

Reconciling flexibility and efficiency: medial entorhinal cortex represents a compositional cognitive map

Received: 5 July 2024

Payam Piray¹✉ & Nathaniel D. Daw²

Accepted: 28 July 2025

Published online: 12 August 2025

 Check for updates

The influential concept of cognitive maps envisions that the brain builds mental representations of objects, barriers, and goals. Computational models show how these representations guide goal-directed behavior, such as planning novel routes to maximize rewards. One key feature of flexible cognitive representations is compositionality, the ability to build complex structures by recombining simpler parts. However, how this applies to neural representations of cognitive maps and map-based planning remains unclear. Compositionality can be difficult to reconcile with efficient planning, as reusing components may limit efficiency. Here, we propose a novel computational model for efficiently creating and planning with compositional predictive maps, which successfully reproduces response fields in the medial entorhinal cortex, particularly object vector cells and grid cells. The model treats each object as an alteration to a baseline map linked to open space, creating predictive maps by combining object-related representations compositionally, providing insights into brain processes supporting efficient, flexible planning.

The principle of compositionality—the ability to reuse representations constructively to build up novel, complex ideas from simpler parts—is widely viewed as fundamental to intelligence. In cognitive science and neuroscience, it is thought that the brain’s impressive capacities for perception, language, and reasoning arise, at least partly, from its compositional nature, assembling thoughts and concepts from simpler components^{1–5}. Similarly, in artificial intelligence, compositionality is seen as an important capability for developing human-level artificial general intelligence. While many modern artificial intelligence systems do not yet exhibit human-like compositional abilities^{4,6–9}, creating models that demonstrate compositional behavior has been a central goal driving advances in the field^{10–15}.

One case in which structural learning has been studied in detail is that of trial-and-error action selection, particularly reward maximization in sequential tasks such as spatial mazes. It has long been argued that animals accomplish this, at least in part, by learning a “cognitive map” or internal representation of the environment and task^{16–20}. This is thought to enable them, for instance, to plan new routes to novel goals and find detours and shortcuts. In turn, these functions have

been widely formalized computationally using a family of algorithms from reinforcement learning (RL), known as model-based learning because they rely on a learned internal model of the environment’s contingencies.

Interestingly, however, although such map or model learning also might seem to involve the composition of meaningful subparts (e.g., a building interior is made up of hallways, rooms, and doors) and this has occasionally been suggested^{21–23}, this aspect has not been widely studied or theoretically developed. Instead, map learning in the RL setting has most often been formally characterized as akin to individually discovering and memorizing each of the edges in a graph^{24,25}. Here, we suggest that this gap reflects a key challenge in constructing and using such maps: creating representations that balance competing demands of flexibility vs. efficiency. In general, using a (“one-step”) map or model to flexibly plan new paths to goals requires significant computation to search iteratively over candidate paths. Meanwhile, optimizations that address this by reusing or “caching” some of this work (for instance, chunking useful extended action sequences) generally simplify computation at the expense of reducing generalizability

¹Department of Psychology, University of Southern California, Los Angeles, CA, USA. ²Department of Psychology, and Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA. ✉e-mail: piray@usc.edu

to new situations (e.g., where different action sequences would be better). Exploiting compositionality in this setting thus remains a challenge because representations that cache demanding computations for efficient planning resist flexible, compositional construction. Accordingly, to the extent that a few previous theories of cognitive maps have captured some compositional-like relational structure in the task graph^{16,26,27}, these models have not generally addressed using these maps to plan or seek reward, and addressing this would require exhaustive and arguably implausible computation at decision time.

Here we introduce compositionality into an alternative class of RL models (known as Successor Representation (SR) models) which have recently become popular for enabling efficient planning, by representing the cognitive map in terms of long-range predictive dependencies across the task space rather than by local adjacency^{28–32}. In particular, we build on a recent development of this class (the “default representation”³⁰); that, by introducing a particular non-linearity and some linear algebraic tricks, largely addresses the issues of generalizability and flexibility. Here we extend this model to address two types of open issues in this domain. First, computationally, we introduce compositionality so as to address how such predictive maps can be built quickly but also in a generalizable way by recombining precomputed, reusable representations. Second, neurally, we argue that a number of aspects about how this model decomposes maps into stable and reusable pieces shed new light on observed neuronal firing patterns in the medial entorhinal cortex (MEC), which previous modelers have suggested may subserve a neural representation of maps of this sort³². Specifically, we argue that previous models fail to account for important aspects of cellular responses in the MEC showing composition-like representations, such as invariant and modular coding response in MEC object vector cells, and how these coexist with fast and stable coding in MEC grid cells.

More specifically, we propose a new computational model for constructing compositional predictive maps that can use the resulting maps to perform complex planning tasks efficiently and flexibly. The presented model is based on the idea that each object can be seen as an alteration to a baseline predictive map of an open-field space. For each object, we derive a representation of its contribution to the predictive map that does not depend on the specific position or rotation of the object in the task space. The complete predictive map of a state space consisting of multiple objects is then built by putting together the representations related to each object. The resulting map can then be used directly and efficiently for planning, without laborious search.

This approach is grounded in several key computational principles. First, objects, barriers, and goals act as perturbations that alter the relational structure of open space. Second, their effects on long-range predictive maps can be efficiently computed using translation- and rotation-invariant matrices, which we term predictive object representations (PORs), whose size depends on the object rather than the full state space. Third, long-range maps can be efficiently built by compositionally integrating these compact PORs. Fourth, an efficient set of basis functions allows the cognitive map to be represented by combining PORs with the eigenbasis of open space. Finally, a learning algorithm enables PORs of complex objects to be constructed from more primitive components, enhancing the model’s adaptability and efficiency.

With regard to the function of the MEC, the model suggests that different spatially tuned cells within the structure can be understood as encoding components of the compositional predictive map. In particular, the theory accounts for a class of cells discovered recently in the MEC, object vector cells, which fire exclusively when animals are near objects^{33–35}. These cells are the most ubiquitous class of cells in the MEC, accounting for roughly 30% of all spatially tuned cells in this region³⁴. According to our model, these cells encode the building blocks of compositionality, a planning-ready computational

representation of objects, borders, and other components of the map specialized in a planning-ready way that enables the system to flexibly build a map and easily use it for planning. In turn, our theory interprets grid cells as providing an efficient code that provides a baseline map over which these components can be incorporated. We show that this successfully models various empirical findings concerning the firing patterns of grid cells following the introduction of barriers and goals.

Results

Default representation (DR)

We build on our recent computational model of map learning and planning in the brain, linear RL³⁰, which exploits long-range predictive maps for planning and shows animal-like behavior across various replanning tasks, such as reward reevaluation (e.g., Tolman’s latent learning task) or transition reevaluation (e.g., Tolman’s detour task).

To briefly summarize (see “Methods”), as with other RL theories, linear RL formalizes planning as computing the value function, or the expected sum of future rewards for different possible courses of action. More specifically, the value function $v^*(s)$ captures the sum of rewards (minus costs) expected by starting at state (e.g., location) s and following an optimal trajectory thereafter. This function contains sufficient information to choose optimally (e.g., by visiting the highest-valued neighboring location at each step). Classic RL algorithms for model-based planning show that solving this interdependent optimization problem is computationally demanding, lacks a closed-form solution, and requires laborious iterative procedures.

The key idea of linear RL is to simplify this laborious iterative procedure by introducing the concept of a default policy, where deviations from this policy incur a control cost. Importantly, optimizing reward minus control cost results in a computationally efficient closed-form solution for the value function^{36,37}. This solution can serve as an approximation of the optimal policy in the original RL problem, which aims to maximize accumulated long-term reward. While this approach is computationally efficient, it can be biased. However, if the default is uniform (i.e., a random walk), it has a negligible effect on the decision policy, and the resulting policy remains a very good approximation of the optimal policy, as it merely encourages stochastic exploration. Throughout this paper, we assume a uniform default policy, represented by matrix \mathbf{T} , which is a square matrix whose size equals the number of states in the environment and encodes transition probabilities between states under a random walk. Similar to SR, linear RL relies on a long-range predictive map $\mathbf{D} = (\text{diag}(\exp(\mathbf{c})) - \mathbf{T})^{-1}$, called the DR. Here, $\text{diag}(\exp(\mathbf{c}))$ is a diagonal matrix whose diagonal values depends on the per-state costs vector \mathbf{c} . The DR in effect captures, for each state, which other states will be visited in the long run following it, and what cumulative per-step costs \mathbf{c} will be expended reaching them. Given \mathbf{D} , it is possible to compute the value function v^* without extensive iteration, but instead by a single matrix-vector multiplication, which is easily implemented in a single-layer neural network. (Here, the vector depends on goal rewards \mathbf{r} , which explains how the system can easily adapt to changes in goals.) A main intuition is that the matrix inversion in the definition of \mathbf{D} arises algebraically from and is equivalent to iteratively summing costs over many transition steps (assuming that the transition is done under the default policy); once this computation is done, planning is easy. Thus, in the present work, we solely focus on the computation of the DR map \mathbf{D} , and in particular on exploiting compositionality to flexibly compute the matrix inversion by reusing precomputed components (see also Bazarjani and Piray³⁸ for an alternative approach that creates the map using a temporal-difference model-free algorithm).

Building the DR compositionally

We present a computational model for making predictive maps that constructs the DR by flexibly putting together representations corresponding to different objects in the environment. Formally, these

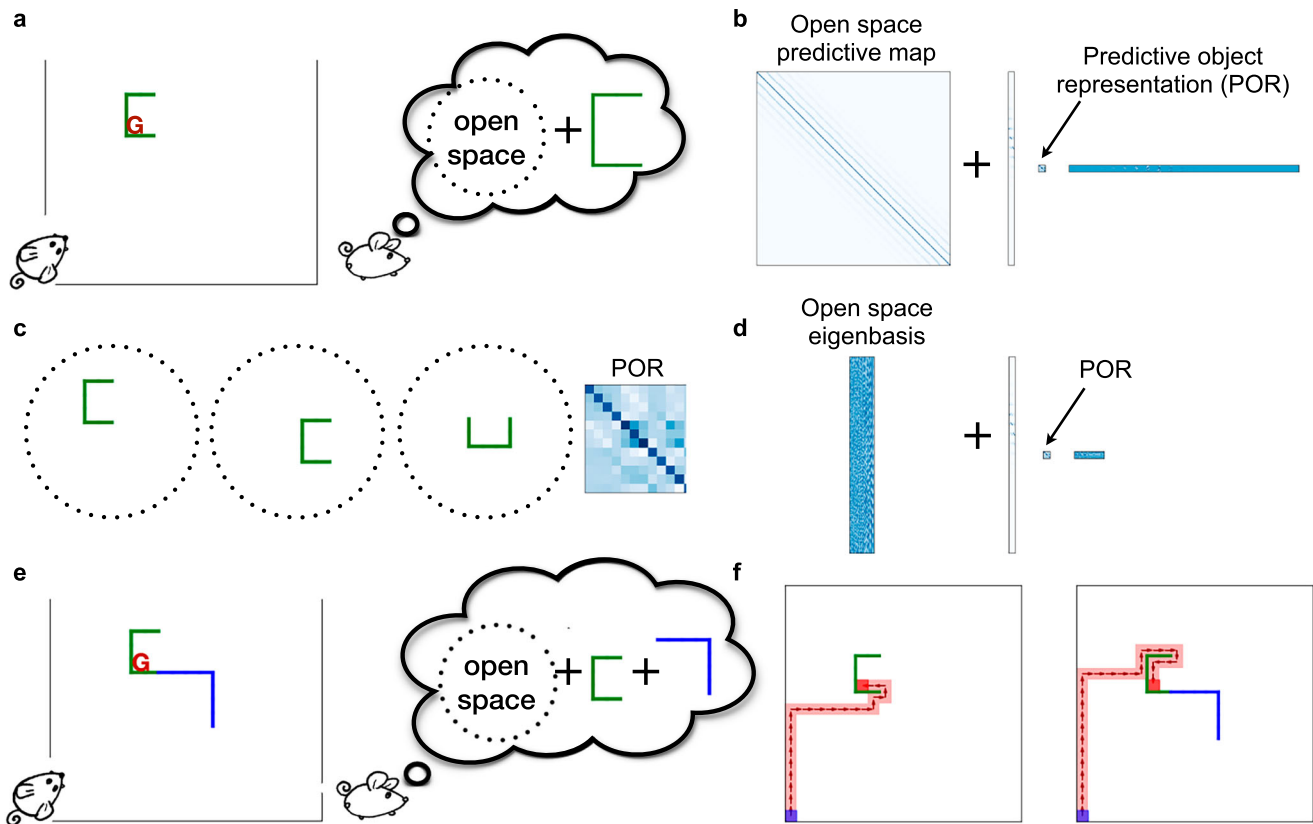


Fig. 1 | Compositional predictive map model. **a** The model creates a predictive map for planning toward the goal (G) by representing each object as a perturbation within the open space field. **b** To compute the predictive map, the model adds a low-rank matrix to the open space predictive map. This low-rank matrix is the product of three components: a column matrix, a row matrix (both serving as readily available lookup tables), and an intermediary matrix called the predictive object representation (POR). The POR’s dimensions depend on the object’s size, making it much smaller than the predictive map. Here, matrices related to the calculation of the predictive map in the environment depicted in (a) are plotted. The matrices involved in calculating the predictive map for the environment shown in (a) are displayed here, with the POR particularly associated with the green object. **c** The POR matrix is invariant to both translation and rotation, as demonstrated for an object in different positions and orientations. **d** This also provides an efficient

method to compress the cognitive map in terms of the eigenbasis of open space by revealing how object effects influence the map when viewed through the open space map’s eigenvector decomposition (see Eq. 2). This is achieved by adding an object-dependent term to the eigenbasis of the open space. This object-dependent term is the product of three components: a column matrix, the POR, and a row matrix representing a linear combination of the eigenbasis in a set of states that are next to the object. **e** A new object is added to the environment shown in (a) to create a new task. The model constructs a compositional predictive map by combining the objects’ PORs, each calculated relative to the open space predictive map, providing a first-order approximation of the new environment. **f** The planned routes in both the original task (a) and the new task (e) demonstrate the model’s effectiveness. When the second object is added, the previously optimal route becomes suboptimal, and the model successfully identifies an appropriate detour.

objects correspond to stereotyped configurations of barriers (e.g., rooms, shapes, or walls) that interrupt travel in the open field (Fig. 1a). In this case, each object can be seen as a perturbation of the open field. Assuming we start by knowing as baseline the DR for the open field, \mathbf{D}_{os} , then matrix algebra (specifically, the Woodbury matrix inversion lemma) indicates that the DR for an environment containing a single object can be efficiently computed by adding a correction term to \mathbf{D}_{os} :

$$\mathbf{D} = \mathbf{D}_{os} + \mathbf{D}_{os} \mathbf{C}_o \mathbf{A}_o \mathbf{R}_o \mathbf{D}_{os}, \quad (1)$$

where \mathbf{D} is the predictive map for the environment in which object o has been introduced, \mathbf{A}_o is the POR matrix whose size depends on the object’s size, \mathbf{C}_o is a binary matrix whose columns encode locations of states immediately adjacent to the object, and \mathbf{R}_o is a matrix that is nonzero only in states that are immediately next to the object. Since the object is typically much smaller than the environment, \mathbf{C}_o and \mathbf{R}_o are column and row matrices (Fig. 1b). These matrices serve as simple lookup tables encoding one-step relations between the object and other states, making them immediately available. Their product with the baseline map \mathbf{D}_{os} yields a linear combination of columns or rows of \mathbf{D}_{os} that are immediately related to the object, a computation that can

be readily implemented in a simple neural network with fixed, pre-initialized weights. Since \mathbf{D}_{os} is a precomputed, reusable matrix that can be computed once (e.g., from random walk in a very large open space; see “Methods”) and used thereafter, the only nontrivial computational term in this equation is the matrix \mathbf{A}_o , which we call the POR for the corresponding object (Fig. 1b). Importantly, the POR’s dimensionality depends on the object’s size rather than the entire state space. Thus, while Eq. 1 may appear notationally complicated, its computation is straightforward and far more efficient than re-inverting the entire matrix from scratch. If S represents the number of states in the environment and O the object size (so $S \gg O$), computing the POR matrix \mathbf{A}_o has complexity $O(O^3)$, while directly calculating the predictive map through naïve matrix inversion has complexity $O(S^3)$.

An important insight is that the POR is both translation- and rotation- invariant, i.e., it does not change if one moves or rotates the object in the state space (Fig. 1c): the effect of any object on trajectories is clearly invariant up to the frame of reference, and the \mathbf{C}_o and \mathbf{R}_o matrices align the object in the broader map.

It is also important to stress what this procedure fundamentally accomplishes. In classic model-based RL, it would of course be straightforward to derive the one-step transition map \mathbf{T} compositionally

by directly adding a set of barriers to an open grid. However, following any such update, it would be necessary to recompute the optimal values by a full matrix inversion (to produce \mathbf{D}), or alternatively, summing over a large number of trajectories. Here, instead, we compositionally build the long-run predictive map \mathbf{D} directly, reusing most of the computation underlying the matrix inversion via \mathbf{D}_{os} and \mathbf{A}_o (the learning of which we address below). Updated optimal action values (and the optimal policy) can then immediately be produced with a matrix-vector multiplication.

Efficient representation of the DR map using open space eigenbasis

Another crucial aspect of the model is its representation of the DR map in terms of the open space map's eigenvectors. If we consider the eigenbasis of the open space map $\mathbf{D}_{os} = \mathbf{U}_{os}\mathbf{U}_{os}'$ (this decomposition exists because \mathbf{D}_{os} is symmetric), Eq. 1 can be written as $\mathbf{D} = \mathbf{G}\mathbf{U}_{os}'$, in which we have defined \mathbf{G} as

$$\mathbf{G} = \mathbf{U}_{os} + \mathbf{D}_{os}\mathbf{C}_o\mathbf{A}_o\mathbf{R}_o\mathbf{U}_{os}. \tag{2}$$

This formulation reveals how an object effects (captured by the POR matrix \mathbf{A}_o) influence the map when viewed through the open space map's eigenvector decomposition (see also Fig. 1d). Since the open space eigenvectors are fixed, this provides a computationally efficient encoding of features in the new space. Moreover, because columns of \mathbf{U}_{os} associated with larger eigenvalues can serve as a low-dimensional representation of the open space map, we can efficiently compress \mathbf{G} by using only a subset of \mathbf{U}_{os} vectors, which serve as a set of basis functions. Notably, these basis functions resemble MEC grid fields, providing insights into various aspects of grid fields that we will explore later.

Compositional representation of the DR map for multiple objects

To construct the DR for environments with multiple objects compositionally, we can, to a first approximation, simply sum (projected) PORs from multiple objects. This is because if two or more objects are sufficiently far apart, their combined POR is approximately equal to the block-concatenation of their corresponding PORs. Thus, formally, if \mathbf{A}_1 and \mathbf{A}_2 are PORs of such two objects, the predictive DR map is approximately given by:

$$\mathbf{D} \simeq \mathbf{D}_{os} + \mathbf{D}_{os}(\mathbf{C}_1\mathbf{A}_1\mathbf{R}_1 + \mathbf{C}_2\mathbf{A}_2\mathbf{R}_2)\mathbf{D}_{os}, \tag{3}$$

where \mathbf{C}_1 and \mathbf{R}_1 are lookup matrices associated with the first object, and \mathbf{C}_2 and \mathbf{R}_2 with the second object, as outlined above. In this way, a set of PORs can be viewed as a set of basis functions for predictive maps reflecting common object motifs in a set of environments (Fig. 1de). This compositional formulation not only offers an often accurate and efficient approximation to the predictive map (Fig. 1f) but also straightforwardly provides the foundation for a simple learning procedure (not elaborated here) for exploring a new environment and quickly estimating its corresponding map \mathbf{D} , while also efficiently planning at each step of learning.

Note that the convenient compositional expression from Eq. 3 is simplified with respect to the exact DR. The latter would be more laboriously obtained by the iterative application of Eq. 1, such that the perturbed \mathbf{D} replaces \mathbf{D}_{os} when adding \mathbf{A}_2 and so on, and thus the object effects do not decompose into a simple sum. The intuition for the difference (Fig. 2a, b) is that when two objects are close (and especially when they touch), their effects on possible trajectories through the environment may no longer be independent. Nevertheless,

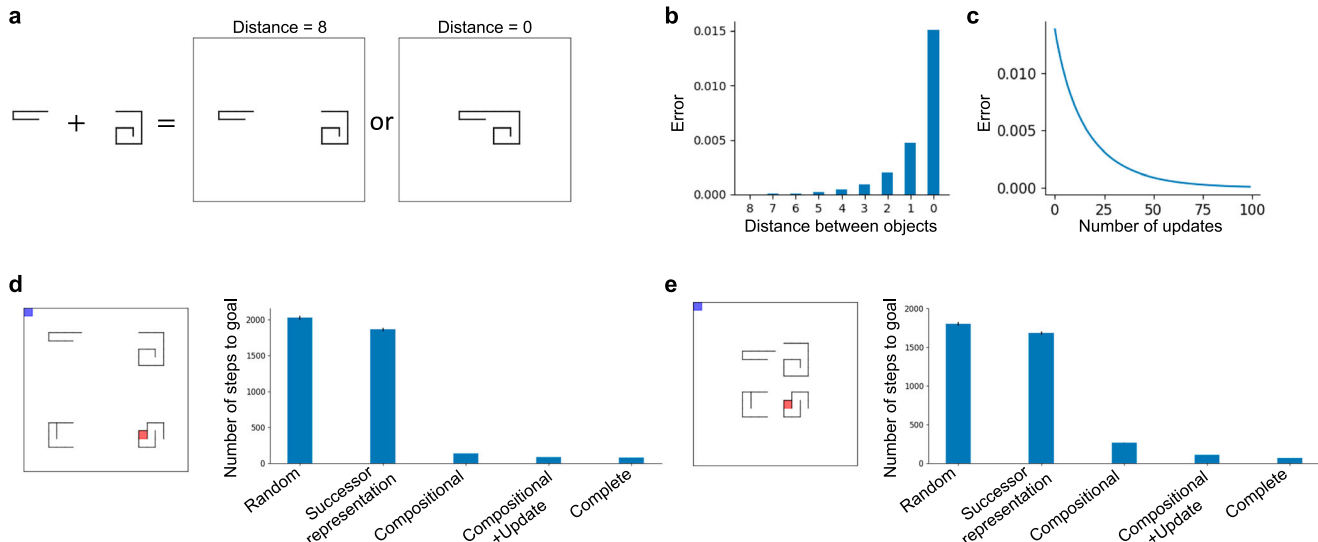


Fig. 2 | The model enables efficient planning. **a, b** The compositional predictive map is a first-order approximation of the complete predictive map. The error between this approximation and the true predictive object representation matrix is plotted as a function of the distance between objects. The error increases as the objects get closer together. **c** The error function is plotted for the learning algorithm that learns the true predictive object representation matrix for the environment with the two objects plotted in (a) (the one with Distance = 0). With each update, the error decreases. The starting point of the algorithm is given by the compositional representation of the corresponding POR matrices associated with the two objects (i.e., Eq. 3). **d** An example planning environment, in which blue and red squares indicate start and terminal points, respectively. We tested the planning performance of several approaches. The predictive map for this new environment was constructed using: the Compositional model; the “Compositional+Update”

model, which starts from the compositional map and updates the map once per step using the algorithm introduced above; and a “Complete” model, which builds the predictive map entirely by continuing the update process until convergence. We also simulated the Successor Representation model, which creates predictive maps using a temporal-difference algorithm and a random walk algorithm (referred to as “Random”). Overall, both the Compositional and Compositional+Update models greatly outperformed the random walk and the Successor Representation and performed on par with the Complete model. **e** A more challenging environment in which objects are closer to each other. As expected, the Compositional model performed worse than the previous simulation, although its performance remained comparable to the Complete model. Simulations in (d, e) were repeated 5000 times, and error bars (might not be visible) represent the standard error of the mean.

even when objects are close, this equation provides a principled first-order approximation of the POR of both objects (Fig. 2).

Moreover, we have developed an efficient and biologically plausible algorithm (see “Methods”) to iteratively update the combined POR for two or more objects and learn it gradually (either by experience or mental replay or both), correcting the initial approximation. The same algorithm is applicable to the case of learning a single POR like \mathbf{A}_1 from more elemental parts (corresponding ultimately to individual barriers). This algorithm is computationally efficient because it works in the space of objects (i.e., the part of the space containing the subcomponents: the size of the combined POR), rather than the whole state space (the size of \mathbf{D}). This is crucial because previous algorithms for learning predictive maps by experience, notably the SR^{39–42}, were not scalable, as they required updating a row vector the size of the entire state space at each iteration. In practice, the proposed algorithm converges quickly, usually within a few trials, which is mainly because Eq. 3 already provides a good approximation of the combined POR (Fig. 2c).

In all, this learning procedure describes how a set of PORs can be built up for stereotyped objects like rooms and, in turn, how the combined PORs from multiple objects close to one another in a new environment can, with learning, be adjusted to a single corrected POR for their combination. This could, in turn, provide the basis for a meta-learning algorithm (which we do not pursue here) that learns an appropriate basis set of PORs matched to a set of object motifs reoccurring across a family of environments.

Planning with a compositional DR

This approach enables efficient planning. To demonstrate this, we conducted a simulation analysis in which different models were exposed to four different environments, each with a distinct object (Fig. 2d, e). This allowed the model to learn four PORs. Subsequently, models were simulated in a new environment containing all four objects. For comparison, we considered the random walk as well as the SR, an alternative way to construct predictive maps experientially (non-compositionally), based on a temporal-difference learning rule^{32,39,42}. We built the predictive map in two different ways and compared its planning capabilities with these baseline models, as well as a complete predictive map.

The first approach involved a simple first-order approximation model, as described above, where PORs for all objects were simply added as in Eq. 3. We used this map with the linear RL algorithm, enabling us to plan a route from a starting point to a terminal point. The second approach additionally updated the combined POR for the four objects using the learning algorithm mentioned above. We assumed that the time required for an update step was equal to the time needed for taking a real navigational step in the world, although it's worth noting that the updates can often be much faster.

As depicted in Fig. 2d, even the first model, which was simply built by a linear composition of individual PORs (i.e., Eq. 3), performed relatively well. This is important because implementing this algorithm is simple and straightforward. Unlike a random walk, which rarely reaches the target in a reasonable time, this algorithm navigates to the target effectively (with an average path length compared to the ground-truth planner of 1.76). Moreover, even with a slow update rate (i.e., one update per step), this algorithm achieved accuracy close to the ground-truth planner (with a relative average path length of 1.09). Importantly, as described above, the update is in the space of objects, not the whole state space, which makes this algorithm much more efficient than previous algorithms, such as the SR, for creating a predictive map. Additionally, it is worth noting that the ground-truth model can be seen as an asymptotic case of the update model, after updates have been performed until convergence.

We repeated this analysis under more challenging conditions, where the objects were placed closer to each other. As expected, the

performance of both algorithms, especially the Initial model, declined, but remained relatively comparable to the complete planner and substantially better than the random walk and the SR (Fig. 2e).

Neural representations of map components in the MEC: object vector cells

A number of previous modeling studies have suggested that spatially tuned neurons in the MEC may in some way collectively represent a cognitive map. For instance, Stachenfeld et al. suggested that grid cells of MEC collectively resemble a factorization of the environment's SR, resembling its eigenvectors³². A second recent proposal, known as the Tolman-Eichenbaum Machine²⁷ develops a related proposal more implicitly by training a network, via backpropagation, to learn motifs of graph connectivity across many environments (similar, in our terms, to meta-learning the one-step map \mathbf{T}). Units inside the trained network have responses resembling those of several types of spatially tuned MEC neurons. The present model elaborates these accounts and, through more explicit analysis of the mapping and planning problem, clarifies the computational mechanisms that, we suggest, underly these observed correspondences.

Recent research in the MEC reports a novel family of neurons known as object vector cells³⁴. These neurons activate when an animal is at a certain distance and direction from an object. It has been observed that 20–30% of the neurons in the MEC are object vector cells, which makes them more abundant than grid cells and border cells.

We suggest these cells can be understood as capturing the perturbation of the map due to the inclusion of the object, i.e., the POR term from Eq. 1. Perhaps the most significant characteristic of object vector cells is their translation invariance. In a study by Høydal et al.³⁴, MEC cells' response fields were examined in two consecutive trials. An object was initially placed on the floor and then moved to a new location. The researchers observed that object vector cells that fired initially exhibited a similar response even after the object's translation. Importantly, the displacement from the object was preserved. Similarly, the model, i.e., the projection of the POR into the state space, demonstrated a consistent response across both trials due to its translation invariance (Fig. 3). Furthermore, Høydal et al. observed that MEC object vector cells exhibit a consistent response to an object when the animal is at a similar distance and orientation relative to the object. The model displays a similar behavior, as the PORs are rotationally invariant (Fig. 3).

Note that we model object vector cells as a row vector of the (projected) POR corresponding to the specific state associated with the particular distance and direction from the object to which the cell is tuned. Generally, this formulation provides a way to account for the activity of “vector” cells in the MEC⁴³, such as object vector cells or boundary vector cells, within the RL framework that assigns a unique state to each spatial location. Vector cells in the MEC exhibit selective firing patterns exclusively when the animal is positioned at a particular distance and direction from the cell's focal point, which could be an object or a boundary in the environment. Within the RL framework, this firing behavior is effectively equivalent to the activity pattern corresponding to the state that shares the same directional and distance relationship with the object or boundary as the animal's current location. For example, an object vector cell tuned to “north of object” corresponds to the state representing that particular spatial relationship. Object vector cells serve as an intermediary step in the process of constructing the final representation of space in the form of a grid map (which we cover later), and their object-relative code facilitates and reflects the compositional construction of the final grid map.

Another property of object vector cells is their ability to generalize across different objects and environments. Unlike landmark vector cells in hippocampal area⁴⁴, object vector cells do not differentiate between different objects as long as they introduce similar disruption to the state space. The model also shows the same behavior.

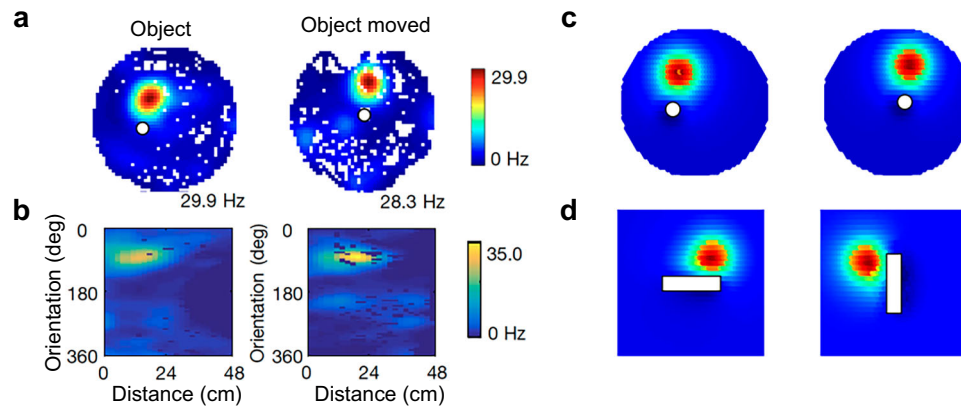


Fig. 3 | Object representations are invariant to distance and orientation relative to the object. **a** Firing rate maps of an example object vector cell. When the object was moved, the firing field was moved accordingly. **b** The vector map for the cell

shown in (a), as a function of allocentric orientation and distance from the center of the object. Model simulations show similar properties when the object was moved (c) or rotated (d). Data in (a, b) are adapted from Høydal et al., Springer Nature³⁴.

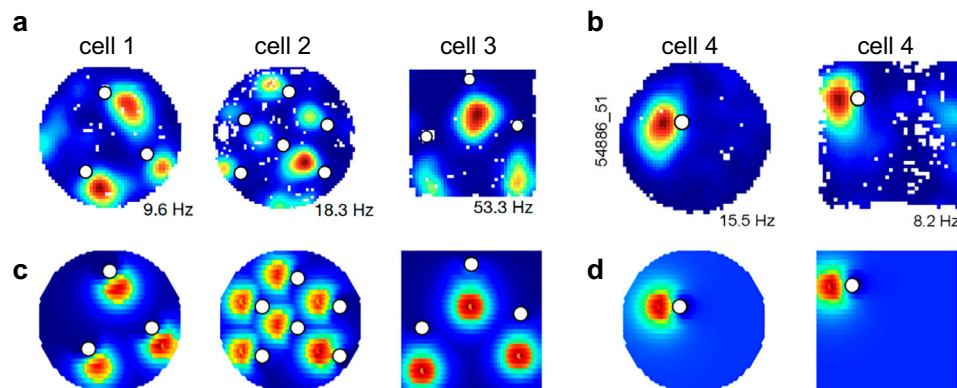


Fig. 4 | Object representations generalize to multi-object environments, and to novel objects and novel environments. **a** The firing field for three object vector cells in a multi-object environment in which a few new or old objects (with similar sizes) have been placed in the space. Regardless of familiarity, the cells exhibit similar firing fields in the vicinity of each object. **b** The firing field for an object

vector cell in two different environments, which shows that the cell generalizes between environments. **c, d** Simulation of the model in similar setups. The model shows the same behavior because PORs are only a function of the object size, not its shape or familiarity, and they are not learning or experience-dependent. Data in (a, b) are adapted from Høydal et al., Springer Nature³⁴.

Since PORs capture perturbations of trajectories through the state space by the object, not its specific properties, they remain the same as long as the way that the object impedes passage remains the same. Furthermore, empirical data showed that object vector cells, unlike landmark vector cells, do not depend on experience and emerge almost from the first trial, even in a novel environment (Fig. 4). The model shows the same behavior because PORs are not learning or experience-dependent (unlike predictions of the model for grid cells as shown in the next section.) Furthermore, when multiple objects are introduced in the field, object vector cells fire in response to each with small differences in distance and orientation (Fig. 4). The model accounts for this because PORs do not encode interactions between objects, and therefore we expect them to generalize to multi-object environments.

In their study, Høydal et al. found that object length has a significant effect on the firing field of object vector cells, although some variability was observed (Fig. 5). Our model exhibits similar behavior, because the length of an object affects the way it alters state transitions (Fig. 5). However, the model predicts that increasing the height of an object should not affect the state space since the height does not impact how the object affects trajectories (apart from climbing or jumping). This is also in line with the empirical results reported by Høydal et al.

A final observation, perhaps less obvious under the current model, is that object vector cells fire even when the object is suspended, allowing passage below³⁴. In the model, this may be because PORs in our model encode computational representations that are precomputed and could be added to the predictive map if needed. In the case of objects that do not actually block passage, these representations would not be incorporated in the predictive map, but they can still be precomputed and represented as a computational basis tied to any object that has the potential to serve as a barrier when placed differently.

Neural representation of the compositionally constructed map: grid cells

It has been proposed that grid cells encode a low-dimensional decomposition of the predictive map³². Specifically, in an influential study, Stachenfeld and coauthors proposed that grid cells encode eigenvectors of a different variant of a predictive map, the SR: collectively, these neurons' response fields capture the map, in effect, as a set of basis functions for it. While this proposal provides a promising characterization of grid cell responses in the open field—particularly when the eigendecomposition is subject to a biologically motivated non-negativity constraint, which improves the match to neural responses (Dordek et al.⁴⁵)—a number of puzzles remain, particularly

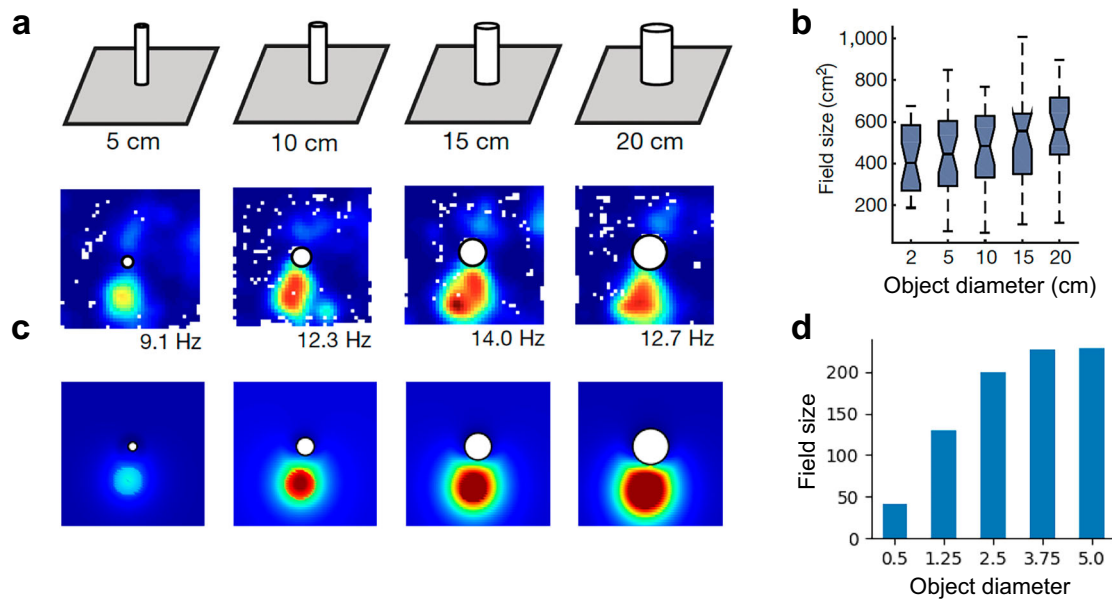


Fig. 5 | Object representations depend on the object size. a The firing rate for an object vector cell illustrates dependency on the object diameter. **b** Distribution of field sizes for all object vector cells shows significant dependency to the object

diameter. **c, d** Simulation of experiment results in **(a, b)**. Data in **(a, b)** are adapted from Høydal et al., Springer Nature³⁴.

concerning how the grid code behaves in the presence of barriers. In general, the addition of barriers to the map requires laboriously re-computing the SR (this is the problem addressed by the current model) and its eigendecomposition, and this in turn has substantial effects on all the eigenvectors (i.e., each cell's simulated tuning). Indeed, even allowing for such recomputation, this model fails to capture the relatively stable code of grid cells when objects and goals are introduced, as eigenvectors tend to change dramatically when the map changes even slightly.

Here, we leverage the current model to propose that grid cells provide a basis space for representation of the DR, which is expanded compositionally to include perturbations due to the PORs as described above. Importantly, the resulting grid code remains globally stable because these perturbations are represented in a space given by the eigenvectors of the open field.

In particular, drawing on recent work^{32,45,46}, we consider a baseline model given by the eigenvectors of the DR from the open space, given by the columns of a matrix \mathbf{U}_{os} . These are computed by non-negative principal component analysis of \mathbf{D}_{os} , using the model of ref. 45, who showed that such components can be learned through a single-layer neural network, with a Hebbian-like learning rule⁴⁷. This procedure produces hexagonal grid patterns (Supplementary Fig. 1). We chose the Dordek et al.⁴⁵ framework for generating open space grid representations because it is based on a random walk, resulting in slightly less geometrically regular but more biologically realistic patterns, compared to the eigenvectors of the SR. This aligns at least stylistically with experimental observations, where hexagonal grid patterns can be imperfect. However, our simulations do not depend on this specific choice of basis functions. Similar results can be achieved using alternative approaches, including the more regular grid patterns proposed by Stachenfeld et al.³², demonstrating that our core findings are robust across different representations of open space.

We then introduce perturbations from PORs as before (presumably arising from object vector cells), projected onto the basis \mathbf{U}_{os} . Formally, we propose that grid fields encode matrix \mathbf{G} presented in Eq. 2. As we stated there, \mathbf{G} essentially provides a representation of the DR \mathbf{D} , as the full predictive map can be easily assembled via the product of \mathbf{G} with the transpose of \mathbf{U}_{os} . Moreover, \mathbf{G} can be compositionally expanded to incorporate additional objects, via Eqs. 2 and 3.

Furthermore, the grid code \mathbf{G} equals the baseline grid code \mathbf{U}_{os} (and its product with itself represents the baseline, open field map \mathbf{D}_{os}) when no objects are present in the environment.

Here, we present simulation results examining how different types of barriers and goals in the open space impact the baseline grid code. Thus, we examine the effects on \mathbf{G} when task-relevant objects are introduced.

A key prediction by our model is that the introduction of barriers and objects should not result in a global remap of grid cells. Instead, the model predicts that such changes in the environment result in local remapping of grid cells. This is because such barriers and objects mainly influence the grid codes near the area of change, through the object-specific lookup table matrices, \mathbf{C}_o and \mathbf{R}_o . This contrasts with the model by Stachenfeld et al., which predicts substantial global grid remapping since barriers alter all eigenvectors³².

To showcase this, we first consider an experiment by Wernle et al., which examined grid cell changes when environments merge⁴⁸. They recorded grid cells in rats exploring two adjacent rectangular compartments, A and B, separated by a dividing wall. When the partition was removed, allowing exploration of the merged compartments, the researchers found grid patterns were largely preserved near the original distal walls (Fig. 6). However, approaching the former partition line, individual fields “almost immediately” reorganized to establish spatial continuity between the two maps⁴⁸. Specifically, grid periodicity emerged locally around the area where the compartments had met. Thus, their results demonstrate that when merging occurs, grid cells quickly reconfigure to reconcile the separate maps, enabling coherent navigation in the new unified space. The model predicts a similar pattern because the partition wall primarily influences the area near the partition, due to the relative selectivity coded by object-specific lookup table matrices, \mathbf{C}_o and \mathbf{R}_o .

This property of grid fields was also recently highlighted in another study examining the effects on grid fields of introducing the rat's home cage into an environment⁴⁹. In this study, the authors found that grid cells did not globally remap when the home cage was placed in the arena. Instead, they showed that the introduction of the home cage in the center of the arena retained the global representation but resulted in a local shift of grid fields near the embedded home (Fig. 7). Thus, the mean normalized rate was selectively increased near the

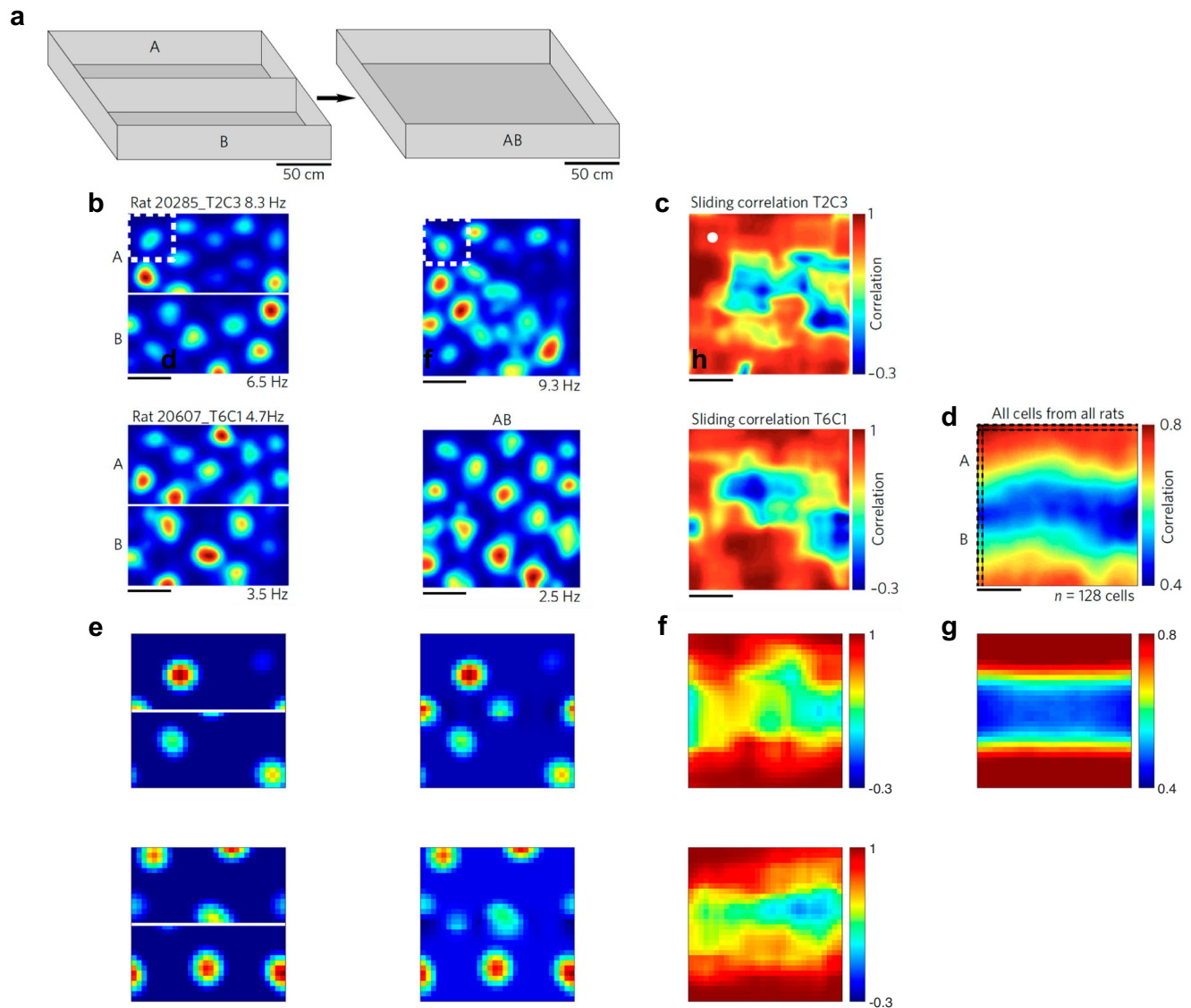


Fig. 6 | Grid maps change locally after removing a wall. **a** The experimental procedure of Wernle et al. is illustrated. The rat underwent training in two rectangular compartments, labeled A and B. These compartments were initially separated by a central partition. During the test phase, the partition was removed, allowing the rats to explore the combined open square environment, denoted as AB. **b** Example firing rate maps for two grid cells from different rats are shown in both the divided A/B and merged AB environments. **c** The sliding spatial correlation heatmap is shown for the two example cells plotted in **(b)**. Spatial correlation is

calculated in moving boxes with the size of the white stippled box shown in **(b)**. **d** The average spatial correlation heatmap across all recorded cells is shown. The impact of removing the partition is mostly limited to locations near the partition. **e–g** Simulation of experimental results in **(b–d)** by the model. The simulation results in **(g)** were repeated 50 times with different randomization seeds, and the average has been plotted. See also Supplementary Fig. 2. Data in **(b–d)** are adapted from Wernle et al., Springer Nature⁴⁸.

home cage, which is also visible in spatially averaging peak normalized rate maps. Further analysis showed that local differences were primarily influenced by the home geometry rather than the specific behavioral properties typically associated with a home cage. Specifically, the introduction of a plain box with similar geometrical properties to the home cage resulted in very similar changes in grid fields, even though the elicited behaviors were quite different. The model predicts the same behavior because it assumes that the effect of the home cage on predictive maps is primarily due to the way it alters transitions in relation to an open field, and because the DR (unlike the SR) codes predictive relationships in a way that is invariant to the expressed behavioral policy.

Of course, in environments where barriers alter almost the entire space relative to open areas, we expect more global remapping. For example, Derdikman et al. recorded grid cells in a hairpin maze

environment, and the grid map was substantially distorted. Notably, a significant number of grid cells showed opposite patterns for alternating arms⁵⁰ suggesting a repetitive, compositional construction. The model predicts the same behavior, as the extent to which the barriers changed the open space is quite extensive here (Fig. 8), and reflects the repetitive, compositional structure of the perturbation. Moreover, Derdikman et al. found that grid cells “reset” at turning points, a finding that is not consistent with classical global space coding theory of grid cells, but it is consistent with the theory that grid cells encode a cognitive map that predicts “fragmentation” in the predictive map. Finally, as we pointed out in our previous study³⁰, unlike the eigenvectors of the SR, experimental data support an implementation of predictive maps that is influenced by actual physical barriers but, in contrast, is relatively stable with respect to stereotypic behavioral policies. For instance, Derdikman et al. also tested whether grid cells

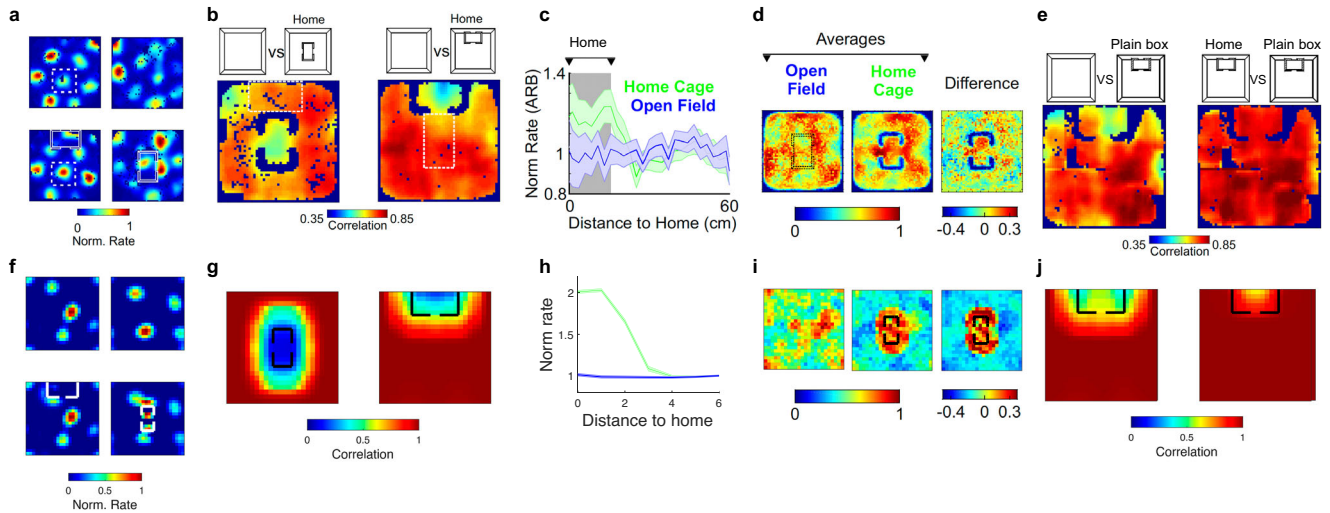


Fig. 7 | Grid maps change locally near the home cage. **a** In the experiment by Sanguinetti-Scheck and Brecht⁴⁹, grid cells are recorded before and after the introduction of a home cage. Here, two example normalized grid fields are plotted for two different home cage locations (north or middle). **b** Sliding window spatial correlation map averaged across all cells is plotted for two different locations of the home cage. The impact of introducing the home cage is mostly limited to locations near the cage. **c** The peak normalized firing map for all cells is plotted as a function of the distance between each cell's peak location and the center of the home cage. The local impact of the home cage on grid maps is also evident in the peak normalized rate map. The mean and standard error of the mean are plotted. **d** The spatial average of the peak normalized rates is illustrated for both the open field

and the home center condition, with the difference also displayed. The same effect is apparent in this comparison. **e** Sliding window spatial correlation map averaged across all cells is plotted for another experimental condition in which a plain box is introduced with the same size as the home cage. The effect of the plain box is very similar to that of the home cage, revealing that changes in grid cells are driven by the geometry rather than the home valence. **f–j** Simulation of experimental results in (**a–e**) by the model. The simulation results in (**f–j**) were repeated 50 times with different randomization seeds, and the average has been plotted. Error bars in (**h**) indicate the standard error of the mean across simulations. See also Supplementary Fig. 3. Data in (**a–e**) are adapted from Sanguinetti-Scheck and Brecht⁴⁹ under a CC BY license: <https://creativecommons.org/licenses/by/4.0/>.

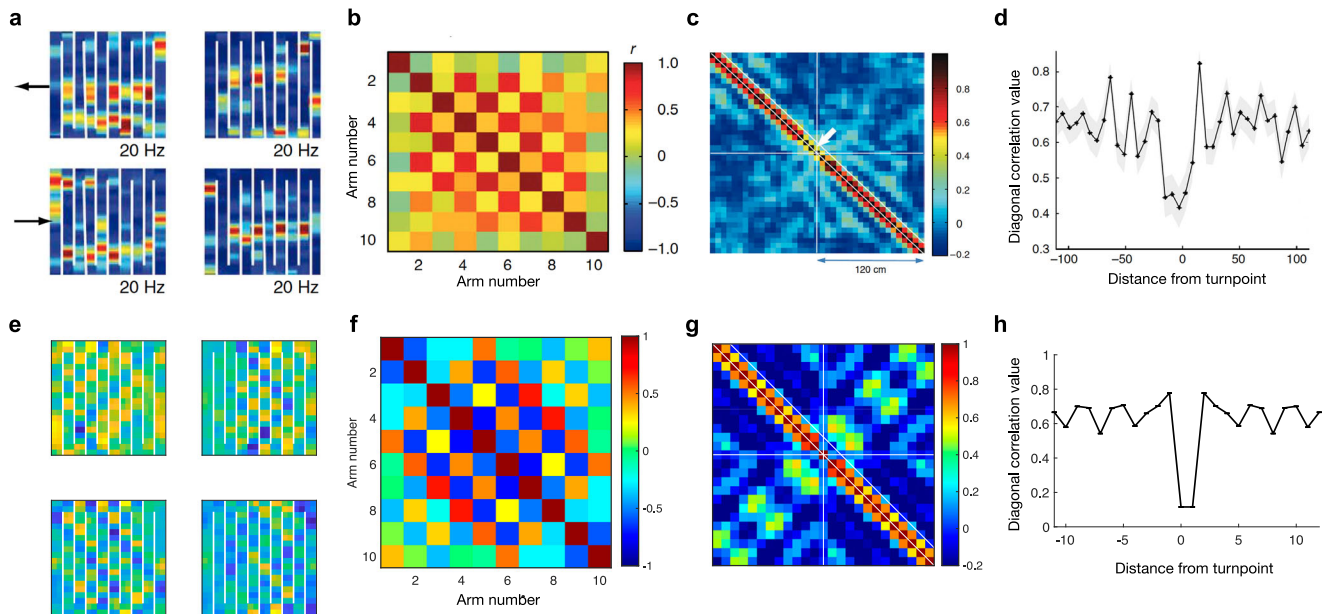


Fig. 8 | Predictive representations dramatically remap if an extensive barrier is introduced in the environment. **a** Firing rate of two example grid incells recorded in the hairpin-like environment shows an alternating firing pattern between different arms. **b** Correlation between different arms across all recorded grid cells. **c** Spatial correlation map is plotted, which is calculated based on population vectors of firing rates at the turning point and in 6-cm bins on both sides of the turning point. Directionally tuned grid cells were excluded from the population vectors.

d Correlation derived from the adjacent position bins selected from the diagonal section labeled in (**a**) is plotted. The mean and standard error of the mean are plotted. **e–h** Simulation of experimental results in (**a–d**) by the model. The simulation results in (**e–h**) were repeated 50 times with different randomization seeds, and the average has been plotted. Error bars in (**h**) indicate standard error across simulations. See also Supplementary Fig. 4. Data in (**a–d**) are adapted from Derdikman et al., Springer Nature⁵⁰.

are affected when rats are trained to run an equivalent hairpin pattern without any actual barriers⁵⁰. They found that grid fields in this case are more similar to the open field grids than the ones observed in the actual hairpin maze environment, a finding that is well explained by

our model (again, due to the off-policy character of the DR), but not by policy-dependent predictive map models³².

Additionally, Derdikman et al.⁵⁰ observed that truncating some arms of the maze caused grid cell firing patterns to remain primarily

anchored to distances from previous turns, rather than being substantially altered by the new endpoints. This has been interpreted as evidence for a path-integration mechanism. Our model predicts that such truncations should affect grid fields locally, particularly around the newly introduced endpoints, while other predictive map models would suggest broader changes in their eigenvectors across the environment. However, due to the lack of direct experimental comparisons before and after truncation, it remains difficult to assess which framework better captures this effect. More generally, this underscores the need to reconcile predictive map models, including ours, with path-integration-based explanations of grid cell activity, particularly in environments where structured barriers introduce distinct movement constraints.

A final point about grid cells is their sensitivity to changing goals⁵¹. Although it had been thought that grid fields are stable relative to the animal's goal, recent studies suggest that goals subtly modulate grid fields^{52,53}. It is not clear how this can be explained by previous models, even by predictive accounts, particularly the nuanced and subtle changes observed in grid maps. This is because the SR and its eigenvectors tend to change dramatically when a new goal is introduced, as this fundamentally alters the policy. Interestingly, though, goals in the current model do have a special status, which is similar to barriers, because the model concerns the so-called “episodic” RL setting, in which episodes terminate (and predictions are truncated; the next episode begins) whenever a goal is reached. Thus, our model distinguishes “terminal” states (containing goals) from other states, and if a state becomes a goal (terminal), this changes the transition probability between the goal state and its neighbors in an open field. Intuitively speaking, the goal state is similar to a one-way barrier because as soon as the agent gets to the goal, it does not move back as the planning episode is completed. Thus, we can similarly formalize the effect of the “goal component” on the predictive map. This aspect of the model accounts for recent studies that investigated the role of goals on grid fields^{31,52,53}. For example, Boccarda et al. tested how grid fields change

when rats are trained to learn three new reward locations daily on a cheeseboard maze⁵². They found a local remapping in grid cells after learning, in which grid fields mainly changed near the goal locations, with fields getting closer to the goal location. This also led to a reduction in the hexagonal grid pattern, consistent with a distortion of grid fields. The model shows the same behavior (Fig. 9).

Discussion

Compositionality is an important aspect of a flexible system, whether biological or artificial. Compositionally, thus seems particularly relevant to the concept of cognitive map, which is supposed to organize knowledge in a way that is generalizable across tasks and environments and enables flexible behavior, such as planning routes or taking novel shortcuts²⁰. Neurons within MEC, such as grid cells and object vector cells, have been proposed as the neural substrates of the cognitive map^{16,17,19,54}, and have a number of properties that seem to suggest they represent different separable components of such a map. However, it has remained elusive how such a cognitive map could support flexible planning while being created compositionally. This includes both traditional spatial metric theories of MEC grid cells⁵⁵, as well as newer models in which MEC represents an abstract state space^{16,27,56} or a predictive SR^{30,32}. While some of these models emphasize one or the other of these features, none of them reconciles the cognitive map's utility for planning with the need for it to be built compositionally from task elements, such as objects, barriers, and goals. Here, we have proposed a predictive model for planning that is also created compositionally by assembling representations related to objects and goals. This also gives a clear and explicit computational interpretation to the properties of different spatially tuned neurons in MEC.

The new model is based on our linear RL theory, which posits that, given a long-range predictive map encoding closeness between all states under a random policy, optimal planning can be accurately and efficiently approximated for any goal³⁰. Moreover, our approach leverages four key computational features: (1) Objects, barriers, and

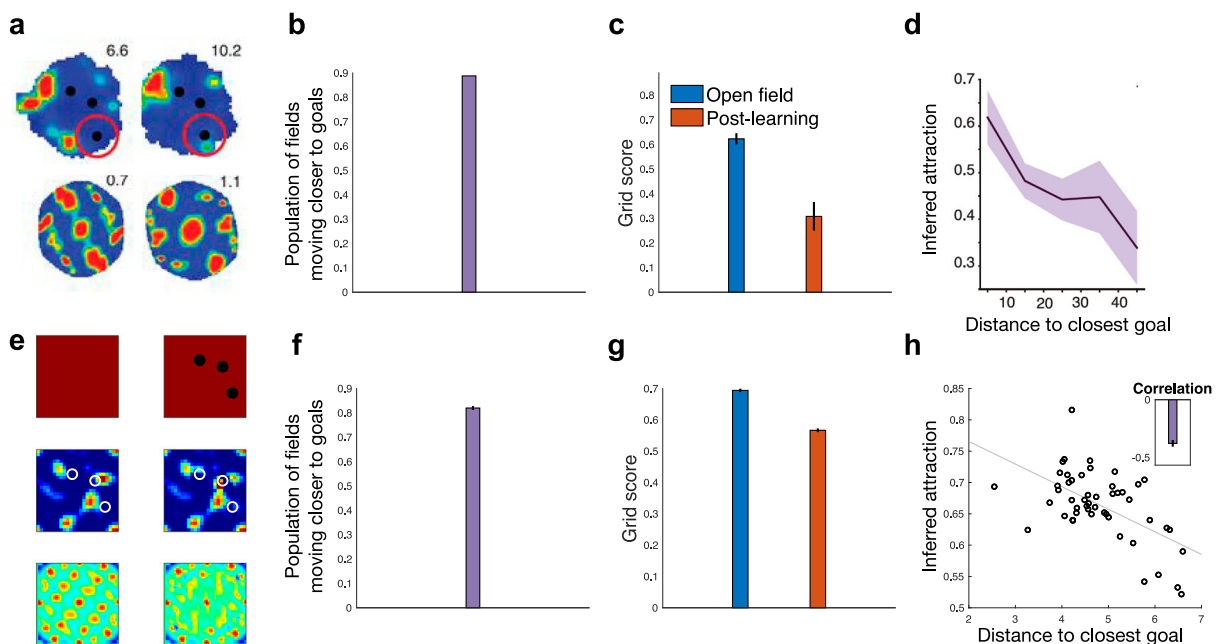


Fig. 9 | Introduction of goals locally modulates grid fields. a Firing rate of an example grid cell is shown before and after learning the goal locations (black dots). **b** Introduction of goals led to grid cells moving closer to goals. **c** Grid score for the open field and after learning the goal locations is plotted. The mean and standard error of the mean are plotted. **d** The degree of attraction of grid fields is inversely related to the distance of their peak from the closest goal. The mean and standard

deviation are plotted. Simulation of experimental results in (a–d) by the model. The simulation results in (e–h) were repeated 50 times with different randomization seeds, and the average has been plotted. Error bars indicate standard error across simulations. See also Supplementary Fig. 5. Data in (a–d) are reproduced/modified from Boccarda et al., <https://doi.org/10.1126/science.aav4837>, AAAS⁵².

goals can be viewed as perturbations disrupting the relational structure of open space. (2) Their effects on long-range predictive maps can be efficiently computed using translation- and rotation-invariant matrices. These are efficient as their size depends on the object, not the full state space. (3) Long-range maps can be efficiently constructed by combining these compact PORs. (4) There exists an efficient set of basis functions for representing the cognitive map by combining PORs with the eigenvectors of the open space.

Our computational approach lays a substantially improved foundation for understanding the function of different cells within the MEC, including object vector cells or border cells encoding elements of the task space, as well as grid cells providing a basis function of the compositional predictive map. In particular, we showed that the proposed code for representing objects resembles remarkable similarity to object vector cells, an important class of cells discovered recently in the MEC³⁴. Similar to object vector cells, PORs are translation- and rotation-invariant, and their field size depends on the object size. The model further provides a computational mechanism of how grid cells might reflect basis functions for cognitive maps. In fact, by combining PORs with eigenvectors of the open space, one can obtain a low-dimensional representation of the proposed compositional predictive maps, a process that can be done efficiently using a simple neural network with known weights. This makes a critical prediction, particularly about the effects of barriers and goals on grid fields. Specifically, we anticipate that the effects of barriers and goals on grid fields remain relatively local, a prediction that is well supported by empirical data⁵⁷.

Our theory relates to earlier work by Stachenfeld et al., proposing that grid cells encode eigenvectors of the SR³². The SR constitutes a predictive map of task space and can be seen as a policy-dependent equivalent to the DR utilized here^{39,42}. In many ways, our work builds upon this idea. However, the Stachenfeld et al. model faces major computational and empirical challenges. Computationally, it is unclear how such a code can be implemented in an efficient way, given that all eigenvectors must be recomputed with any change to goals or environment structure. Notably, the SR's size is quadratic in the number of states in the environment—thus computing its eigenvectors even once is often intractable in a large state space, let alone repeated full computation of eigenvectors with every change. The SR also makes predictions about the neural code that are contradicted by data, in that eigenvectors of the SR would exhibit extensive remapping when barriers, objects, or goals are introduced. This contrasts with experimental data showing local grid remapping in such situations⁵⁷, especially in naturalistic environments^{49,52,53,58,59}.

Our new theory also relates to a recent but growing body of work proposing probabilistic, graph-like computational models of MEC cognitive maps, which argue that MEC encodes relational structure between different states in the world^{16,27,60}. The DR framework naturally aligns with these models because it shares key mathematical properties with the transition matrix (T), including the same eigenvectors, which allow it to inherit T's graph-like structure. More specifically, DR represents changes in reachability between states due to objects or barriers, meaning it captures relational structure in terms of how these elements modify the topology of navigation. This connection implies that DR implicitly encodes the same type of graph-like relationships that are fundamental to these cognitive map theories. Our model extends this line of work in two critical ways. First, these previous models are often agnostic to planning, because they do not explicitly model reward as an entity that organisms seek to maximize. Indeed, because they generally represent the one-step map, building planning over this representation is computationally quite laborious. Our model creates cognitive maps that, together with the linear RL framework, provide an efficient method for planning. Second, while one of these models²⁷ does capture a sort of compositionality, in that it is based on learning to generalize relational graph connectivity motifs across different environments, this property emerges only implicitly and somewhat opaquely by training a black-box

neural network via backpropagation. This, in turn, limits the functional interpretation of units within the simulated networks, which are observed to resemble those in MEC. Our current model additionally provides an explicit, algebraically derived method for constructing maps compositionally from individual components representing barriers and goals. Furthermore, while our current implementation does not consider velocity and path integration, recent graph-based extensions incorporating these features⁴⁶ could in principle be combined with our compositional framework to handle directed transitions and velocity modulation. Thus, our model builds on these graph-based cognitive mapping theories, while more explicitly exposing the map's compositional structure, its role in planning, and the computational role of different neural responses.

The current theory expands on our prior linear RL model for planning, which efficiently reduces planning to constructing predictive maps³⁰. One key feature of the model, which distinguishes it from similar predictive algorithms like the SR, is that it is policy independent. That is, it represents predictions about long-run state encounters under a fixed, “default” policy (e.g., a random walk), but it can be used to solve approximately for the value function under any goal-optimized policy. In the earlier work, we showed how this enabled various types of transfer learning relevant to cognitive maps; in the current work, it is crucial computationally for enabling the stable compositional construction of the map and empirically for explaining various phenomena related to the stability of grid fields. While successor features offer another approach for improving the transfer of predictive maps beyond traditional SR^{61,62}, they lack compositionality in the predictive space as the entire map must be recomputed when features change. Overall, the current theory complements our linear RL model by delineating how these maps can be created compositionally from task elements, such as barriers, objects, and goals. Together, this provides an efficient, comprehensive, and psychobiologically plausible theory of model-based planning.

When facing two or more objects, we demonstrate that a first-order approximation of the solution can be obtained through the compositional combination of the representations related to each individual object. We further propose a learning algorithm for modeling interactions between objects. Importantly, this algorithm is efficient as its starting point is already a decent approximation of the full solution, as depicted in Fig. 1. Additionally, it is computationally efficient since the object-related representation matrices scale only with object size rather than environment size. This contrasts with much of the previous work on SR, in which learning the map at least required updating entire rows or columns of the SR matrix, which is the size of the state space⁴². Thus, by leveraging compositionality and initially approximating multi-object solutions from single-object representations, our approach provides an efficient framework for learning object interactions and constructing multi-object cognitive maps. Although the neural basis of this process remains unclear, mental simulation through hippocampal replay may offer insights into this process. During periods of wakefulness, place cells exhibit rapid, sequenced reactivations known as awake replays^{63–67}, which have been linked to planning^{66,68} and structural inference⁶⁹. Though the neural mechanisms are still unknown, awake replays represent a promising phenomenon to explore in future research on the neuroscience of compositional mapping.

The model makes distinct behavioral and neural predictions. Behaviorally, it predicts that when two previously seen objects are introduced together in a navigation task, performance will decline as a function of the distance between them, as closer objects require more substantial updates to their PORs. In contrast, locally compositional (but non-predictive) models do not predict a specific relationship with distance, since they only account for how objects affect transitions between neighboring states. Neurally, our model predicts relatively stable representations in the MEC, which is thought to encode the

relational structure of cognitive maps, with barriers only locally disrupting the map. This contrasts with other predictive models—most notably Stachenfeld et al.'s influential work³²—where grid fields, as eigenvectors of the predictive map, change dramatically with the introduction of barriers or objects. Furthermore, our model offers both biological and psychological plausibility that the eigenvector account lacks. Rather than relying on eigendecomposition, which becomes intractable in even moderately sized environments, our approach reframes the problem of learning a predictive map, reducing computational complexity from scaling with environment size to scaling with barrier size.

Our framework extends beyond simple, fully observable objects by transforming the challenge of learning complete cognitive maps into a more tractable problem of learning object-related representations. This approach copes somewhat gracefully with environment maps that are gradually learned, such as burrows that must be explored while they are being mapped (by incorporating objects and barriers into the baseline map as they become visible), as well as complex object interactions. Notably, this perspective may help explain the high proportion of object vector cells (30%) in the MEC observed experimentally, as these neurons must represent objects from multiple viewpoints and distances while handling interactions—computational demands that align with the challenges animals face in natural environments. Importantly, Høydal et al.³⁴ identified this high proportion using only simple objects, raising an intriguing question for future research on how MEC represents more complex objects.

Høydal et al.'s³⁴ experiments also highlight a fundamental distinction between object vector cells and border cells, particularly in response to suspended objects. While border cells encode barriers already integrated into the cognitive map, object vector cells respond to objects regardless of whether they physically constrain movement, e.g., firing for both grounded and suspended objects. We propose that this distinction reflects a computational difference: object vector cells maintain compositional representations that can be flexibly combined with the map as needed, rather than directly encoding current navigational constraints. This suggests that object vector cells act as a library of precomputed spatial relationships, ready for integration when relevant. Their invariance to distance and orientation relative to their preferred object, in contrast to border cells, supports this hypothesis. Additionally, previous methods for integrating objects into predictive maps impose significant computational complexity, even for simple objects. In contrast, our model naturally scales learning difficulty with object complexity, making it well-suited for environments with varying levels of structural detail.

However, while our framework aligns with key experimental observations, the responses to suspended objects also present a potential limitation. Since our model assumes that object vector cells encode predictive relationships based on how objects influence movement, our interpretation that these relationships are nevertheless latently represented for objects that do not, given their current placement, actually affect movement, remains debatable. Future research is needed to clarify this issue and further investigate how object vector cells contribute to flexible cognitive map construction. More broadly, while our model captures key computational principles of compositional cognitive maps, it does not yet address how agents navigate and maintain spatial knowledge in dynamically changing environments where it is initially difficult to adequately ascertain barrier placement. While the model is capable of building up maps gradually by adding barriers as they are observed, this may still be insufficient in real-world scenarios, where visual occlusion and the need to track object permanence pose important challenges for spatial navigation. Extending our framework to incorporate these elements would require mechanisms for maintaining and updating cognitive maps when the line of sight is blocked and for reasoning about occluding objects that may be movable. Our current focus on establishing core principles of compositional

planning using predictive maps aligns with other influential work on allocentric mapping in the MEC while abstracting away from the complexities of visual processing, but integrating these aspects remains an important avenue for future research.

The current theory provides a comprehensive and biologically realistic computational framework for constructing cognitive maps useful for model-based planning in the brain. This establishes a foundation for future work investigating the neural substrates underlying cognitive mapping. Although we have focused on the MEC's role in cognitive maps, map making likely involves multiple neural systems rather than a single localized mechanism. Different systems may construct maps in distinct contexts, like social, emotional, or spatial. While our framework models interactions between subprocesses of map making, we do not imply these computations are localized to specific regions. More realistically, various neural systems participate in different computational subprocesses of cognitive mapping. Overall, our theory provides a unifying computational account of cognitive mapping and model-based planning while remaining compatible with distributed, systems-level neural implementation. This integrative approach opens promising directions for examining the neural mechanisms of how diverse brain areas cooperate to enable efficient map making, and flexible planning.

Recent work has also highlighted that entorhinal cells often exhibit joint selectivity to multiple behavioral variables⁷⁰, raising important questions about how such mixed selectivity might be incorporated into our framework. Our model currently focuses on how objects and barriers modify the transition structure of open space, and is agnostic to joint selectivity, making this a straightforward potential extension. Our mapping of PORs to object vector cells is supported by their abundance in MEC³⁴. Moreover, recent evidence suggests boundary vector cells⁷¹, which are selective to boundaries and exhibit vectorial tuning (unlike border cells), may play a more substantial role than previously thought⁷². Although our model primarily focuses on object vector cells, it is, in principle, extendable to boundary vector cells, as both cell types encode vectorial relationships to environmental features and influence the transition structure of open space. Future work could explore this extension, further refining our understanding of how these distinct but related spatial representations contribute to cognitive mapping.

In this work, we have focused exclusively on navigation problems where objects and barriers alter the transition structure of physical space. However, many decision-making problems are framed in more abstract, discrete state spaces, where the notions of barriers and objects are less intuitive. Importantly, our proposed model remains applicable to such scenarios, as it can compute the effects of any change in transition structure on the predictive map, regardless of the specific nature of the state space. That said, there are certain classes of problems that our current framework cannot easily handle in its present form. Most notably, our study and the linear RL framework more broadly, has primarily addressed problems with discrete state spaces, which limits its direct applicability to classical RL problems involving continuous state spaces, such as motor control tasks, or problems with stochastic transitions, such as prey/predators. However, it is worth emphasizing that the linear RL framework is rooted in the linearly solvable class of Markov decision processes developed by Todorov^{36,37}, which naturally extends to continuous problems. This theoretical foundation offers a promising avenue for future work to adapt and expand our framework to continuous domains, with potential applications in sensorimotor neuroscience.

This work also aligns with emerging theoretical and empirical research tapping into the neural basis of representation learning, or how human and other animals construct task representations that allow efficient decision making^{73,74}. In particular, our model suggests that the way that an object or barrier changes the relational structure of the state space is of primary importance for planning, whereas the

sensory features of such objects are irrelevant and ignored by our model’s representations. This is not only compatible with representations of object vector cells in the MEC as shown here, but it is also consistent with recent evidence suggesting that neural representations in MEC and medial prefrontal cortex are relatively invariant to sensory aspects of task elements^{27,75}. While testing representational models remains challenging, recent advances demonstrate feasibility. For instance, combining neuroimaging and representational similarity analysis techniques^{76,77} has elucidated how the hippocampus and orbitofrontal cortex encode social cognitive maps^{75,78,79}. Further developing such techniques will be key to unlocking the computational principles represented in neural systems. An integrative interplay between computational modeling and representation-focused experimentation will likely offer new insights into the algorithms underlying cognitive map-making and model-based planning.

Within our model, maps are created by integrating object-related representations with the basis functions of the open space. This suggests local interactions between object vector cells and grid cells. More specifically, our approach suggests that the grid cells might receive inputs from other cell types, such as object vector cells⁸⁰. Although the neural mechanisms underlying the implementation of the proposed grid code in the MEC are still unclear, recent advances have brought us to the point where these underlying mechanisms can soon be elucidated^{57,81}. In particular, new electrophysiological techniques now enable simultaneous recording of hundreds of MEC neurons during navigation⁸². It would be important to leverage these techniques to conduct multi-neuronal recordings of different MEC cell types as animals move through environments containing objects and goals. Analyzing the interactions between cell types under these conditions could reveal how the grid code emerges from local microcircuit computations and whether those are consistent with the quantities predicted by the model.

Methods

Linear RL

We start with a review of the linear RL model³⁰, which considers finite-horizon Markov decision problems with deterministic dynamics, such as mazes. The linear RL model maximizes long-term gain, defined as a linear function of the reward for each state. In particular, if $r(s)$ is the reward in state s , and $\lambda > 0$ is a constant called control cost, then gain, g , is defined as $g(s) = r(s) - \lambda \text{KL}(\pi || \pi^d)$. Here, $\text{KL}(\pi || \pi^d)$ is the Kullback-Leibler divergence between the decision policy, π , and a default policy π^d (the policy under which the map is made, e.g., a random walk). This is a measure of how two probability distributions diverge from each other. It is zero if π equals π^d , and otherwise, it is positive. Throughout this work, we always assume a uniform default policy, which is equivalent to moving to all successor (i.e., immediately reachable) states with equal probability.

It is then straightforward to demonstrate that the optimal value function for this problem, $v^*(s)$, is analytically solvable. (It can also be viewed as an approximation to the value of the original RL problem, without control costs.) We start by defining the one-step state transition matrix, denoted as \mathbf{T} . The (i, j) element of \mathbf{T} represents the probability of transitioning from state i to state j under the default policy (i.e., probability of the action under the default policy that leads to transition from state i to state j). We further distinguish states into terminal states (containing goals) and all other nonterminal states. Next, we define $\mathbf{t} = \mathbf{T}_{NT} \exp(r_T/\lambda)$, a vector that depends only on the transition to the terminal state, \mathbf{T}_{NT} , and its reward, r_T . Now, if \mathbf{v}^* is the vector of optimal values across all nonterminal states, then $\exp(\mathbf{v}^*/\lambda) = \mathbf{D}_{NN} \mathbf{t}$. Here, \mathbf{D}_{NN} is a subblock of the DR matrix, \mathbf{D} , corresponding to nonterminal states. The DR is defined as:

$$\mathbf{D} = (\text{diag}(\exp(-\mathbf{r}/\lambda)) - \mathbf{T})^{-1}, \tag{4}$$

where \mathbf{r} is the reward vector across all states (assuming, without loss of generality, that reward at terminal states is not 0). Thus, the DR, \mathbf{D} , is a matrix where the (i, j) element indicates the expected occupancy of state j , discounted by exponential state rewards (which are negative), when starting from state i and following the default policy. It differs from the SR, which represents expected occupancy under the decision policy (and thus changes whenever the optimal values change, such as when goals move). The DR matrix \mathbf{D} is the main focus of the current work.

Compositional map making

Suppose that \mathbf{D}_{os} is the DR of the open space for a random walk (if one takes actions randomly according to a uniform transition probability). In practice, it can be efficiently defined by calculating the DR for a significantly larger environment and then extracting the DR corresponding to a subregion from the middle. The subregion should be of a size equivalent to the task environment. We assume that all state costs are equally given by the constant c . Thus, if \mathbf{T}_{os} is the transition matrix under the random walk, \mathbf{I} is the identity matrix with size equal to \mathbf{T}_{os} , and $\mathbf{L}_{os} = \exp(c)\mathbf{I} - \mathbf{T}_{os}$, then $\mathbf{D}_{os} = \mathbf{L}_{os}^{-1}$.

Now consider an object o that changes the transition matrix to \mathbf{T}_o . It is easy to see that \mathbf{T}_o can be written as a function of \mathbf{T}_{os} , and two low-rank matrices:

$$\mathbf{T}_o = \mathbf{T}_{os} + \mathbf{C}_o \mathbf{R}_o, \tag{5}$$

where \mathbf{C}_o and \mathbf{R}_o are, respectively, $S \times O$ and $O \times S$ matrices, where O is the number of states whose transition changed by introduction of the object, and $S \gg O$ is the number of states in the open space environment. Since the object only changes the transition for states that are neighbors to one of the states that is blocked by the object, O is essentially close to the size of the object. Both \mathbf{C}_o and \mathbf{R}_o are sparse matrices with nonzero elements only on rows or columns associated with the object and its neighbor states. In particular, if $\{s_1, s_2, \dots, s_o\}$ represents the set of states whose transitions were affected by the introduction of the object, the j th column of matrix \mathbf{C}_o is a binary vector with its s_j th element set to one. Additionally, the j th row of matrix \mathbf{R}_o is defined as the s_j th row of matrix $\mathbf{T}_o - \mathbf{T}_{os}$.

Now, if $\mathbf{L}_o = \exp(c)\mathbf{I} - \mathbf{T}_o = \mathbf{L}_{os} - \mathbf{C}_o \mathbf{R}_o$, we can use Woodbury matrix inversion identity⁸³ to write $\mathbf{D}_o = \mathbf{L}_o^{-1}$ as a function of \mathbf{D}_{os} :

$$\mathbf{D}_o = \mathbf{D}_{os} + \mathbf{D}_{os} \mathbf{C}_o \mathbf{A}_o \mathbf{R}_o \mathbf{D}_{os}, \tag{6}$$

where \mathbf{A}_o is a $O \times O$ matrix representing the POR for the object o . If we define $\mathbf{Z}_o = \mathbf{R}_o \mathbf{D}_{os} \mathbf{C}_o$, then \mathbf{A}_o is given by:

$$\mathbf{A}_o = (\mathbf{I} - \mathbf{Z}_o)^{-1}. \tag{7}$$

If $|\mathbf{Z}_o| < 1$, \mathbf{A}_o is given by Neumann series:

$$\mathbf{A}_o = \sum_{k=0}^{\infty} \mathbf{Z}_o^k = \mathbf{I} + \mathbf{Z}_o + \mathbf{Z}_o^2 + \mathbf{Z}_o^3 + \dots \tag{8}$$

Thus, $\mathbf{A}_o \simeq \mathbf{I} + \mathbf{Z}_o$ provides a first-order approximation of \mathbf{A}_o .

Now, let’s consider an environment that includes two distinct objects, labeled 1 and 2. We assume these objects are distinguished in such a way that there is no single state whose transition is affected by both objects. As a result, the lookup table matrices for the composition of the objects can be expressed as column- and row- concatenation of their associated matrices, i.e., $[\mathbf{C}_1 \ \mathbf{C}_2]$ and $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix}$.

By combining Eqs. 6 and 8, it is easy to see that Eq. 3 provides an approximation for the DR matrix \mathbf{D} of the new environment. This approximation approaches the exact solution as the distance between the two objects increases.

Learning PORs

Equation 3 provides a first-order approximation of the POR for two (or more) objects. Building on this, we have developed an algorithm for learning the POR. First, we define matrix \mathbf{Z} based on the lookup table matrices, \mathbf{C} and \mathbf{R} , for the composition of the two objects. These are defined as column- and row-concatenation of their corresponding matrices, i.e., $\mathbf{C} = [\mathbf{C}_1 \mathbf{C}_2]$ and $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix}$. Thus, we define $\mathbf{Z} = \mathbf{C}\mathbf{D}_{os}\mathbf{R}$. If \mathbf{A} is initially set to $\mathbf{I} + \mathbf{Z}$, and $0 < \alpha < 1$ is a scalar parameter representing the step size, we iteratively update our POR estimate, \mathbf{A} , using the following equation:

$$\mathbf{A} \leftarrow \mathbf{A} + \alpha(\mathbf{I} + \mathbf{Z} - \mathbf{A}). \quad (9)$$

It can be demonstrated that \mathbf{A} consistently converges to the exact solution, i.e., $(\mathbf{I} - \mathbf{Z})^{-1}$, with each update, as long as the Neumann series (Eq. 8) exists for \mathbf{Z} , i.e., $|\mathbf{Z}| < 1$. In addition, this algorithm is highly efficient for two reasons. Firstly, the initial point provides a strong approximation of the POR. Secondly, unlike classical temporal-difference algorithms for updating the SR^{39,42}, this one operates within the object space, given that the sizes of matrices \mathbf{A} and \mathbf{Z} are object-dependent.

Simulation of object vector cells

Our model characterizes object vector cells as encoding the projected POR, which captures the change in the DR induced by the introduction of an object into the environment (Eq. 1), i.e., $\mathbf{D}_o = \mathbf{D}_{os}\mathbf{C}_o\mathbf{A}_o\mathbf{R}_o\mathbf{D}_{os}$. Specifically, we model object vector cells as a row vector of \mathbf{D}_o corresponding to the specific state, let's say s , associated with the particular distance and direction from the object to which the cell is tuned (due to the vector coding⁴³). Thus, the object vector cell code is given by the s th row of the \mathbf{D}_o . For example, an object vector cell tuned to “north of object” corresponds to the row of the \mathbf{D}_o matrix representing that particular spatial relationship. This state-specific encoding aligns with the vector coding framework proposed for spatial navigation⁴³.

The environment for simulating the object vector cells was created by discretizing the plane into a network of triangular lattice points. It was assumed that every state is connected to any other state in its vicinity, determined by parameter r . This is equivalent to the assumption that the width place code is given by r . Additionally, we assumed that a barrier representation for object vector cells requires the object to be visible from the agent's position, aligning with empirical observations³⁴. This was modeled by assuming that a transition was possible from the agent's position to the object's location before the introduction of the object, and the object subsequently obstructed that transition.

Simulation of grid cells

The open space grid map was modeled using the framework and code developed by Dordek et al.⁴⁵. The open space environment consisted of uniformly distributed 2D Gaussian-shaped place cells, organized in a grid pattern. The activity of each place cell i at location \mathbf{x} was modeled as a normal distribution with mean c_i and variance σ_i^2 . The mean indicates the center of the place field for cell i , while the variance σ_i^2 determines the width of the place field. The place field centers were uniformly distributed across the environment, and all had the same width. The activity of N place cells over T time-points was governed by a random uniform policy dictating the agent's movement. This produced an $N \times T$ matrix of place cell activities. The eigenvectors were computed by running an eigendecomposition (or, equivalently, a principal component analysis) on this matrix. As in Dordek et al.⁴⁵, a non-negativity constraint was applied. The first 50 principal components calculated using the Oja rule⁴⁷ were used in our simulations. Since these components are calculated for the covariance matrix of place cell activities, we used the covariance matrix as \mathbf{D}_{os} to calculate \mathbf{G} in Eq. 2.

The task environment for each grid field simulation was created as a 2D square grid representing the tabular state space. For each task, barriers were modeled by blocking the corresponding transitions between states.

Simulation details

For the model simulation in Figs. 1 and 2, we assumed a 20×20 square maze environment. The cost for all states was set to 0.1, with the goal state having 0 cost. To minimize the effects of the borders on the open space DR, we first computed the DR for a significantly larger open space (100×100) and subsequently selected the subsection that corresponds to our maze environment. The step size in Equation 9 was 0.3. For implementing the SR, we assumed the reward vector was given (similar to the assumption for our model), and the algorithm only learned the SR map using a step size of 0.2. Actions were selected based on a softmax policy with decision noise of 1. Since this was a finite-horizon problem (i.e., planning towards a goal state), the discount factor was set to 1. Models that require updating their representations (such as SR and our Update model) are assumed to update once per step, similar to typical temporal-difference algorithms. For all simulations related to object vector cells presented in Figs. 3–5, the environment size was 20, and $r=2$ was used as the place code width. The cost for all states was 0.1.

For grid field simulations, we generally used the same set of parameters as the open space grid field. For the simulation in Fig. 6, each compartment was 30×30 , and σ was 0.35. To calculate the sliding spatial correlation, we assumed a box size width of 6, ensuring that its ratio compared to the maze size roughly resembles that of the original experiment. For the simulation presented in Fig. 7, the environment was 25×25 and σ was 0.35. It was assumed that states at the center of the home cage serve as terminal states, an assumption not made for the plain box. The box size for calculating the spatial correlation was 3. The normalized firing rate was computed as the average of fields within the corresponding distance surpassing a threshold, divided by the same metric across the entire maze in the open space environment. The threshold was defined as the average of all grid vectors in the open space. The grid vectors plotted in Fig. 7F were normalized by their maximum values, resulting in a range of values from 0 to 1. For calculating all spatial correlation maps, all states at the edge of the environment were excluded.

For the hairpin experiment presented in Fig. 8, the environment was 20×20 , and σ was 0.35. To calculate the correlation matrix between hairpin arms, the median correlation across all grid vectors exhibiting a significant alternating pattern (based on Wilcoxon rank difference) was computed for each arm. The turning point activity analysis involved computing the population activity of firing rates within paths containing a turning point and adjacent bins on both sides. Following the original study, our focus was on the activity of grid vectors that did not exhibit alternating patterns when calculating the turning matrix activity. As a result, the population firing rate was defined as the average of all such grid vectors. In our simulation, the turning point actually encompassed four states: one on one side of the barrier, two situated just next to the barrier but unblocked, and one on the other side of the barrier. The population activity at the turning point was determined as the median activity across all four states associated with the turning point. Furthermore, population activity across 12 states along each side of the barrier was also considered, resulting in vectors with a length of 25. The resulting population activity matrix was organized as a 25×9 matrix, where 9 represents the number of turning points in the environment. The turning correlation matrix was then calculated by computing correlations across columns of this matrix, resulting in a 25×25 matrix. This analysis was repeated 50 times with different randomization seeds, and the mean matrices were plotted in Fig. 8f–h.

For the simulation presented in Fig. 9, the environment was 25×25 and σ was 0.35. For calculating the number of grid fields moving closer to the goals, we first calculated the locations around the goal (all states within a radius of 2.5 of the center of a goal location). The peak activity near goals per grid vector was defined as the mean of all grid fields whose value was larger than a threshold. The threshold here was defined based on the mean normalized activity across all grid vectors, where the normalized activity is the median of the grid vector divided by its maximum. Mean number of vectors whose peak activity near goals was larger after the introduction of goal was the index plotted in Fig. 9f. The grid score³⁴ was calculated using the code implemented by Dordek et al.⁴⁵. The mean of grid scores was calculated across all vectors with an open space grid score exceeding 0.5 and used in Fig. 9g. For analyzing attraction as a function of distance, we took into account all active fields, defined as those with normalized activity (ranging between 0 and 1) exceeding half. The mean distance between the field peak location and the center of each of the three goals served as the corresponding distance metric for that active field. For each active field, the attraction vector was defined if the median activity of the vector in the vicinity of the peak location increased following the introduction of the goals, considering any location within a distance of less than 2.5 as part of the field's vicinity. The field attraction plotted in Fig. 9h was defined as the mean of the attraction vector. This analysis was repeated 50 times with different randomization seeds, and the mean metrics were plotted in Fig. 9f–h.

The control cost parameter, λ , in Eq. 4 was set to 1 for all simulations.

Reporting summary

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

Data availability

No new data were collected for this study. Simulated data are provided along with the code.

Code availability

Analyses were conducted using custom code written in Python and MATLAB. The codes are available at <https://doi.org/10.5281/zenodo.15811496>. The MATLAB code uses the code by Dordek et al.⁴⁵.

References

- Bienenstock, E., Geman, S. & Potter, D. Compositionality, MDL priors, and object recognition. In *Proc. 9th International Conference on Neural Information Processing Systems* 838–844 (MIT Press, 1996).
- Chang, M. B., Ullman, T., Torralba, A. & Tenenbaum, J. B. A Compositional Object-Based Approach to Learning Physical Dynamics. In *International Conference on Learning Representations* (2017).
- Frankland, S. M. & Greene, J. D. Concepts and compositionality: in search of the Brain's Language of Thought. *Annu. Rev. Psychol.* **71**, 273–303 (2020).
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behav. Brain Sci.* **40**, e253 (2017).
- Spelke, E. S. *What Babies Know: Core Knowledge and Composition 1* (Oxford University Press, 2022).
- Fodor, J. A. & Pylyshyn, Z. W. Connectionism and cognitive architecture: a critical analysis. *Cognition* **28**, 3–71 (1988).
- Gershman, S. J., Horvitz, E. J. & Tenenbaum, J. B. Computational rationality: a converging paradigm for intelligence in brains, minds, and machines. *Science* **349**, 273–278 (2015).
- Goyal, A. & Bengio, Y. Inductive biases for deep learning of higher-level cognition. *Proc. R. Soc. Math. Phys. Eng. Sci.* **478**, 20210068 (2022).
- Hill, C. A. et al. A causal account of the brain network computations underlying strategic social behavior. *Nat. Neurosci.* **20**, 1142–1149 (2017).
- Greff, K., van Steenkiste, S. & Schmidhuber, J. On the binding problem in artificial neural networks. Preprint at <https://doi.org/10.48550/arXiv.2012.05208> (2020).
- Lake, B. M. Compositional generalization through meta sequence-to-sequence learning. *Adv. Neural Inf. Process. Syst.* **32**, 9791–9801 (2019).
- Lake, B. M. & Baroni, M. Human-like systematic generalization through a meta-learning neural network. *Nature* **623**, 115–121 (2023).
- Liu, N. et al. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision* (eds. Avidan, S. et al.) 423–439 (Springer, 2022).
- Meng, C. et al. SDEdit: guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations* (2022).
- Nie, W., Vahdat, A. & Anandkumar, A. Controllable and compositional generation with latent-space energy-based models. *Adv. Neural Inf. Process. Syst.* **34**, 13062–13074 (2021).
- Behrens, T. E. J. et al. What is a cognitive map? Organizing knowledge for flexible behavior. *Neuron* **100**, 490–509 (2018).
- McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I. & Moser, M.-B. Path integration and the neural basis of the 'cognitive map'. *Nat. Rev. Neurosci.* **7**, 663–678 (2006).
- O'Keefe, J. & Nadel, L. *The Hippocampus as a Cognitive Map* (Oxford University Press, 1978).
- Schacter, D. L., Addis, D. R. & Buckner, R. L. Remembering the past to imagine the future: the prospective brain. *Nat. Rev. Neurosci.* **8**, 657–661 (2007).
- Tolman, E. C. Cognitive maps in rats and men. *Psychol. Rev.* **55**, 189–208 (1948).
- Ho, M. K. et al. People construct simplified mental representations to plan. *Nature* **606**, 129–136 (2022).
- Hunt, L. T. et al. Formalizing planning and information search in naturalistic decision-making. *Nat. Neurosci.* **24**, 1051–1064 (2021).
- Tsividsi, P. A. et al. Human-level reinforcement learning through theory-based modeling, exploration, and planning. Preprint at <https://doi.org/10.48550/arXiv.2107.12544> (2021).
- Daw, N. D., Niv, Y. & Dayan, P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nat. Neurosci.* **8**, 1704–1711 (2005).
- Glascher, J., Daw, N., Dayan, P. & O'Doherty, J. P. States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron* **66**, 585–595 (2010).
- George, D. et al. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nat. Commun.* **12**, 2392 (2021).
- Whittington, J. C. R. et al. The Tolman-Eichenbaum Machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell* **183**, 1249–1263.e23 (2020).
- de Cothi, W. et al. Predictive maps in rats and humans for spatial navigation. *Curr. Biol.* **32**, 3676–3689.e5 (2022).
- Duvelle, É. et al. Hippocampal place cells encode global location but not connectivity in a complex space. *Curr. Biol.* **31**, 1221–1233.e9 (2021).
- Piray, P. & Daw, N. D. Linear reinforcement learning in planning, grid fields, and cognitive control. *Nat. Commun.* **12**, 4942 (2021).
- Rueckemann, J. W., Sosa, M., Giocomo, L. M. & Buffalo, E. A. The grid code for ordered experience. *Nat. Rev. Neurosci.* **22**, 637–649 (2021).
- Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643–1653 (2017).

33. Andersson, S. O., Moser, E. I. & Moser, M.-B. Visual stimulus features that elicit activity in object-vector cells. *Commun. Biol.* **4**, 1219 (2021).
34. Høydal, ØA., Skytøen, E. R., Andersson, S. O., Moser, M.-B. & Moser, E. I. Object-vector coding in the medial entorhinal cortex. *Nature* **568**, 400–404 (2019).
35. Kinkhabwala, A. A., Gu, Y., Aronov, D. & Tank, D. W. Visual cue-related activity of cells in the medial entorhinal cortex during navigation in virtual reality. *eLife* **9**, e43140 (2020).
36. Todorov, E. Linearly-solvable Markov decision problems. In *Proc. Advances in Neural Information Processing Systems* Vol. 19 (eds Schölkopf, B., Platt, J. C. & Hoffman, T.) 1369–1376 (MIT Press, 2007).
37. Todorov, E. Efficient computation of optimal actions. *Proc. Natl. Acad. Sci. USA* **106**, 11478–11483 (2009).
38. Bazarjani, A. & Piray, P. Efficient learning of predictive maps for flexible planning. Preprint at <https://doi.org/10.31234/osf.io/ak57f> (2025).
39. Dayan, P. Improving generalization for temporal difference learning: the successor representation. *Neural Comput.* **5**, 613–624 (1993).
40. Gershman, S. J. The successor representation: its computational logic and neural substrates. *J. Neurosci.* **38**, 7193–7200 (2018).
41. Momennejad, I. et al. The successor representation in human reinforcement learning. *Nat. Hum. Behav.* **1**, 680–692 (2017).
42. Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Comput. Biol.* **13**, e1005768 (2017).
43. Bicanski, A. & Burgess, N. Neuronal vector coding in spatial cognition. *Nat. Rev. Neurosci.* **21**, 453–470 (2020).
44. Deshmukh, S. S. & Knierim, J. J. Influence of local objects on hippocampal representations: landmark vectors and memory. *Hippocampus* **23**, 253–267 (2013).
45. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *eLife* **5**, e10094 (2016).
46. Yu, C., Behrens, T. E. J. & Burgess, N. Prediction and generalisation over directed actions by grid cells. In *International Conference on Learning Representation* (2021).
47. Oja, E. A simplified neuron model as a principal component analyzer. *J. Math. Biol.* **15**, 267–273 (1982).
48. Wernle, T. et al. Integration of grid maps in merged environments. *Nat. Neurosci.* **21**, 92–101 (2018).
49. Sanguinetti-Scheck, J. I. & Brecht, M. Home, head direction stability, and grid cell distortion. *J. Neurophysiol.* **123**, 1392–1406 (2020).
50. Derdikman, D. et al. Fragmentation of grid cell maps in a multi-compartment environment. *Nat. Neurosci.* **12**, 1325–1332 (2009).
51. Nyberg, N., Duvelle, É, Barry, C. & Spiers, H. J. Spatial goal coding in the hippocampal formation. *Neuron* **110**, 394–422 (2022).
52. Boccara, C. N., Nardin, M., Stella, F., O'Neill, J. & Csicsvari, J. The entorhinal cognitive map is attracted to goals. *Science* **363**, 1443–1447 (2019).
53. Butler, W. N., Hardcastle, K. & Giocomo, L. M. Remembered reward locations restructure entorhinal spatial maps. *Science* **363**, 1447–1452 (2019).
54. Bellmund, J. L. S., Gärdenfors, P., Moser, E. I. & Doeller, C. F. Navigating cognition: Spatial codes for human thinking. *Science* **362**, eaat6766 (2018).
55. Burak, Y. & Fiete, I. R. Accurate path integration in continuous attractor network models of grid cells. *PLOS Comput. Biol.* **5**, e1000291 (2009).
56. Sorscher, B., Mel, G. C., Ocko, S. A., Giocomo, L. M. & Ganguli, S. A unified theory for the computational and mechanistic origins of grid cells. *Neuron* **111**, 121–137.e13 (2023).
57. Ginosar, G., Aljadeff, J., Las, L., Derdikman, D. & Ulanovsky, N. Are grid cells used for navigation? On local metrics, subjective spaces, and black holes. *Neuron* **111**, 1858–1875 (2023).
58. Ginosar, G. et al. Locally ordered representation of 3D space in the entorhinal cortex. *Nature* **596**, 404–409 (2021).
59. Krupic, J., Bauza, M., Burton, S. & O'Keefe, J. Local transformations of the hippocampal cognitive map. *Science* **359**, 1143–1146 (2018).
60. Peer, M., Brunec, I. K., Newcombe, N. S. & Epstein, R. A. Structuring knowledge with cognitive maps and cognitive graphs. *Trends Cogn. Sci.* **25**, 37–54 (2021).
61. Barreto, A. et al. Successor features for transfer in reinforcement learning. In *Proc. Advances in Neural Information Processing Systems* Vol. 30 (Curran Associates, Inc., 2017).
62. de Cothi, W. & Barry, C. Neurobiological successor features for spatial navigation. *Hippocampus* **30**, 1347–1355 (2020).
63. Davidson, T. J., Kloosterman, F. & Wilson, M. A. Hippocampal replay of extended experience. *Neuron* **63**, 497–507 (2009).
64. Diba, K. & Buzsáki, G. Forward and reverse hippocampal place-cell sequences during ripples. *Nat. Neurosci.* **10**, 1241–1242 (2007).
65. Foster, D. J. & Wilson, M. A. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* **440**, 680–683 (2006).
66. Pfeiffer, B. E. & Foster, D. J. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature* **497**, 74–79 (2013).
67. Schuck, N. W. & Niv, Y. Sequential replay of nonspatial task states in the human hippocampus. *Science* **364**, eaaw5181 (2019).
68. Jadhav, S. P., Kemere, C., German, P. W. & Frank, L. M. Awake hippocampal sharp-wave ripples support spatial memory. *Science* **336**, 1454–1458 (2012).
69. Evans, T. & Burgess, N. Coordinated hippocampal-entorhinal replay as structural inference. In *Proc. Advances in Neural Information Processing Systems* Vol. 32 (Curran Associates, Inc., 2019).
70. Hardcastle, K., Maheswaranathan, N., Ganguli, S. & Giocomo, L. M. A Multiplexed, heterogeneous, and adaptive code for navigation in medial entorhinal cortex. *Neuron* **94**, 375–387.e7 (2017).
71. Lever, C., Burton, S., Jeewajee, A., O'Keefe, J. & Burgess, N. Boundary vector cells in the subiculum of the hippocampal formation. *J. Neurosci.* **29**, 9771–9777 (2009).
72. Muessig, L. et al. Environment geometry alters subiculum boundary vector cell receptive fields in adulthood and early development. *Nat. Commun.* **15**, 982 (2024).
73. Niv, Y. Learning task-state representations. *Nat. Neurosci.* **22**, 1544–1553 (2019).
74. Wilson, R. C., Takahashi, Y. K., Schoenbaum, G. & Niv, Y. Orbitofrontal cortex as a cognitive map of task space. *Neuron* **81**, 267–279 (2014).
75. Park, S. A., Miller, D. S. & Boorman, E. D. Inferences on a multi-dimensional social hierarchy use a grid-like code. *Nat. Neurosci.* **24**, 1292–1301 (2021).
76. Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis—connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* **2**, 4 (2008).
77. Nili, H. et al. A toolbox for representational similarity analysis. *PLoS Comput. Biol.* **10**, e1003553 (2014).
78. Park, S. A., Miller, D. S., Nili, H., Ranganath, C. & Boorman, E. D. Map making: constructing, combining, and inferring on abstract cognitive maps. *Neuron* **107**, 1226–1238.e8 (2020).
79. Mahmoodi, A., Luo, S., Harbison, C., Piray, P. & Rushworth, M. F. S. Human hippocampus and dorsomedial prefrontal cortex infer and update latent causes during social interaction. *Neuron* **112**, 1–14 (2024).
80. Moser, M.-B., Rowland, D. C. & Moser, E. I. Place cells, grid cells, and memory. *Cold Spring Harb. Perspect. Biol.* **7**, a021808 (2015).

81. Tukker, J. J. et al. Microcircuits for spatial coding in the medial entorhinal cortex. *Physiol. Rev.* **102**, 653–688 (2022).
82. Gardner, R. J. et al. Toroidal topology of population activity in grid cells. *Nature* **602**, 123–128 (2022).
83. Hager, W. W. Updating the Inverse of a Matrix. *SIAM Rev.* **31**, 221–239 (1989).
84. Sargolini, F. et al. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science* **312**, 758–762 (2006).

Acknowledgements

This work was supported by grants R21MH134217 from the National Institute of Mental Health (P.P.), IIS-1822571 from the National Science Foundation, part of the CRNCS program, and ARO grant W911NF-16-1-0474 (N.D.D.).

Author contributions

P.P. and N.D.D. designed the study. P.P. created the model and performed analyses. P.P. and N.D.D. wrote the manuscript.

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/s41467-025-62733-7>.

Correspondence and requests for materials should be addressed to Payam Piray.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

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

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025