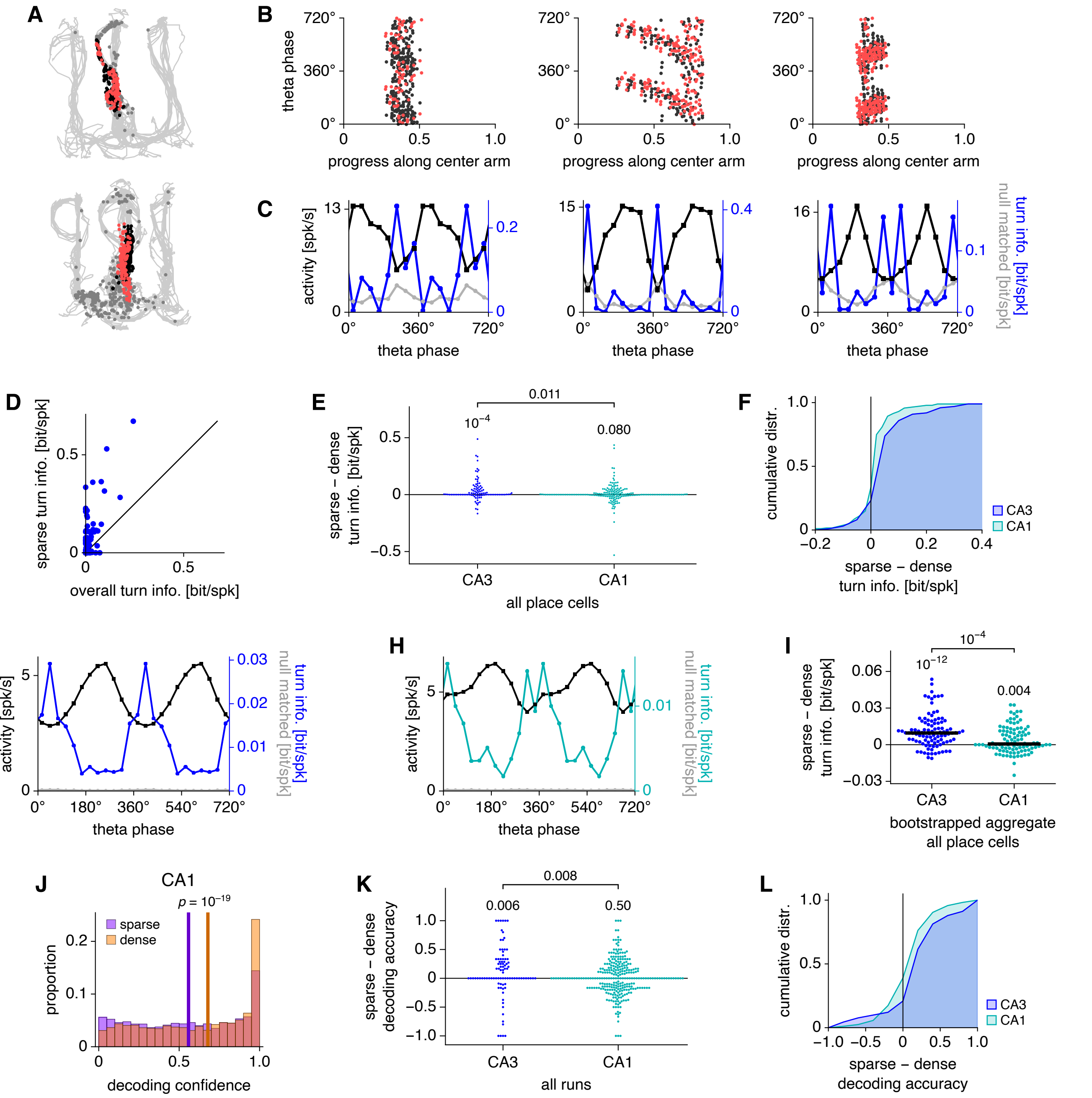
Supplementary Figure 5: *(Continued from the previous page.)*
(**C**) Similar to Fig. 5L, but comparing place fields in CA3 with those in CA1. CA3 $n = 35$ and CA1 $n = 47$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests for each population and by the two-tailed Mann-Whitney U test for the comparison between them. (**D–J**) Results for aggregate fields formed by accumulating spikes across phase precessing place fields. (**D**) Spikes aggregated across 57 CA3 place fields. (**E**) Activity (black), raw position information per spike (blue), and mean null-matched position information (gray) by theta phase for the aggregate field in **D**. (**F**, **G**) Similar to **D**, **E**, but for 55 CA1 place fields. (**H**) Similar to **C**, but comparing 100 bootstrap subsamples per region of 1000 spikes from the aggregate fields in **D** and **F**. (**I**) Parametric plots of features of the aggregate CA3 field in **D** with respect to theta phase. Field width and field center are respectively the standard deviation and mean of progress values over spikes. Each point represents one theta phase and its size is proportional to total activity. Arrows start at the phase with lowest activity and point towards increasing phase. (**J**) Similar to **I**, but for the aggregate CA1 field in **F**. For all results, spikes during each traveling direction are separately analyzed. In **C**, **H**, **I**, and **J**, information is sparsity-corrected with horizontal lines indicating medians. Source data are provided as a Source Data file.



Supplementary Figure 6: Extended results for Fig. 6 of the main text. (**A**) Two additional CA3 place cells along a linear track. For each neuron, spikes are represented by two points at equivalent phases and are accumulated over position (top) and phase (right). (**B**) Similar to Fig. 6E, but for CA1 place cells. Track scale 1/16 $n = 122$, 1/8 $n = 137$, and 1/4 $n = 144$. (**C**–**E**) Similar to Fig. 6C, E, F, but for an alternative method for binning positions across track scales. (**C**) Four bins are still used for all scales, but bins cycle across the whole track. (**D**) For coarser scales, dense phases convey more position information per spike, as in Fig. 6E. Each track scale $n = 56$. (**E**) Shuffled data exhibit no relationship between position information and theta phase across track scales, as in Fig. 6F. (**F**–**H**) Similar to Fig. 6C, E, F, but for a third method for binning positions across track scales. (**F**) Different numbers of bins are used across scales.

(Continued on the next page.)

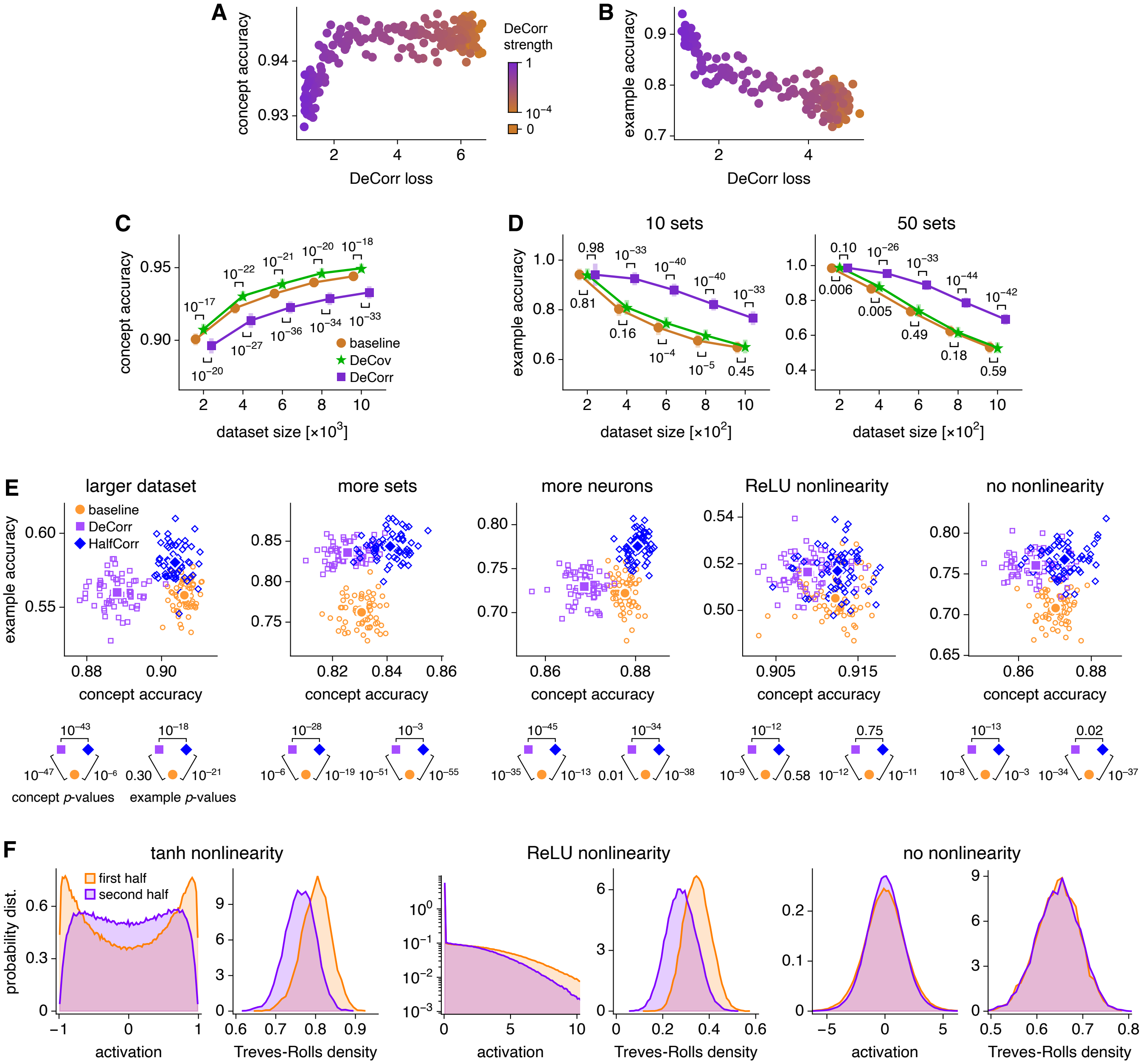
Supplementary Figure 6: *(Continued from the previous page.)*
(**G**) At all scales, dense phases convey more position information per spike, which differs from Fig. 6E. Each track scale $n = 27$.
(**H**) Shuffled data exhibit a relationship between position information and theta phase for finer scales, which differs from Fig. 6F and invalidates this binning method. For all results, spikes during each traveling direction are separately analyzed. In **B**, **D**, **E**, **G**, and **H**, information is sparsity-corrected with horizontal lines indicating medians. Source data are provided as a Source Data file.



Supplementary Figure 7: Extended results for Fig. 7 of the main text. (**A**–**I**) Additional single-neuron results. (**A**) Spikes from Fig. 7C (top) and Fig. 7E (bottom) superimposed on the animal trajectory (light gray line) and other spikes (dark gray points). (**B**) Similar to Fig. 7C, E, but for three additional CA3 place cells. (**C**) Similar to Fig. 7D, F, but for the fields in **B**. (**D**) Average turn information per spike conveyed by CA3 place cells over sparse theta phases and over all phases. Each point represents one neuron, $n = 99$. Note that many neurons convey close to zero overall turn information but convey substantial sparse turn information. (**E**) Similar to Fig. 7G, but comparing place cells in CA3 with those in CA1. CA3 $n = 99$ and CA1 $n = 187$. Numbers indicate p-values calculated by two-tailed Wilcoxon signed-rank tests for each population by the two-tailed Mann-Whitney U test for the comparison between them. (**F**) Cumulative distribution functions for values in **E**. (**G**) Activity (black), raw position information per spike (blue), and mean null-matched position information (gray) by theta phase for spikes aggregated across 98 CA3 place cells. For each place cell, the turn direction with higher activity across all phases is identified. Aggregation is performed by collecting spikes corresponding to more active turn directions and those corresponding to less active directions. (**H**) Similar to **G**, but for 187 CA1 place cells.

(Continued on the next page.)

Supplementary Figure 7: *(Continued from the previous page.)*
(**I**) Similar to **E**, but comparing 100 bootstrap subsamples per region of 1000 spikes from the aggregates analyzed in **G** and **H**. (**J**–**L**) Additional Bayesian population decoding results. (**J**) Similar to Fig. 7J, but for CA1 place cells. (**K**) Similar to Fig. 7K, but comparing runs encoded by CA3 with those encoded by CA1. CA3 $n = 91$ and CA1 $n = 282$. (**L**) Cumulative distribution functions for values in **K**. For all results, spikes during each traveling direction are separately analyzed. In **D**, **E**, **F**, and **I**, information is sparsity-corrected with horizontal lines indicating medians. Source data are provided as a Source Data file.



Supplementary Figure 8: Extended results for Fig. 8 of the main text. (**A**–**D**) Additional results for the single-task architecture in Fig. 8B. (**A**, **B**) Concept and example accuracies as functions of DeCorr loss for 192 networks trained with various strengths of the DeCorr loss function. Increasing DeCorr strength decreases the final DeCorr loss, decreases concept accuracy, and increases example accuracy. Dataset sizes are respectively $10\,000$ and 500 in **A** and **B**, and 100 sets are used in **B**. (**C**, **D**) Similar to Fig. 8E, F, but including networks trained with the DeCov loss function developed by Michael Cogswell and colleagues[1]. Unlike DeCorr, DeCov improves concept accuracy and does not substantially improve example accuracy compared to baseline. (**E**, **F**) Additional results for the multitask architecture in Fig. 8G. (**E**) Similar to Fig. 8I, but for different conditions. In each condition, HalfCorr networks exhibit the best combined performance. From left to right: dataset size of 3000 instead of 1000; 50 sets instead of 10; 500 neurons in each hidden layer instead of 100; ReLU activation function in each hidden layer instead of tanh; and linear activation in the second hidden layer and ReLU activation function in the first hidden layer, which makes the network equivalent to a single-layer perception. (**F**) Activation properties within the final hidden layer of HalfCorr networks with various activation functions described in **E**. Except for the linear activation case, the second, decorrelated half of the layer is sparser than the first, correlated half. Values elicited by 1000 train images in each of 8 trained networks. Treves-Rolls density is the *sparsity* defined in Treves and Rolls[2] and is computed with the absolute value of activations as in Willmore and Tolhurst[3]. Source data are provided as a Source Data file.

Supplementary Methods

Decorrelation in binary feedforward networks

Network architecture

We explore how the correlation of binary activity patterns changes when activity is propagated from one network to another. The two networks are termed presynaptic and postsynaptic. They have sizes N_{pre} and N_{post}. The presynaptic network exhibits activity patterns $x_{\nu i}^{\text{pre}} \in \{0, 1\}$, where $\nu = 1, \ldots, s$ indexes patterns and $i = 1, \ldots, N_{\text{pre}}$ indexes neurons.

W is the connectivity matrix from the presynaptic network to the postsynaptic network. For simplicity, we consider binary synaptic weights, so $W_{ij} \in \{0, 1\}$. The postsynaptic patterns $x_{\nu i}^{\text{post}} \in \{0, 1\}$ are determined by a simple threshold operation:

$$x_{\nu i}^{\text{post}} = \Theta\Bigg[\sum_j W_{ij} x_{\nu j}^{\text{pre}} - \theta\Bigg], \tag{1}$$

where Θ is the Heaviside step function and θ is the activity threshold. We can rewrite Eq. 1 as

$$x_{\nu i}^{\text{post}} = \Theta\big[g_{\nu i} - \theta\big], \quad \text{where} \quad g_{\nu i} = \sum_j W_{ij} x_{\nu j}^{\text{pre}}. \tag{2}$$

Here, $g_{\nu i}$ is the synaptic input onto postsynaptic neuron i for pattern ν.

We characterize these patterns, either presynaptic or postsynaptic, by their density

$$a = \langle x_{\nu i} \rangle \tag{3}$$

and the correlation per neuron between two different patterns $\nu \neq \omega$

$$\rho = \frac{\langle x_{\nu i} x_{\omega i} \rangle - \langle x_{\nu i} \rangle \langle x_{\omega i} \rangle}{\langle x_{\nu i} x_{\nu i} \rangle - \langle x_{\nu i} \rangle \langle x_{\nu i} \rangle} = \frac{\langle x_{\nu i} x_{\omega i} \rangle - a^2}{a(1-a)}. \tag{4}$$

The angle brackets indicate averages over patterns and neurons.

We now assume that the activity patterns and the connectivity matrix are generated via random processes. To be explicit, we will write $X_{\nu i}^{\text{pre}}$ as the random variable for the activity of presynaptic neuron i in pattern ν; the same capitalization applies to the postsynaptic activities $X_{\nu i}^{\text{post}}$ and inputs $G_{\nu i}$. Lowercase letters represent samples of these variables.

We assume that each $X_{\nu i}^{\text{pre}}$ is an identically distributed random variable. We assume that the W_{ij}'s are independent and identically distributed (iid) Bernoulli random variables with parameter u:

$$W_{ij} \sim \text{Ber}(u). \tag{5}$$

In this case, each $X_{\nu i}^{\text{post}}$ is also an identically distributed random variable.

We can then write expressions for density and correlation as

$$\begin{aligned} a &= \mathrm{E}[X_{\nu i}] \\ \rho &= \frac{\mathrm{E}[X_{\nu i} X_{\omega i}] - a^2}{a(1-a)}. \end{aligned} \tag{6}$$

These are population values; Eqs. 3 and 4 indicate the sample estimates and should be written as $\hat{a}$ and $\hat{\rho}$, but we will ignore this distinction.

Generating presynaptic activity patterns

For mathematical tractability, we will enforce density on a per-pattern basis; that is, each presynaptic pattern has the same number of active neurons:

$$n \equiv N_{\text{pre}} a_{\text{pre}} = \sum_i X^{\text{pre}}_{\nu i}. \tag{7}$$

We generate correlated patterns obeying this restriction as follows:

1. We create a concept pattern $\mathbf{x}^{\text{pre}} \in \{0,1\}^{N_{\text{pre}}}$ by randomly choosing n neurons to be 1 and the rest to be 0. Let $\mathcal{S}$ be the set of all active neurons (value 1) and its complement $\mathcal{S}^{\text{c}}$ be the set of all inactive neurons (value 0).
2. To create each example pattern $\mathbf{x}^{\text{pre}}_\nu$, we randomly select a fraction d of neurons in $\mathcal{S}$ and set them to 0. We then randomly select the same number nd of neurons in $\mathcal{S}^{\text{c}}$ and set them to 1.

Now we calculate the correlation ρ_{pre} between the patterns $\mathbf{X}^{\text{pre}}_\nu$ generated by this method. To do so, we investigate the distribution of M, a random variable for the number of active neurons common to two different patterns. We note that

$$M = M_{\mathcal{S}} + M_{\mathcal{S}^{\text{c}}}, \tag{8}$$

where $M_{\mathcal{S}}$ is the number of neurons in $\mathcal{S}$ that remain active in both patterns, and $M_{\mathcal{S}^{\text{c}}}$ is the number of neurons in $\mathcal{S}^{\text{c}}$ that are activated in both patterns. These numbers have hypergeometric distributions:

$$\begin{aligned} M_{\mathcal{S}} &\sim \text{Hyp}\big[n, n(1-d), n(1-d)\big], \\ M_{\mathcal{S}^{\text{c}}} &\sim \text{Hyp}\big[N_{\text{pre}} - n, nd, nd\big]. \end{aligned} \tag{9}$$

We can then calculate

$$\text{E}[X^{\text{pre}}_{\nu i} X^{\text{pre}}_{\omega i}] = \frac{\text{E}[M]}{N_{\text{pre}}} = \frac{1}{N_{\text{pre}}}\left(\frac{n^2(1-d)^2}{n} + \frac{n^2 d^2}{N_{\text{pre}} - n}\right) = a_{\text{pre}}(1-d)^2 + \frac{a_{\text{pre}}^2 d^2}{1 - a_{\text{pre}}}. \tag{10}$$

Substituting this expression into Eq. 6, we obtain

$$\begin{aligned} \rho_{\text{pre}} &= \left(\frac{1 - a_{\text{pre}} - d}{1 - a_{\text{pre}}}\right)^2, \\ d &= \big(1 - a_{\text{pre}}\big)\big(1 - \sqrt{\rho_{\text{pre}}}\big). \end{aligned} \tag{11}$$

Thus, we know the number of activity flips required to produce presynaptic patterns with a given correlation.

Identifying the probability distribution of $G_{\nu i}$

Postsynaptic activity patterns are produced by Eq. 2. Written in terms of random variables, it becomes

$$X^{\text{post}}_{\nu i} = \Theta\big[G_{\nu i} - \theta\big], \quad \text{where} \quad G_{\nu i} = \sum_j W_{ij} X^{\text{pre}}_{\nu j}. \tag{12}$$

Thus, the statistics of $\mathbf{X}_\nu^{\text{post}}$ are determined by $\mathbf{G}_\nu$. To calculate second-order statistics such as ρ_{post}, we need to determine the joint distribution of $G_{\nu i}$ and $G_{\omega i}$ for two patterns $\nu \neq \omega$.

To do so, we define $\mathcal{R}_0$ as the set of active neurons that are common to both patterns $\mathbf{X}_\nu^{\text{pre}}$ and $\mathbf{X}_\omega^{\text{pre}}$. We define $\mathcal{R}_\nu$ and $\mathcal{R}_\omega$ as sets of neurons only active in patterns ν and ω, respectively. We can then write

$$\begin{aligned} G_{\nu i} &= \sum_{j\in\mathcal{R}_0} W_{ij} + \sum_{j\in\mathcal{R}_\nu} W_{ij} \equiv G_{0i} + \tilde{G}_{\nu i} \\ G_{\omega i} &= \sum_{j\in\mathcal{R}_0} W_{ij} + \sum_{j\in\mathcal{R}_\omega} W_{ij} \equiv G_{0i} + \tilde{G}_{\omega i}. \end{aligned} \tag{13}$$

These sets have cardinalities

$$|\mathcal{R}_0| = M, \quad |\mathcal{R}_\nu| = n - M, \quad |\mathcal{R}_\omega| = n - M, \tag{14}$$

where n is given by Eq. 7 and M is given by Eqs. 8 and 9. Since the elements of W are iid Bernoulli random variables (Eq. 5),

$$\begin{aligned} \big(G_{0i} \mid M = m\big) &\sim \text{Bin}(m, u), \\ \big(\tilde{G}_{\nu i} \mid M = m\big) &\sim \text{Bin}(n - m, u), \\ \big(\tilde{G}_{\omega i} \mid M = m\big) &\sim \text{Bin}(n - m, u). \end{aligned} \tag{15}$$

Since $\mathcal{R}_0$, $\mathcal{R}_\nu$, and $\mathcal{R}_\omega$ are mutually disjoint, G_{0i}, $\tilde{G}_{\nu i}$, and $\tilde{G}_{\omega i}$ are mutually independent when conditioned on M.

Calculating the joint probability density function of $G_{\nu i}$ and $G_{\omega i}$

We take the $N_{\text{pre}} \to \infty$ limit in which the discrete probability distributions for $X_{\nu i}$ and $G_{\nu i}$ are replaced by their continuous limits. First, we present a few points on notation.

1. A continuous random variable A has probability density $f_A(a)$ at value $A = a$. We may simplify it as $f(a)$.

2. Variables A and B have joint probability density $f_{A,B}(a, b)$ at values $A = a$ and $B = b$. We may simplify it as $f(a, b)$.

3. Similarly, the conditional probability density given that the random variable B takes value b is $f_{A|B=b}(a)$. We may simplify it as $f(a|b)$.

We can now write the joint probability density function (pdf) for $G_{\nu i}$ and $G_{\omega i}$. For notational convenience, we will drop the subscript i for all relevant variables. The pdf is

$$\begin{aligned} f(g_\nu, g_\omega) &= \int \mathrm{d}g_0\, f(g_\nu, g_\omega, g_0) \\ &= \int \mathrm{d}g_0\, f_{\tilde{G}_\nu, \tilde{G}_\omega, G_0}(g_\nu - g_0, g_\omega - g_0, g_0) \\ &= \int \mathrm{d}m \int \mathrm{d}g_0\, f_{\tilde{G}_\nu, \tilde{G}_\omega, G_0}(g_\nu - g_0, g_\omega - g_0, g_0 \mid m) f(m) \\ &= \int \mathrm{d}m \int \mathrm{d}g_0\, f_{\tilde{G}_\nu}(g_\nu - g_0|m) f_{\tilde{G}_\omega}(g_\omega - g_0|m) f(g_0|m) f(m). \end{aligned} \tag{16}$$

The second line is obtained using the change-of-variables formula for a joint pdf.

We have expressions for each pdf in the integrand. In the large N_{pre} limit, the binomial distributions in Eq. 15 can be approximated by normal distributions (dropping the subscript i for notational convenience):

$$\begin{aligned}
\big(G_0 \mid M = m\big) &\sim \mathcal{N}\big[mu, mu(1-u)\big],\\
\big(\tilde{G}_\nu \mid M = m\big) &\sim \mathcal{N}\big[(n-m)u, (n-m)u(1-u)\big],\\
\big(\tilde{G}_\omega \mid M = m\big) &\sim \mathcal{N}\big[(n-m)u, (n-m)u(1-u)\big].
\end{aligned} \tag{17}$$

Thus, we find

$$\begin{aligned}
&\int \mathrm{d}g_0\, f_{\tilde{G}_\nu}(g_\nu - g_0|m) f_{\tilde{G}_\omega}(g_\omega - g_0|m) f(g_0|m)\\
&\quad \propto \int \mathrm{d}g_0\, \exp\left[-\frac{\big(g_\nu - g_0 - (n-m)u\big)^2}{2(n-m)u(1-u)}\right] \exp\left[-\frac{\big(g_\omega - g_0 - (n-m)u\big)^2}{2(n-m)u(1-u)}\right] \exp\left[-\frac{(g_0 - mu)^2}{2mu(1-u)}\right]\\
&\quad = \int \mathrm{d}g_0\, \exp\left[-\frac{\big[g_0 - \big(\frac{g_\nu + g_\omega}{2} - (n-m)u\big)\big]^2}{(n-m)u(1-u)}\right] \exp\left[-\frac{\big(\frac{g_\nu - g_\omega}{2}\big)^2}{(n-m)u(1-u)}\right] \exp\left[-\frac{(g_0 - mu)^2}{2mu(1-u)}\right]\\
&\quad \propto \exp\left[-\frac{\big(\frac{g_\nu + g_\omega}{2} - nu\big)^2}{(n+m)u(1-u)}\right] \exp\left[-\frac{\big(\frac{g_\nu - g_\omega}{2}\big)^2}{(n-m)u(1-u)}\right].
\end{aligned} \tag{18}$$

We can write the terms inside the exponential as

$$\begin{aligned}
&\frac{\big(\frac{g_\nu + g_\omega}{2} - nu\big)^2}{(n+m)u(1-u)} + \frac{\big(\frac{g_\nu - g_\omega}{2}\big)^2}{(n-m)u(1-u)}\\
&\quad = \frac{\left[\frac{(g_\nu - nu) + (g_\omega - nu)}{2}\right]^2}{(n+m)u(1-u)} + \frac{\left[\frac{(g_\nu - nu) - (g_\omega - nu)}{2}\right]^2}{(n-m)u(1-u)}\\
&\quad = \frac{1}{2}\left[\begin{pmatrix} g_\nu - nu & g_\omega - nu \end{pmatrix} \frac{1}{2u(1-u)} \begin{pmatrix} \frac{1}{n+m} + \frac{1}{n-m} & \frac{1}{n+m} - \frac{1}{n-m} \\ \frac{1}{n+m} - \frac{1}{n-m} & \frac{1}{n+m} + \frac{1}{n-m} \end{pmatrix} \begin{pmatrix} g_\nu - nu \\ g_\omega - nu \end{pmatrix}\right]\\
&\quad = \frac{1}{2}(\mathbf{g} - \boldsymbol{\mu}_G)^\top \boldsymbol{\Sigma}_G^{-1} (\mathbf{g} - \boldsymbol{\mu}_G).
\end{aligned} \tag{19}$$

The last expression is written in terms of the variable vector, mean vector, and covariance matrix for G_ν and G_ω:

$$\begin{aligned}
\mathbf{g} &= \begin{pmatrix} g_\nu \\ g_\omega \end{pmatrix}\\
\boldsymbol{\mu}_G &= \begin{pmatrix} nu \\ nu \end{pmatrix} = \begin{pmatrix} N_{\text{pre}} a_{\text{pre}} u \\ N_{\text{pre}} a_{\text{pre}} u \end{pmatrix}\\
\boldsymbol{\Sigma}_G &= \begin{pmatrix} nu(1-u) & mu(1-u) \\ mu(1-u) & nu(1-u) \end{pmatrix} = \sigma_G^2 \begin{pmatrix} 1 & \rho_G \\ \rho_G & 1 \end{pmatrix},
\end{aligned} \tag{20}$$

where the covariance and correlation are

$$\sigma_G^2 = N_{\text{pre}} a_{\text{pre}} u(1-u) \quad \text{and} \quad \rho_G = \frac{m}{N_{\text{pre}} a_{\text{pre}}}. \tag{21}$$

Combining Eqs. 16, 18, and 19, we obtain

$$f(g_\nu, g_\omega) \propto \int \mathrm{d}m \, \exp\left[-\frac{1}{2}(\mathbf{g}-\boldsymbol{\mu}_G)^\top \boldsymbol{\Sigma}_G^{-1}(\mathbf{g}-\boldsymbol{\mu}_G)\right] f(m). \tag{22}$$

Now we consider $M = M_\mathcal{S} + M_{\mathcal{S}^\mathrm{c}}$ (Eq. 8). From Eq. 9, we see that $M_\mathcal{S}$ and $M_{\mathcal{S}^\mathrm{c}}$ have means and variances

$$\begin{aligned}
\mu_\mathcal{S} &= n(1-d)^2 && = N_\mathrm{pre}a_\mathrm{pre}(1-d)^2 \\
\sigma_\mathcal{S}^2 &= nd^2(1-d)^2 && = N_\mathrm{pre}a_\mathrm{pre}d^2(1-d)^2 \\
\mu_{\mathcal{S}^\mathrm{c}} &= \frac{n^2d^2}{N_\mathrm{pre}-n} && = N_\mathrm{pre}\frac{a_\mathrm{pre}^2 d^2}{1-a_\mathrm{pre}} \\
\sigma_{\mathcal{S}^\mathrm{c}}^2 &= \frac{n^2d^2\big(N_\mathrm{pre}-n(1+d)\big)^2}{(N_\mathrm{pre}-n)^3} && = N_\mathrm{pre}\frac{a_\mathrm{pre}^2 d^2(1-a_\mathrm{pre}-a_\mathrm{pre}d)^2}{(1-a_\mathrm{pre})^3}.
\end{aligned} \tag{23}$$

Note that the flip fraction d can be expressed in terms of a_pre and ρ_pre via Eq. 11.

As $N_\mathrm{pre} \to \infty$, the hypergeometric random variables $M_\mathcal{S}$ and $M_{\mathcal{S}^\mathrm{c}}$ approach normal distributions with the means and variances in Eq. 23. Thus, their distributions become sharply peaked around their means, and we can approximate the pdf of M by a delta-function at its mean:

$$\begin{aligned}
f(m) &\to \delta(m-\mu_M), \quad \text{where} \\
\mu_M &= N_\mathrm{pre}a_\mathrm{pre}(1-d)^2 + N_\mathrm{pre}\frac{a_\mathrm{pre}^2 d^2}{1-a_\mathrm{pre}} \\
&= N_\mathrm{pre}a_\mathrm{pre}(a_\mathrm{pre}+\rho_\mathrm{pre}-a_\mathrm{pre}\rho_\mathrm{pre}).
\end{aligned} \tag{24}$$

We now have our final expression for the joint pdf of $G_{\nu i}$ and $G_{\omega i}$. Reintroducing the neural index i and the normalization factor, Eq. 22 becomes

$$f(g_{\nu i}, g_{\omega i}) = \frac{1}{2\pi\sqrt{\det \boldsymbol{\Sigma}_G}} \exp\left[-\frac{1}{2}(\mathbf{g}_i-\boldsymbol{\mu}_G)^\top \boldsymbol{\Sigma}_G^{-1}(\mathbf{g}_i-\boldsymbol{\mu}_G)\right], \tag{25}$$

where

$$\mathbf{g}_i = \begin{pmatrix} g_{\nu i} \\ g_{\omega i} \end{pmatrix}, \qquad \boldsymbol{\mu}_G = \begin{pmatrix} \mu_G \\ \mu_G \end{pmatrix}, \qquad \boldsymbol{\Sigma}_G = \sigma_G^2 \begin{pmatrix} 1 & \rho_G \\ \rho_G & 1 \end{pmatrix} \tag{26}$$

and

$$\mu_G = N_\mathrm{pre}a_\mathrm{pre}u, \qquad \sigma_G^2 = N_\mathrm{pre}a_\mathrm{pre}u(1-u), \qquad \rho_G = a_\mathrm{pre}+\rho_\mathrm{pre}-a_\mathrm{pre}\rho_\mathrm{pre}. \tag{27}$$

Supplementary Figure 1 shows a plot of the joint pdf $f(g_{\nu i}, g_{\omega i})$ along with a histogram obtained through numerical simulation. The theoretical formula Eq. 25 agrees very well with the numerical data.

Integrating the joint probability density function to obtain a_post and ρ_post

With the joint pdf for $G_{\nu i}$ and $G_{\omega i}$ (Eq. 25), we can compute the postsynaptic pattern density a_post and correlation ρ_post using Eqs. 6 and 12. According to Eq. 12, $X_{\nu i}^\mathrm{post}$ acts as an indicator random variable for $G_{\nu i} > \theta$, and the product $X_{\nu i}^\mathrm{post} X_{\omega i}^\mathrm{post}$ acts as an indicator random variable for $G_{\nu i} > \theta \cap G_{\omega i} > \theta$. Thus,

$$\mathrm{E}[X_{\nu i}^\mathrm{post}] = P(G_{\nu i} > \theta) \quad \text{and} \quad \mathrm{E}[X_{\nu i}^\mathrm{post} X_{\omega i}^\mathrm{post}] = P(G_{\nu i} > \theta \cap G_{\omega i} > \theta). \tag{28}$$

We can calculate

$$\begin{aligned}\mathrm{E}[X^{\text{post}}_{\nu i}] &= \int_\theta^\infty \mathrm{d}g_{\nu i} \int_{-\infty}^\infty \mathrm{d}g_{\omega i}\, f(g_{\nu i}, g_{\omega i}) \\ &= \int_\theta^\infty \mathrm{d}g_{\nu i}\, \frac{1}{\sqrt{2\pi}\sigma_G} \exp\biggl[-\frac{(g_{\nu i}-\mu_G)^2}{2\sigma_G^2}\biggr] \\ &= \frac{1}{2}\operatorname{erfc}\frac{\theta-\mu_G}{\sqrt{2}\sigma_G}. \end{aligned} \tag{29}$$

Thus, the postsynaptic pattern density is immediately

$$a_{\text{post}} = \mathrm{E}[X^{\text{post}}_{\nu i}] = \frac{1}{2}\operatorname{erfc}\frac{\phi}{\sqrt{2}}, \quad \text{where} \quad \phi = \frac{\theta-\mu_G}{\sigma_G}. \tag{30}$$

The rescaled threshold ϕ is the standardized version of θ.

We next need to calculate

$$\begin{aligned}\mathrm{E}[X^{\text{post}}_{\nu i}X^{\text{post}}_{\omega i}] &= \int_\theta^\infty \mathrm{d}g_{\nu i} \int_\theta^\infty \mathrm{d}g_{\omega i}\, f(g_{\nu i}, g_{\omega i}) \\ &= \frac{1}{2\pi\sqrt{\det \boldsymbol{\Sigma}_G}} \int_\theta^\infty \mathrm{d}g_{\nu i} \int_\theta^\infty \mathrm{d}g_{\omega i}\, \exp\biggl[-\frac{1}{2}(\mathbf{g}_i-\boldsymbol{\mu}_G)^\top \boldsymbol{\Sigma}_G^{-1}(\mathbf{g}_i-\boldsymbol{\mu}_G)\biggr]. \end{aligned} \tag{31}$$

By standardizing the variables of integration with $h_{\nu i} = (g_{\nu i}-\mu_G)/\sigma_G$, this integral can be expressed in terms of the standard bivariate normal:

$$\mathrm{E}[X^{\text{post}}_{\nu i}X^{\text{post}}_{\omega i}] = \frac{1}{2\pi\sqrt{1-\rho_G^2}} \int_\phi^\infty \mathrm{d}h_{\nu i} \int_\phi^\infty \mathrm{d}h_{\omega i}\, \exp\biggl[-\frac{h_{\nu i}^2 + h_{\omega i}^2 - 2\rho_G h_{\nu i}h_{\omega i}}{2(1-\rho_G^2)}\biggr]. \tag{32}$$

This double integral cannot be evaluated in closed form, but we can reduce it to a single integral[4]:

$$\mathrm{E}[X^{\text{post}}_{\nu i}X^{\text{post}}_{\omega i}] = \Gamma[\phi, \rho_G] \equiv \frac{1}{2\pi} \int_{\arccos \rho_G}^{\pi} \mathrm{d}\psi\, \exp\biggl[-\frac{\phi^2}{1+\cos\psi}\biggr]. \tag{33}$$

Therefore, the expression for the postsynaptic correlation follows:

$$\rho_{\text{post}} = \frac{\Gamma[\phi,\rho_G] - a_{\text{post}}^2}{a_{\text{post}}(1-a_{\text{post}})}, \tag{34}$$

where a_{post} can be expressed in terms of the standardized threshold ϕ with Eq. 30. On the other hand, we can stipulate a desired a_{post} and then recover ϕ and ρ_{post} with

$$\begin{aligned}\phi &= \sqrt{2}\operatorname{erfc}^{-1}(2a_{\text{post}}), \\ \rho_{\text{post}} &= \frac{\Gamma\bigl[\sqrt{2}\operatorname{erfc}^{-1}(2a_{\text{post}}),\, a_{\text{pre}} + \rho_{\text{pre}} - a_{\text{pre}}\rho_{\text{pre}}\bigr] - a_{\text{post}}^2}{a_{\text{post}}(1-a_{\text{post}})}. \end{aligned} \tag{35}$$

This is Eq. 1 of the main text. Figure 2E shows that this formula for the postsynaptic correlation agrees well with values obtained through numerical simulation across a variety of parameter values.

Exploring ρ_{post} as a function of a_{pre}, a_{post}, and ρ_{pre}

In Supplementary Fig. 2B, we plot ρ_{post} as a function a_{pre} and a_{post} for various ρ_{pre}. We see that decorrelation ($\rho_{\text{post}} < \rho_{\text{pre}}$) occurs when a_{post} is low. Thus, a downstream (postsynaptic) network with many neurons but low activity naturally decorrelates patterns of the upstream (presynaptic) network. The low activity can be achieved by low connectivity u or a high threshold θ; Eq. 35 does not differentiate between the two.

Supplementary Figure 2C demonstrates that if presynaptic patterns are sparse and decorrelated (lower left corner of the plot), postsynaptic patterns also exhibit low correlation even if they are denser. Thus, once patterns are sparsified and decorrelated, they will remain decorrelated for subsequent feedforward layers. Note that this panel also shows the symmetry in interchanging $a_{\text{pre}} \leftrightarrow \rho_{\text{pre}}$ present in Eq. 35.

CA3 model with random binary patterns

We use a network size of $N_{\text{CA3}} = 10\,000$. We generate MF example patterns $\mathbf{x}_{\mu\nu}^{\text{MF}}$ with desired density a_{MF} and correlation 0 by randomly activating $N_{\text{CA3}} a_{\text{MF}}$ neurons. We generate PP concept patterns $\mathbf{x}_{\mu}^{\text{PP}}$ with desired density 0.5 by randomly activating each neuron with probability 0.5. We then generate PP example patterns $\mathbf{x}_{\mu\nu}^{\text{PP}}$ with desired correlation ρ_{PP} by randomly flipping each concept neuron with probability $(1 - \sqrt{\rho_{\text{PP}}})/2$. Simulations are initiated without cue noise and a sharp activation threshold ($\beta \to \infty$) to assess the best possible network performance.

To calculate capacities in Fig. 3H, I, we use a higher strength of PP inputs $\zeta = 0.2$ to make the capacity values more computationally accessible. For a given load of concepts per neuron, we perform a grid search over the load of examples per concept using 8 networks per load and testing 20 cues in each network. Each cue is identical to its target pattern. We search for the load at which the average overlap crosses a threshold, which is $1/2$, $(1 + \rho_{\text{PP}})/2$, and $(1 + \sqrt{\rho_{\text{PP}}})/2$ for MF examples, PP examples, and PP concepts, respectively, to account for the positive overlap of off-target PP patterns if they are correlated[5]. For MF examples, we explore activity thresholds θ' between 0.43 and 0.85 and use the value that maximizes average overlap. For PP examples and concepts, we set the activity threshold θ' to 0.

CA3 model behavior during oscillating threshold

The oscillation analysis in Fig. 4C characterizes network behavior between update cycles 60 and 120. Consider a single oscillation cycle. For example-related behavior, we consider its high-threshold half. Let $m_1(t)$ and $m_2(t)$ be the largest and second-largest overlaps within the target concept at time t, and let $m_0(t)$ be the largest overlap within other concepts. If $m_1(t) > 0.8$, $m_1(t) > 2 \cdot m_2(t)$, and $m_1(t) > 2 \cdot m_0(t)$ at any time, the behavior of the oscillation cycle is categorized as *single examples within a target concept*. If $m_1(t) > 0.8$, $m_1(t) < 2 \cdot m_2(t)$, and $m_1(t) > 2 \cdot m_0(t)$, the behavior is categorized as *mixed examples within a target concept*. If at least half of the oscillation cycles receive a certain categorization, it is considered the network behavior. Otherwise, the network behavior is *examples within other concepts*. For concept-related behavior, we consider the low-threshold half of each oscillation cycle. Let $m_1(t)$ be the overlap with the target concept and $m_0(t)$ be the largest overlap with other concepts. If $m_1(t) > 0.1$ and $m_1(t) > 2 \cdot m_0(t)$ at any time, the behavior of the oscillation cycle is categorized as *target concept* (for the random binary patterns in Supplementary Fig. 4B, we use $m_1(t) > 0.2$ instead). If at least half of the oscillation cycles receive this categorization, it is considered the network behavior. Otherwise, the network behavior is *other concepts*.

Experimental data preprocessing

In all experimental analyses, we consider each traveling direction separately. Thus, each recorded neuron effectively yields two neurons in our analyses with their own spikes and trajectory occupancies.

Linear track data from the CRCNS hc-3 dataset is used to produce the results in Figs. 5 and 6[6]. For CA3, we use all linear track sessions from rats ec013, ec016, gor, and vvp with CA3 neurons recorded, and for CA1, we use all linear track sessions between 761 and 882 from rat ec013. For rats gor and vvp, the size of the linear track was changed during recording, and we only consider data before the change. Animal positions are taken to be the mean of the two LED lights on the microdrive. The track axis is taken to be the first principal component of sampled positions. Animal velocities are differences between tracked position samples divided by the sampling rate and smoothed with a Gaussian filter with standard deviation 0.1 s.

We use the recommended quality criteria involving eDist, RefracRatio, and RefracViol when selecting units. Since we are interested in the theta oscillation, we only consider spikes occurring during locomotion with speed greater than 10 cm/s. We require units to have at least 50 spikes occurring within the central 70% of the track to avoid neurons whose behavior may be dominated by boundary effects. To determine the theta signal for each unit, we identify the tetrode from which it was recorded and average the LFP over all channels on the tetrode. This signal is bandpass-filtered between 6–10 Hz, and the complex argument of its Hilbert transform is the local theta phase.

To extract place fields for Fig. 5, we first discretize track positions with 1 cm bins. For a given place cell, we first compute the activity across all theta phases as a function of position and apply a Gaussian filter whose standard deviation is 0.01 times the track length. We find activity peaks whose maximum is 0.6 standard deviations above the mean; if the activity exhibits multiple peaks while remaining above this threshold, the largest is chosen. We then find the closest flanking positions where the activity falls below 0.2 times the peak value. The region in between is the place field. If two place fields overlap, they are divided at the activity minimum located between their peaks.

W-maze data from the CRCNS hc-6 dataset is used to produce the results in Fig. 7[7]. We use all run sessions from rats bon, con, dud, fra, mil, and ten. We remove 2 sessions from dud and 3 sessions from fra without position samples in one of the side arms. Animal positions and velocities are taken directly as the 30 Hz-interpolated samples from the dataset. We only consider spikes occurring during locomotion with speed greater than 5 cm/s. We require units to have at least 30 spikes occurring within the center arm. Spike theta phases are taken directly from the dataset.

To extract arm identity from the position samples, we first linearly rescale both position coordinates to span from 0 to 1. We define a maze skeleton consisting of lines that represent the left arm from $(0.1, 0.1)$ to $(0.1, 1)$, the center arm from $(0.5, 0.1)$ to $(0.5, 1)$, the right arm from $(0.9, 0.1)$ to $(0.9, 1)$, and the base from $(0.1, 0.1)$ to $(0.9, 0.1)$. We then fit transformations of this skeleton to the position samples, allowing for stretching along the first coordinate, rotation about its center, and translations. The fit is performed by minimizing the total squared distance between each sampled position and its closest point on the transformed skeleton. Empirically, this process yields excellent fitting without any need for manual intervention. Then, each position sample is assigned to either the left, center, or right arm based on closest distance.

To extract runs along the central arm, we first smooth arm samples by encoding arm identity as a one-hot vector, applying a Gaussian filter with standard deviation 0.2 s, and identifying the largest element in each sample. We then consider each span of central-arm samples. We find the times within the span at which the animal crosses scaled positions 0.4 and 0.7, where 0 corresponds to the smallest position value at the base of the maze and 1 corresponds to the largest position value at the far end of the arms. The run duration must be between 0.1 s and 10 s, and we then pad the run by 0.5 s at both ends. Outward runs cross scaled position 0.7 before 0.4, and the subsequent arm identity is used to determine the future turn direction. Inward runs cross scaled position 0.4 before 0.7, and the previous arm identity is used to determine the past turn direction.

Experimental data aggregate analysis

The aggregate fields in Supplementary Fig. 5D–J are formed from phase-precessing place fields. We do not enforce a minimum spike count or ensure theta modulation on a single-neuron basis. In total, we collect 19 678 spikes from 57 CA3 place fields and 29 664 spikes from 55 CA1 place fields. We perform bootstrapping by sampling with replacement 1000 spikes at a time. For each subsample, we bin spikes into 10 progress bins and phase bins of width 15°.

The aggregate fields in Supplementary Fig. 7G–I are formed from place cells with at least 30 spikes within the central arm. We do not ensure theta modulation on a single-neuron basis. For each neuron, we identify the turn directions with higher and lower activities, and collect spikes occurring during each condition. In total, we collect 47 196 spikes from 331 CA3 place cells and 72 461 spikes from 436 CA1 place cells. We perform bootstrapping by sampling with replacement 1000 spikes at a time. For each subsample, we bin spikes into 2 directions (more active and less active) and phase bins of width 30°.

References

[1] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv* 1511.06068, 2015.

[2] A. Treves and E. T. Rolls. What determines the capacity of autoassociative memories in the brain? *Netw. Comput. Neural Syst.*, 2(4):371–397, 1991.

[3] B. Willmore and D. J. Tolhurst. Characterizing the sparseness of neural codes. *Netw. Comput. Neural Syst.*, 12(3):255–270, 2001.

[4] D. B. Owen. Tables for computing bivariate normal probabilities. *Ann. Math. Stat.*, 27(4):1075–1090, 1956.

[5] L. Kang and T. Toyoizumi. Hopfield-like network with complementary encodings of memories. *Phys. Rev. E*, 108(5):054410, 2023.

[6] K. Mizuseki, A. Sirota, E. Pastalkova, K. Diba, and G. Buzsáki. Multiple single unit recordings from different rat hippocampal and entorhinal regions while the animals were performing multiple behavioral tasks. *CRCNS.org*, 2013.

[7] M. Karlsson, M. Carr, and L. M. Frank. Simultaneous extracellular recordings from hippocampal areas CA1 and CA3 (or MEC and CA1) from rats performing an alternation task in two W-shapped tracks that are geometrically identically but visually distinct. *CRCNS.org*, 2015.