Neural sampling from cognitive maps enables goal-directed imagination and planning

(August 6, 2025)

Hui Lin[1,2,*], Yukun Yang[2,*], Rong Zhao[1], Giovanni Pezzulo[3], Wolfgang Maass[2,†]

[1]Department of Precision Instruments, Center for Brain-Inspired Computing Research (CBICR), Tsinghua University, Beijing, China.
[2]Institute of Machine Learning and Neural Computation, Graz University of Technology, Graz, 8010, Austria.
[3]Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy.
[*]These authors contributed equally.
[†]Corresponding author.

Contributing authors: linhui21@mails.tsinghua.edu.cn; yukun.yang@tugraz.at; giovanni.pezzulo@istc.cnr.it; maass@igi.tugraz.at;

Abstract

AI systems are becoming increasingly more intelligent, but at a very high cost in terms of energy consumption and training requirements. In contrast, our brains only require 20W of energy, they learn online, and they can instantly adjust to changing contingencies. This begs the question what data structures, algorithms, and learning methods enable brains to achieve that, and whether these can be ported into artificial devices. We are addressing this question for a core feature of intelligence: The capacity to plan and solve problems, including new problems that involve states which were never encountered before. We examine from the algorithmic perspective three tools that brains are likely to employ for achieving that: Stochastic neural computation, cognitive maps, and compositional coding. We integrate these tools into a transparent neural network model without hidden units, and demonstrate its power for flexible planning and problem solving. Importantly, this approach is suitable for implementation by in-memory computing and other energy-efficient neuromorphic hardware. In particular, it only requires self-supervised local synaptic plasticity that is suited for on-chip learning. Hence a core feature of brain intelligence, the capacity to generate solutions to problems that were never encountered before, does not require deep neural networks or large language models, and can be implemented in energy-efficient edge devices.

1 Introduction

Planning and problem solving are at the core of human intelligence. This capability is apparent in multiple domains, such as when planning routes to reach a goal location in a physical environment or when imagining solutions to abstract problems. Importantly, we can apply it also for coping with new challenges that go beyond previous experience. Experimental data provide important clues about tools that the brain employs for this: generative neural computing, imagination of consequences of actions, and compositional coding. But we are lacking a deeper understanding of these tools both on the functional level, and on the mechanistic level of neural network implementations. AI tools for

solving such problems have suboptimal latency, and come with high costs in terms of hardware size, energy consumption, and offline training requirements on large data sets.

Numerous experimental data suggest that *cognitive maps*, i.e., data structure for coherently organizing learned experiences by encoding relational information between states and actions, are essential tools that the brain employs for planning. The most widely studied ones are allocentric spatial maps in the rodent hippocampal formation and entorhinal cortex that relate locations in an environment with changes caused by ego-motion (O'Keefe and Nadel, 1978; Behrens et al., 2018; Raju et al., 2024; Moser et al., 2008; George et al., 2021). They rely crucially on the largely innate grid cell system for their formation (Behrens et al., 2018; Hafting et al., 2005; Dong and Fiete, 2024; McNaughton et al., 2006). These cognitive maps support goal-oriented spatial navigation by providing self-localization and a sense of direction towards goals (Epstein et al., 2017; Bush et al., 2015; Tsoar et al., 2011). Recent experimental work in neuroscience and cognitive science goes beyond a purely spatial definition of cognitive maps, suggesting that they are also employed by the brain, especially the human brain, for problem solving in non-spatial domains, including abstract concept spaces (Behrens et al., 2018; Bottini and Doeller, 2020; Tavares et al., 2015; Xiao et al., 2025; Buzsáki and Moser, 2013; Iyer et al., 2024; FeldmanHall et al., 2025).

Furthermore, multiple lines of evidence suggest that cognitive maps of the brain not only encode past experience but also serve a prospective function, commonly referred to as *imagination*, which enables individuals to anticipate and prepare for situations they have never directly experienced. A rodent study showed that spontaneously generated ("replay") sequences of place cells in the hippocampus can be decoded as imagined trajectories in 2D spaces that resemble locomotion plans (Pfeiffer and Foster, 2013; Ólafsdóttir et al., 2018; Vollan et al., 2025). Furthermore, when novel barriers are introduced in the environment, replay trajectories depict goal-directed trajectories around them and largely avoid barrier violations, without changes in the place fields that are encoded by hippocampal neurons (Widloski and Foster, 2022). Hippocampal replay can also represent novel experiences, by recombining previous experiences in compositional ways. For example, in rodents, replay activity can encode "shortcuts" between two connected parts of a maze (Gupta et al., 2010). The hippocampus can also exhibit coordinated replay with other brain regions, such as the prefrontal cortex, in support of goal-directed behavior (Shin and Jadhav, 2016; Tang et al., 2021; Van de Maele et al., 2024).

The human brain is able to imagine and plan also in compositional domains, such as natural language, where a virtually unlimited repertoire of new states and potential goals can be described through new combinations of known components. It has recently been shown in (Schwartenbeck et al., 2023) that the human brain engages replay activity also for problem solving in such *compositional computing* tasks. The task domain which was examined there consisted of a virtually unlimited repertoire of new compositions of a known set of building blocks (tiles) into 2D silhouettes. This domain is arguably more transparent than natural language, but it still exhibits a core feature of compositional computing, its computational complexity. In fact, this decomposition task is NP-hard, which implies that every known deterministic algorithm requires computational resources that grow exponentially with the problem size(Moore and Robson, 2001; Russell and Norvig, 2016). Recent experimental data suggest that the human brain solves it by addressing possible building blocks in a sequential manner (Schwartenbeck et al., 2023). Other experimental studies demonstrate more generally that replay from spatial and non-spatial cognitive maps provides a neural substrate for goal-directed imagination and compositional computing. But few studies have so far advanced mechanistic explanations of these phenomena (Foster, 2017; Stoianov et al., 2022; Bakermans et al., 2025; Kurth-Nelson et al., 2023; Wittkuhn et al., 2021).

We integrate these three identified tools that brains employ for planning and problem solving, stochastic neural computation, cognitive maps, and compositional computing, into a novel model for planning and problem solving that also can solve problems that involve previously not experienced parts of spatial or abstract task spaces. This model can be implemented through simple and transparent neural networks that do not require Backprop (BP) or Backprop through time (BPTT) for learning. Instead, it learns autonomously in an online manner from own experiences, using only biologically plausible rules for local synaptic plasticity that are easy to implement in neuromorphic hardware.

Our approach provides a major conceptual and methodological advance in neural sampling, i.e., for stochastic computations in neural networks of the brain and in artificial devices. The idea that noise is an important feature of brain intelligence, rather than a bug, had gained substantial traction through experimental data from the Lab of József Fiser and others, see e.g. (Berkes et al., 2011).

These experimental data suggest that the brain employs probability distributions as data structures for learnt knowledge, from which it samples during inference. The article (Buesing et al., 2011) coined the term neural sampling for this style of neural computation. This concept has since been addressed in a large number of studies in neuroscience, cognitive science, neuromorphic engineering, and computational theory. In particular, the resulting paradigm for stochastic neural computation had been widely adopted in neuromorphic engineering, e.g. on the neuromorphic chip Loihi from Intel (Davies et al., 2018, 2021).

This approach has arguably plateaued during recent years, partially because insight was lacking about the nature of computational tasks that are solved by the brain through neural sampling, and about neural network architectures which it uses for that. The previously cited studies suggest that brains use neural sampling for planning and problem solving, and that cognitive maps in the hippocampus play a key role for that. Hence these are essential factors of the new model for neural sampling, more precisely for goal-directed sampling, that we are presenting.

We exhibit three major application domains for this new approach to neural sampling. These encompass two different models of cognitive maps – a model of the grid cell system for 2D environments (McNaughton et al., 2006; Dong and Fiete, 2024) and the *cognitive map learner (CML)* (Stöckl et al., 2024) for handling complex planning tasks in non-spatial domains. We show how these models can be turned into probabilistic generative models that generate sequences of (imagined) actions in a goal-directed manner. This extension permits testing the novel hypothesis that we advance in this study: namely, that sampling from cognitive maps supports goal-directed planning and imagination, both for spatial and for non-spatial domains. Surprisingly, cognitive maps provide a sense of direction even in complex compositional task domains such as composing and decomposing silhouettes from given building blocks. Essential for that is that the standard forward model that defines the relational structure of a cognitive map is complemented by an inverse model that maps state difference to neural codes for actions which cause these state differences. Furthermore, very simple inverse models for cognitive maps often suffice, which provide especially attractive generalization capabilities. In general, inverse models have become a standard modeling tool for biological motor control and in robotics (Wolpert and Kawato, 1998; Kawato, 1999), but they have apparently not yet been considered in the context of cognitive maps.

Reinforcement learning (RL) is the standard tool in AI for learning to solve planning or more general problem solving tasks (Sutton and Barto, 2018). RL also provides the most common framework in computational neuroscience for modeling learning of intelligent behavior. But in contrast to cognitive map based models, which can instantly be applied to plan a route to a new goal, most RL methods require relearning and substantial computational overhead when the goal changes. The reason is that RL is based on the computation of value functions, which depend in general on the given goal and the current policy. This value function is replaced in our cognitive map based approach by a sense of direction in a cognitive map, which enables action selection with "foresight" also in the absence of a value function. A simple example is spatial navigation, where steps for reaching a distal goal are chosen based on an analysis whether they point in the direction of the goal. Importantly, this method can be applied to any new goal, without requiring any extra computing when the goal changes.

We will first validate the functional abilities of our model for 2D spatial navigation. We use there a simple model for learning cognitive maps that is based on the grid cell system, which uses path integration to update the spatial location that is encoded by grid cell firing (McNaughton et al., 2006; Dong and Fiete, 2024). We show that the resulting cognitive map also gives rise to a generative model, enabling sampling of imagined trajectories towards any given spatial goal which resemble replay trajectories in the rodent brain (Foster, 2017; Widloski and Foster, 2022).

Next, to address goal-directed sampling in non-spatial domains, we use a simple model for learning cognitive maps that is based on learning predictions of action outcomes through local and biologically-plausible synaptic plasticity rules (Stöckl et al., 2024). We introduce a generative variant of this Cognitive Map Learner (CML) model, the *Generative Cognitive Map Learner (GCML)*, where observations of next states that result from own actions are replaced by internally generated state predictions. If noise is present in the selection of virtual next actions, one arrives at a probabilistic generative model that permits sampling of possible trajectories to any given goal in the cognitive map. We first illustrate functional capabilities of the GCML in a standard framework of AI for problem solving: Finding a short path from a given start to a given goal in an abstract graph (Russell and Norvig, 2016). In contrast to the CML, the GCML can not only generate an approximately shortest path, but a multitude of possible solutions. Hence it provides approximate solutions to the k-shortest-paths problem, where one wants the generate the k shortest paths for a given number k.

More generally, the GCML provides a menu of possible problem solutions, out of which the agent can select one that best satisfies additional contingencies or preferences. Finally, we show that the GCML can also be applied in a compositional domain. More precisely, we use it to solve the same tiling task that was considered in the recent study (Schwartenbeck et al., 2023) for elucidating human algorithms for solving problems in a compositional domain. It turns out that the same method can solve a variety of other tasks in this compositional domain, involving not only decomposition but also composition of building blocks.

2 Results

2.1 Sampling from a 2D cognitive map reproduces brain data on imagined spatial trajectories

Numerous neural recordings during self-generated imagination processes in the brain are available from the rodent hippocampus. In particular, it was shown in (Pfeiffer and Foster, 2013) that before navigation towards a known goal location ("home"), the hippocampus produced during specific resting phases (short wave ripples), sequential activity of place cells that could be decoded as imagined trajectories from the current location to the goal location. These decoded trajectories exhibited a startling diversity, even for the same starting and goal location, and were rarely straight (see Fig. 2 and Suppl. Fig. 11 and 17 of (Pfeiffer and Foster, 2013)). A subsequent study showed that these decoded trajectories can flexibly reach goals even when this requires rerouting around novel barriers (Widloski and Foster, 2022). These findings cannot be easily accounted for by standard sampling approaches used in statistics and machine learning, such as Markov Chain Monte Carlo (MCMC) and Monte Carlo Tree Search (MCTS) (Bishop, 2006; Gelly and Silver, 2011; Buesing et al., 2011), since these are lacking the goal directed feature. Hence new model for goal-directed stochastic sampling is needed, which we will present here.

We start with a standard model for a cognitive map for spatial navigation that is based upon the grid cell system, see Fig. 1a, and Methods for details. Grid cells produce an allocentric map of a 2D environment based on path integration, thereby relating ego-motion to estimates of spatial locations in the 2D environment (McNaughton et al., 2006; Behrens et al., 2018; Hafting et al., 2005; Liu et al., 2023; Dong and Fiete, 2024). Experimental data suggest that the grid cell system is largely genetically programmed in the rodent, but its development is sped up through spatial navigation (Ulsaker-Janke et al., 2023). Importantly for our context, it has been shown in (Vollan et al., 2025) that the grid cell systems is instrumental in generating imagined trajectories (forward sweeps) in the absence of movements. In fact, their data suggest the presence of a forward model where grid cell states are sequentially updated in the direction of an imagined movement. The allocentric map formed by the grid cell system corresponds on a large scale to a torus (Gardner et al., 2022). We focus on a local segment of the torus, which is approximately flat (Fig. 1b). Whereas each grid cell fires for numerous locations in 2D, they can be used as basis for training place cells that each fire only around a specific location, see Fig. 1a.

Previous analyses and models suggested that the diversity of 2D trajectories that are decoded from sequential activity of place cells exhibits signs of an underlying stochastic sampling process (Ujfalussy and Orbán, 2022; Stoianov et al., 2022). But a model was missing that could reproduce sampling of goal directed trajectories, such as the virtual trajectories to a "home" location in the rodent (Pfeiffer and Foster, 2013). We show here that such a model emerges if one complements the standard model for the representations by grid- and place cells by a simple self-supervised learning process for an inverse model, that learns during active locomotion to map small changes in grid cell activity to local movement commands which caused them. This inverse model can be implemented by a weight matrix $\mathbf{W}$, see Fig. 1a. It is analogous to numerous kinds of inverse models that have previously been employed for modeling biological motor control (Wolpert and Kawato, 1998; Kawato, 1999).

Learning of W during exploration of the 2D space (see Fig. 1d) can be implemented by a simple local plasticity rule for synaptic connections from grid cells to action-generating neurons. Due to the approximate linearity of the representation of the 2D space by grid cells and the linearity of $\mathbf{W}$ one can use this learnt map from state changes to action commands also for selecting a first action that moves into the direction of a distant goal s*, see Fig. 1b and the underlying theory presented in (Stöckl et al., 2024). If this action selection is in addition subject to noise, a model for goal directed stochastic sampling of trajectories emerges. Specifically, the $\mathbf{W}$ matrix maps the difference between grid cell codes $(\mathbf{s}^* - \hat{\mathbf{s}}_t)$ for a given goal and the current location onto a superposition (a) of basic

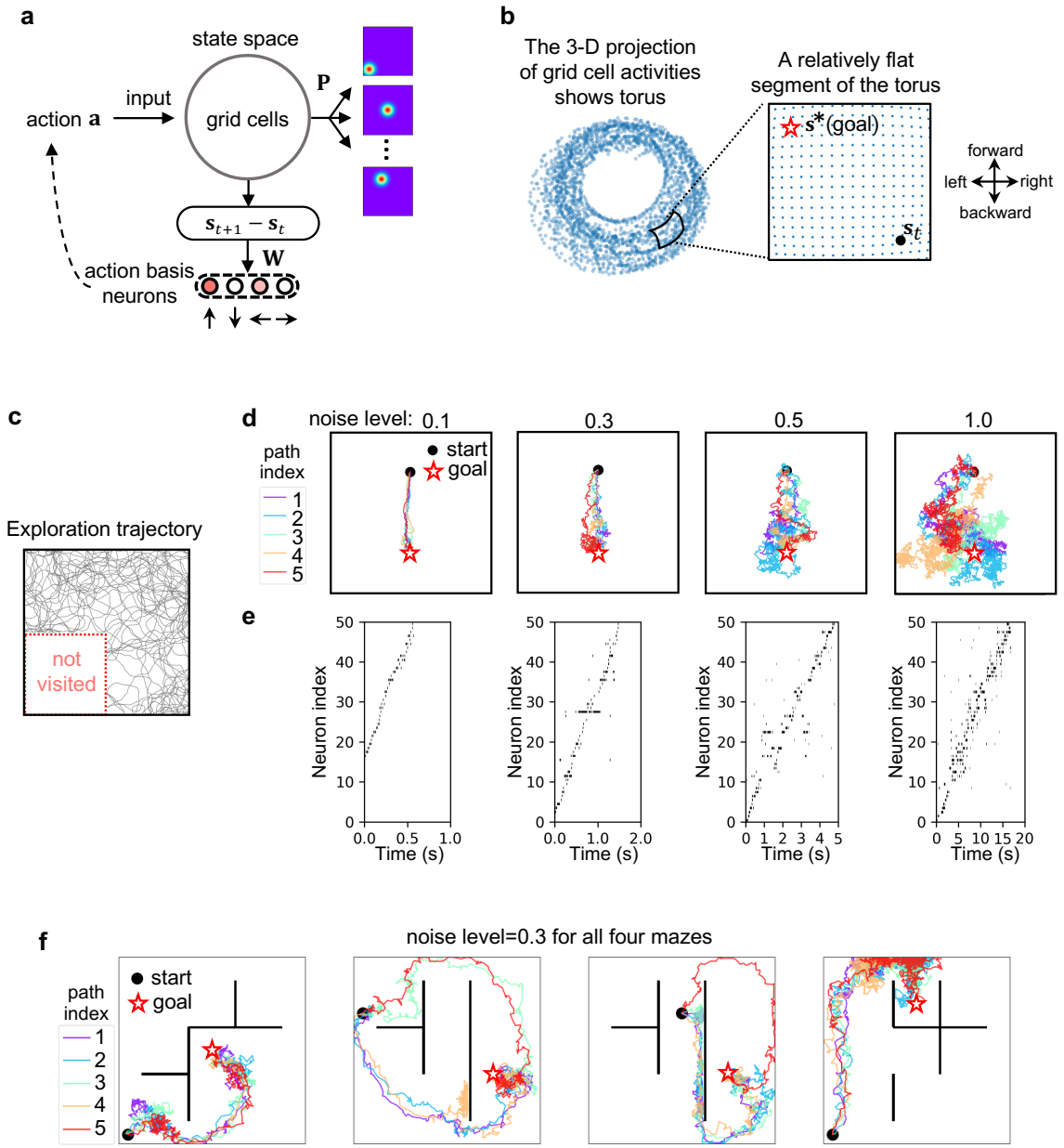

Fig. 1: Sampling from a 2D cognitive map supports goal-directed imagination of spatial trajectories. (a) Model architecture. A standard path-integration model for grid cells (Campbell et al., 2018; Sorscher et al., 2023) is used to update grid cell firing on the basis of movement commands (actions) a. These are represented here as (normalized) superpositions of basic movement commands in 4 directions. Grid cells provide synaptic inputs to an array of place cells that are trained through a simple local learning mechanism. Simultaneously, also synaptic connections $\mathbf{W}$ from grid cells to basis action neurons are learned by a simple local synaptic plasticity rule, yielding a linear inverse model. (b) The output of grid cells (for a specific scale) maps a 2D spatial environment onto a segment of the surface of a torus (Gardner et al., 2022). If the grid cells operate on a sufficiently large scale, this segment is approximately flat for a given 2D arena, thereby supporting navigation based on a sense of direction in 2D. (c) Exploration trajectory used to learn the inverse model $\mathbf{W}$. We deliberately left a quarter of the space unvisited, where the trajectory is reflected at its perimeter, in order to test whether our model can generate imagined trajectories that also visit unexplored parts of the 2D arena. (d) Impact of different noise levels (0.1, 0.3, 0.5, and 1.0) during action generation on the variance of the generated trajectories. We show five trajectories for each noise level. (e) Spike trains of 50 selected place cells for one of the trajectories (marked in red) in each corresponding plot of panel d. (f) Goal-directed trajectories generated by the same model where repulsive forces from obstacles affect action selection. These trajectories look remarkable similar to those observed in the rodent brain (Widloski and Foster, 2022).

action commands that would cause, if executed, a movement step from the current location into the direction of the goal. If one replaces the grid cell representation of the next location that would result from this movement step by an estimate that is produced by the forward model, one can iterate this process, yielding an imagined sequence of steps in the direction of the goal. The variance of these imagined trajectories depends on the level of the noise that is superimposed on each action selection, see Fig. 1d. The diversity of trajectories to the given goal that are generated in this way by the model is in the high noise regime qualitatively similar to the one shown in Fig. 4b of (Pfeiffer and Foster, 2013). This predicts that the underlying level of noise in the hippocampus during generation of imagined trajectories is fairly high. Fig. 1e shows the spiking activity of place cells (based on the input which they receive from grid cells according to the architecture of Fig. 1a). More precisely, the underlying spike trains that generate the red-colored trajectories in Fig. 1d are shown. They correspond to recorded spike sequences of place cells commonly observed in replay studies (Foster, 2017). The staggered firing of these neurons suggests an inherent functional role for spike timing.

Importantly, the goal-directed sampling of our model has an inherent generalization mechanism, allowing the generation of imagined movement trajectories that traverse parts of the space never encountered during training, as observed in the rodent (Gupta et al., 2010). During learning of $\mathbf{W}$ a large part of the 2D space was not accessible, see Fig. 1c. But yet the sampled trajectories can pass through it, as shown in the plots for high noise levels in Fig. 1d.

Finally, we include in the model the contribution of hippocampal object cells (Manns and Eichenbaum, 2009) and barrier cells (Rivard et al., 2004), whose firing encodes distance and direction of an obstacle or wall. We model the contribution of these cells as repulsive forces from obstacles and walls during the stochastic action selection. As a result, the sampled goal-directed trajectories avoid objects and walls (Fig. 1f). This model provides a potential explanation for the puzzling finding that hippocampal replay trajectories show flexible rerouting around novel barriers, without changes (remapping) in place fields (Widloski and Foster, 2022). In fact, the goal-directed trajectories that the model generates look remarkable similar to the recorded ones in that study, see e.g. their Figure 3a.

To sum up, we have shown that stochastic sampling from a standard model for a cognitive map of 2D space supports goal-directed imagination of trajectories from start to goal locations, capturing key qualitative patterns of rodent hippocampal replay in the same setting, including generalization to un-explored parts of the space, and flexible adjustment to novel contingencies. In the following sections we will examine whether sampling from cognitive maps can also reproduce goal-directed imagination and planning in non-spatial environments.

2.2 Sampling from learnt cognitive maps for abstract concept spaces enables generic problem solving through goal-directed imagination

Experimental data from neuroscience and cognitive science suggest that the brain—especially the human brain—employs cognitive maps not only for spatial navigation, but also for navigation in more abstract concept spaces, where spatial locations are replaced for example by images, or by combinations of learnt ranks of an item in different linear orders (Behrens et al., 2018; Bottini and Doeller, 2020; Tavares et al., 2015; Buzsáki and Moser, 2013; Iyer et al., 2024; Xiao et al., 2025). We show here that the principles that enable according to the preceding section goal-directed imagination for 2D spatial cognitive maps generalize to cognitive maps for abstract concept spaces, and enable there generic problem solving (see Fig. 2a for an illustration of a non-spatial problem solving task). Since the role of the grid cell system in the human brain for the generation of cognitive maps for abstract concept spaces is not clear, we employ here a generic model for learning cognitive maps, the Cognitive Map Learner (CML) from (Stöckl et al., 2024) that is based on the commonly accepted principle of predictive coding. As for the cognitive maps that are created through the rodent grid cell system, the key principle is to learn a forward model that relates current actions to the neural codes for the resulting states or observations. This is achieved in the CML model by learning embeddings $\mathbf{Q}$ and $\mathbf{V}$ of observations and actions into a common high-D space, see Fig. 2b. In biological terms, this high-D space models the space of neural activity patterns of neurons that encode places within the cognitive map. Hence this coding space has in the brain a dimension that is equal to the number of neurons in that area. However, a much smaller dimension – of around 1000 – suffices for the tasks considered in this study. Similarly, it suffices to employ linear maps $\mathbf{Q}$ and $\mathbf{V}$ that embed sensory inputs and efferent copies of action commands into this high-D coding space. Furthermore, simple local synaptic plasticity rules suffice for learning these embeddings in a self-supervised manner during exploration, see Fig. 2c. These local plasticity rules (see Equation 13 and Equation 14 in Section

4) approximate gradient descent for learning state predictions according to (Stöckl et al., 2024). If nonlinear embeddings would be more useful, linear readouts from fixed nonlinear circuits, as proposed by (Maass et al., 2002), could be trained equally well.

The learnt embeddings $\mathbf{Q}$ and $\mathbf{V}$ provide a forward model that predicts the next state in a straighforward manner:

$$\mathbf{Q}\mathbf{o}_{t+1} \approx \mathbf{Q}\mathbf{o}_t + \mathbf{V}\mathbf{a}_t, \tag{1}$$

where $\mathbf{o}_t$ is the observation at time step t and $\mathbf{a}_t$ is the action carried out at time step t. Furthermore, the CML can learn during exploration through Hebbian learning an inverse model in the form of a matrix $\mathbf{W}$ that maps state differences $\mathbf{s}_{t+1} - \mathbf{s}_t$ onto actions a_t that cause them, see Fig. 2d. This procedure is analogous to the learning of the $\mathbf{W}$ matrix used in the previous section for goal-directed spatial navigation. After learning $\mathbf{Q}$, $\mathbf{V}$, and $\mathbf{W}$, the CML can be tasked with reaching an arbitrary goal state $\mathbf{s}^* = \mathbf{Q}\mathbf{o}^*$ that results from a desired observation $\mathbf{o}^*$; for example, a goal location for spatial navigation or, more generally, a problem to solve, see Fig. 2a and the examples provided in (Stöckl et al., 2024). The learned matrix $\mathbf{W}$ can be used to select actions that are likely to be useful for reaching this goal state $\mathbf{s}^*$. More precisely, $\mathbf{W}$ assigns to any possible action $\mathbf{a}_t$ a utility that scores its estimated usefulness (or value) for reaching $\mathbf{s}^*$ from $\mathbf{s}_t$. This strategy relies on the hypothesis that the geometry of the learnt cognitive map provides a sense of direction when choosing the next action. Note that $\mathbf{W}$ was trained to map local state differences $\mathbf{s}_{t+1} - \mathbf{s}_t$ onto actions a_t which caused them, but is applied here for choosing imaginary actions to reach a distant goal $\mathbf{s}^*$ with a much larger state differences $\mathbf{s}^* - \mathbf{s}_t$. Linearity of the inverse model $\mathbf{s}^*$ supports this generalization. The utility estimates of the CML are analogous to value estimates in RL (Sutton and Barto, 2018; Sutton, 1992). However, the learned matrix $\mathbf{W}$ provides a universal value function that assigns values for reaching any possible state, and these estimates do not dependent on an a-priori chosen policy as in RL. However, similar to RL, in the CML model it is necessary to multiply the utility estimate of an action by an *affordance* factor that vetoes actions that cannot be executed in the current state (or to increase or decrease in more complex task environments the value of actions that are easy or challenging to execute in the current state). We denote the product of the current utility of an action (for reaching some goal $\mathbf{s}^*$) with this affordance factor as its current *eligibility*. At each time point, goal-directed navigation can be realized by simply selecting the action $\mathbf{a}_t$ with the highest eligibility. After action execution, the CML makes a transition to a new state $\mathbf{s}_{t+1}$, which it represents by the embedding of the next observation $\mathbf{o}_{t+1}$ that it receives ($\mathbf{s}_{t+1} = \mathbf{Q}\mathbf{o}_{t+1}$). From there it can iterate its action selection process, until it reaches the goal.

In order to apply this model to the generation of imagined action sequences where no observations are received, we replace observations by predictions of resulting next states, see Fig. 2e. Specifically, after generating a virtual action command $\mathbf{a}_t$, the model does not receive any observation $\mathbf{o}_{t+1}$, and replaces it by an internally generated prediction $\hat{\mathbf{s}}_{t+1} = \mathbf{s}_t + \mathbf{V}\mathbf{a}_t$. It continues from there to generate the next virtual action. Using this mechanism permits generating, for any given initial state $\mathbf{s}_t$ and any given goal state $\mathbf{s}^*$, an imagined trajectory $\mathbf{s}_t$, $\hat{\mathbf{s}}_{t+1}$, $\hat{\mathbf{s}}_{t+2}$, ... in the direction of the goal $\mathbf{s}^*$, without receiving any observation. This becomes a probabilistic generative model if one adds noise in this imaginary trajectory generation process. Specifically, we have added random values from a Gaussian distribution to the current estimate of the eligibility of each action. We call this probabilistic generative model a *Generative Cognitive Map Learner (GCML)*. We refer to Section B and Fig. S1 of the Supplement for the detailed algorithm and a schematics of neural network implementations.

As a first example for the functional capability of the GCML we show that it produces a menu of possible solutions for solving a problem. We use here a generic formalization of problem solving that is widely adopted in AI and cognitive science: As task to find a short path from a start to a goal node in an abstract graph – whose nodes represent problem solving states and whose edges represent possible actions (Russell and Norvig, 2016; Newell et al., 1972; Hills et al., 2015; Geffner and Bonet, 2013). Consider the generic random graph with 32 nodes shown in Fig. 3a. Finding a short path from start to goal in this and other graphs of non-trivial size is a challenging computational task. Whereas the CML generates according to (Stöckl et al., 2024) a single approximation to this shortest path problem, the GCML provides a selection of reasonable heuristic solutions. These take here the form of various paths from start to goal whose length is close to minimal. Such a selection of possible solutions is useful if there are other criteria besides the length which make a solution attractive. Mathematically, one can describe the output of the GCML as heuristic online solution to the well-known k-shortest path problem (Eppstein, 1998), where the objective is to find for some given k the k shortest paths from a given start to a given goal. As shown in Fig. 3c, most trajectories generated by the GCML

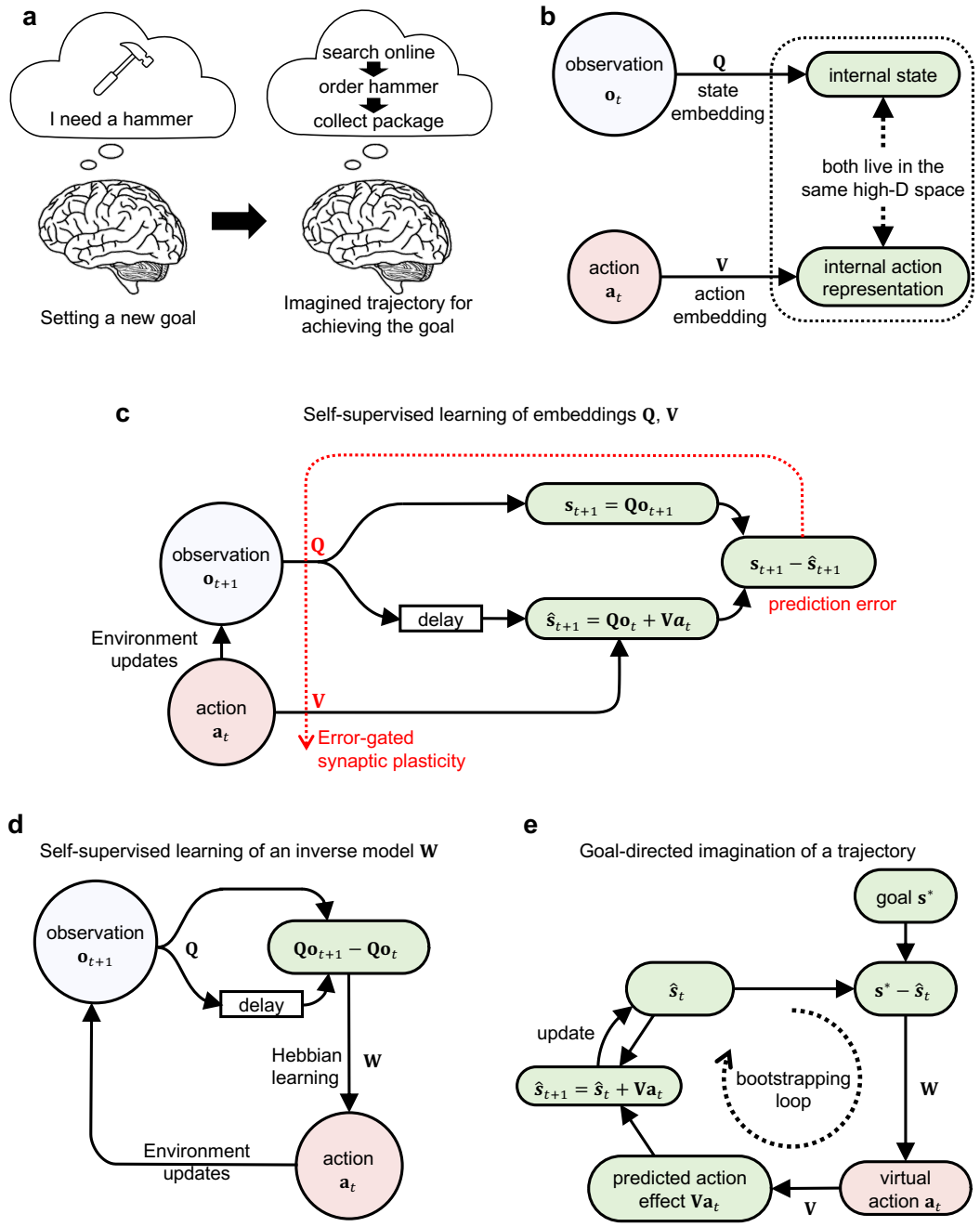

Fig. 2: Architecture and learning process of the generative cognitive map learner (GCML). (a) Illustration of a non-spatial problem which brains are able to solve through goal-directed imagination: A plan can be imagined for reaching any given goal, such as getting a hammer. (b) The GCML employs the same cognitive map learning strategy as the CML model of (Stöckl et al., 2024): The cognitive map arises by learning embeddings of observations and (efferent copies of) actions into a common high-D state space so that the geometrical structure of the resulting cognitive map (embeddings) captures the relations between observations and actions. These embeddings $\mathbf{Q}$ and $\mathbf{V}$ can be chosen to be linear for the tasks that we consider, and can be learned by simple local rules for synaptic plasticity (c) The learning of the cognitive map, i.e., of the embeddings $\mathbf{Q}$ and $\mathbf{V}$, is driven by a prediction learning goal: Predicting the embedding of the next observation (i.e., the next state) in terms of the current observation and action according to Equ. 1. (d) Simultaneously an inverse model in the form of a matrix $\mathbf{W}$ can be learnt that maps state differences onto actions which caused them. Note that if the learnt embedding $\mathbf{V}$ (the forward model) is linear and if Equ. 1 is satisfied, the existence of a linear inverse model is guaranteed. (e) To generate imagined trajectories toward a target state s^*, the estimated current state $\hat{s}_t$ is iteratively updated. However, importantly, no actions are carried out. Rather, predictions $\hat{\mathbf{s}}_{t+1}$ of resulting next observations (states) replace actual feedback from the environment. The next action is virtually selected for the next predicted state (bootstrapping). The difference between a goal state $\mathbf{s}^*$ and the estimated current state $\hat{\mathbf{s}}_t$ drives action selection through the learned matrix $\mathbf{W}$ like for real action selection, thereby enabling goal-directed imagination.

in a low-noise condition (noise = 0.15) have the shortest possible length. Higher noise levels (noise = 0.25) yield a larger diversity of solutions, some of which are slightly longer paths (Fig. 3d). But their length is still clustered around the optimum. Remarkably is that they still reach the goal, in spite of the higher level of noise. Even if a bad move is initially selected, it is later compensated by moving on average in the direction of the goal. In other words, the GCML has a self-correcting homing-in heuristic to the given goal.

As illustration for a functional use of having diverse heuristic solutions to a problem we consider in Fig. 3e a case where specific rewards (blue) or losses (red) are subsequently assigned to the nodes of the graph. These could encode preferences or disadvantages of using specific nodes in a solution paths, such as avoiding on a trip a transfer at an airport where a thunderstorm is reported. We show in Fig. 3f that higher noise levels of the GCML, corresponding intuitively to more fantasy in problem solving, provide solution paths that collect a larger number of rewards, In this case we let the GCML generate for each noise level 40 trajectories, and select for each noise level the trajectory with the largest sum of rewards.

The GCML provides a neural network model for problem solving that complements previously studied ones. With the Boltzmann Machine, or via neural sampling from a spiking neural network, one had already be able to solve problems in a less flexible manner, where all constraints and desiderata are programmed into the synaptic weights of the network (Buesing et al., 2011; Habenschuss et al., 2013; Jonke et al., 2016). In contrast, the problem solving goal can be communicated to a GCML through synaptic input to the underlying neural network (see Fig. S1). This more flexible method is consistent with numerous experimental data from the brain, where current goals have been decoded from the current firing activity, see e.g. (Zutshi et al., 2025). In addition, in contrast to these preceding models, the GCML generates solution strategies in a clear step-wise manner that can readily be adopted for execution. In addition, the GCML can also solve problems in compositional domains that were out of reach for the earlier models for problem solving by neural networks.

From a biological perspective, the mechanism used by the GCML – the stochastic generation of multiple paths and their subsequent selection based upon expected reward – could be associated with the neural circuit formed by the hippocampus and the ventral striatum. Various studies suggest that this circuit supports goal-directed navigational planning and "vicarious trial-and-error", with the hippocampus generating candidate future trajectories and the ventral striatum assessing the reward associated with those trajectories (Redish, 2016; Van Der Meer et al., 2012; Pezzulo et al., 2014; Pennartz et al., 2011).

2.3 Sampling from cognitive maps with compositional structure enables strong generalization of problem solving

Humans are able to imagine plans to achieve truly novel goals, such as wanting to sit together with a goat on the Schafberg mountain, once they receive a verbal (or otherwise symbolic, i.e., compositional) description of such goal. In terms of cognitive maps, these goals correspond to states that were never encountered before during exploration, hence the cognitive map structure around them could not be shaped by direct experience. We had already encountered in Fig. 1c a case where learned relational information from the cognitive map was instantly generalized beyond prior experience: never encountered states (locations) and path segments (actions) between states were automatically integrated into imagined paths. This was made possible by the inherently compositional, i.e., locally repetitive, structure of 2D space.

We are addressing here the question to what extent this strong generalization can also be applied to more complex compositional structures, such as those that are handled by the human brain. Therefore we considered a benchmark task for problem solving in a compositional domain that was recently used in a magnetoencephalography (MEG) study of compositional planning processes in the human brain (Schwartenbeck et al., 2023). Participants were asked to decide whether a given silhouette (like the one shown in black in Fig. 4a) could be decomposed using the given set of building blocks, as shown in Fig. 4b. Their recordings from the human brain showed that during planning, participants considered the possible building blocks that could be removed from the silhouette in a sequential manner, starting with the most obvious candidates. Furthermore, the authors suggested that replay sequences supported a hypothesis testing process for silhouette decomposition. This compositional task domain appears to be on first sight computationally much easier than natural language processing, but it is actually NP-hard (Moore and Robson, 2001), even for silhouettes that are just composed of the first 4 building blocks shown in Fig. 4b. This implies that there is no known determinitic

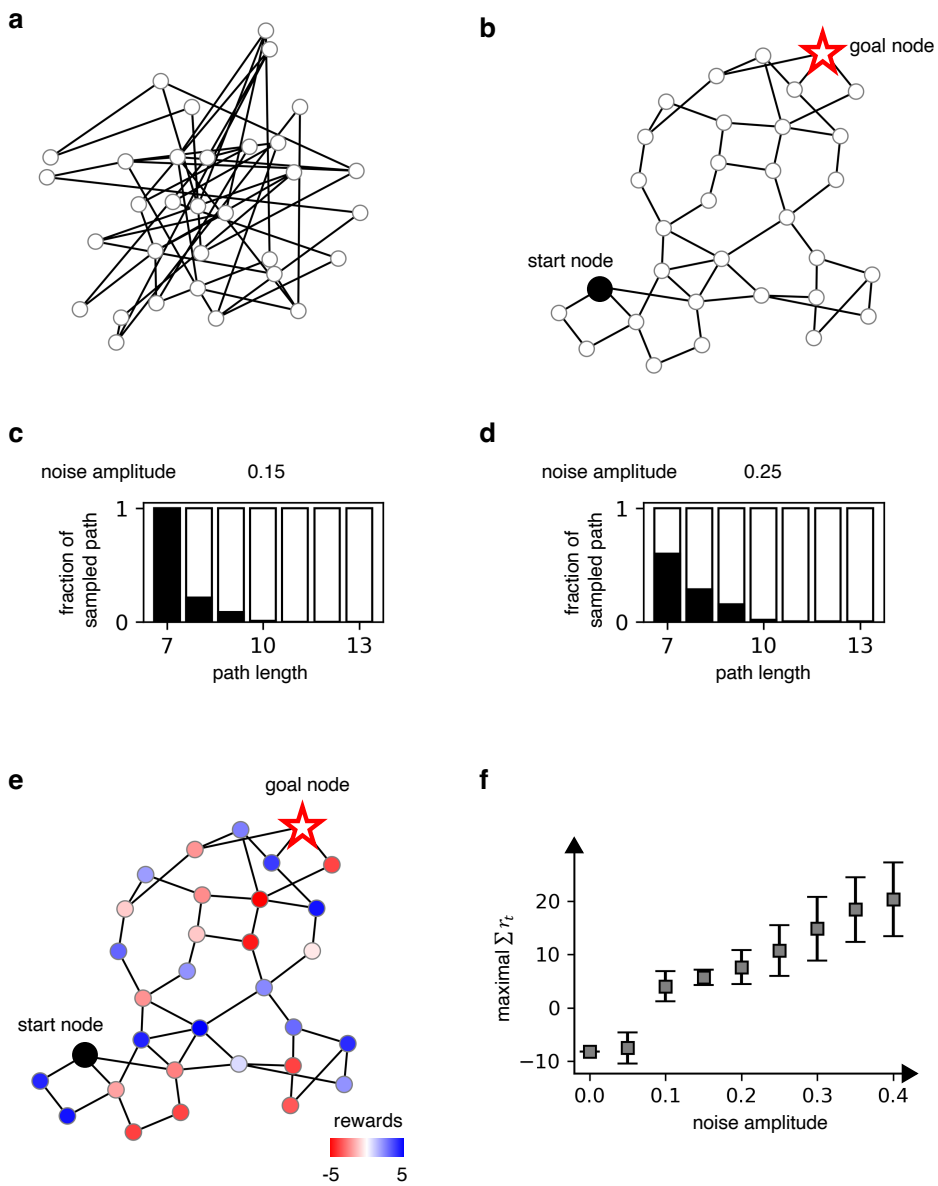

Fig. 3: **Demonstration of generic problem solving capabilities of the GCML.** (a) A random graph with 32 nodes, which can be viewed as abstract model for a problem solving environment. (b) 2D projection (via t-sne) of the cognitive map that the CML creates after exploring this graph with the help of its learned embeddings of observations and actions. The geometry of this graph representation supports a simple geometric heuristic strategy for selecting actions (edges) from any given node in order to reach any distal goal node: choose an edge that points into the direction of the goal. (c) and (d) show for two levels of noise the length statistics of paths that are generated by the GCML for an arbitrarily fixed choice of start and goal node (indicated in panel b). The lengths of paths are in both cases clustered around the minimum. The lower noise level in panel c provides a heuristic solution to the k-shortest paths problem. (e) and (f) Illustration of a functional benefit of generating a larger diversity of solution paths (with a higher level of noise in the GCML). Panel e shows a possible adhoc assignment of rewards and losses to individual nodes. If the goal is to generate a path with a maximal sum of rewards then a higher level of noise is beneficial, as shown in panel f. For each noise level 40 trajectories were generated by the GCML, and their maximal sum of rewards is indicated on the y-axis. The error bars indicate the standard deviation over 100 times repeated experiments with different random seeds. The starting node and goal node are the same as in b and e.

algorithm that can solve this decomposition task without using computational resources that grow exponentially with the size of the silhouette.

We show here that a GCML can solve through goal-directed imagination also these quite demanding compositional tasks. A key step is that one makes the embedding $\mathbf{Q}$ of observations (here, 2D silhouettes) into the cognitive map compositional. We use the same set of building blocks that was used in the experiments of (Schwartenbeck et al., 2023), except that we remove one of the building blocks (an atomic square tile) that would make every silhouette decomposable. Observation of a silhouette is encoded by a binary vector that has a 1 at each pixel position in a 2D grid space that corresponds is black in the silhouette. The simplest possible compositional embedding $\mathbf{Q}$ of silhouettes is the identity map over these binary vectors. This provides already decent performance of the GCML. However, its performance increases substantially, and its cognitive map is endowed with a better sense of direction if the embedding $\mathbf{Q}$ slightly departs from the identity map and assigns greater weight to black pixels that are surrounded in a silhouette by white pixels, see Section 4.4 for details. This embedding increases the utility of removing building blocks that cover black pixels which protrude from a silhouette. This heuristic is beneficial since the choice of these building blocks is more constrained and therefore less error-prone. It also creates a cognitive map of the compositional domain that supports the same geometric heuristic for action selection as in the previously discussed GCML applications: Choose an edge from the current node that best points into the direction of the goal, see Fig. 4d. Further details and the full algorithm are provided in Methods and part C of the Supplement.

The embedding $\mathbf{V}$ of actions (here: removal of a particular building block at a particular position of the silhouette) and the inverse model $\mathbf{W}$ learned in a self-supervised manner from successful decomposition examples (see Fig. 4c). More precisely, the action embeddings $\mathbf{V}$ learned to predict the remaining silhouette after removal of the corresponding building block, see Equ. 1 and Methods for details. Decompositions that could be completed in 5 steps were used for learning. Examples are shown in Fig. 4c. The building blocks that have already been removed are indicated there in white with a black outline. This visual representation indicates the temporal decomposition process quite well, but is slightly deceptive from a mathematical perspective, because removed tiles are no longer represented in intermediate or final silhouettes. In particular, the goal state is always the same one: an empty silhouette. Hence in this case of goal-directed sampling the goal state is familiar, but the start state (the given silhouette) is in general novel.

The learnt cognitive map is for this task domain more difficult to visualize in 2D because of the large number of actions that can be applied to each silhouette (state). But one can see in Fig. 4d and Fig. S2 for the case of simple initial silhouettes that the 2D projection of the cognitive map still provides a "sense of direction" that favors actions that go into the direction of the target state (empty silhouette, indicated by a red star). The four paths from start to goal that are plotted in Fig. 4d with solid, dotted, dashed and dash-dotted line segments correspond to four trajectories that solve the problem. Their intermediate states are indicated at the right of Fig. 4c. Note that all of these correct trajectories result from actions that move into the direction of the goal in this cognitive map. The top part of the left plot in the panel depicts three silhouettes that result from removing a "wrong" building block from the starting silhouette that does not lead to a full decomposition of the initial silhouette. Note that they lie relative to the starting point on the opposite direction of the goal. Another dead end configuration, that arises if one does not move towards the goal node in the second removal step of the paths indicated by solid and dotted lines, is shown at the bottom. Additional examples of cognitive maps and generated trajectories to decompose silhouettes are shown in Figs. S2.

Remarkably, after learning a cognitive map from decompositions of silhouette that are composed of 5 building blocks (Fig. 4c), the GCML is also able to decompose silhouettes consisting of 8 building blocks (Fig. 4e). Fig. 4f shows for this more demanding case, for various numbers of samples generated by the GCML, that it outperforms a baseline algorithm ("Random") that removes building blocks randomly and the CML model. Furthermore, the GCML shows a remarkable, near perfect success rate, even with few samples.

Finally, we show in Fig. 5 that goal-directed sampling from the same cognitive map as discussed before can also solve more general types of compositional problems. One example are problems where the goal is not the empty silhouette, but a partial decomposition indicated as orange goal silhouette in Fig. 5a. Fig. 5c shows that the GCML can also solve this task, for silhouettes consisting of 6 building blocks that did not occur during learning, through goal-directed sampling from the cognitive map.

Another example are problems where building blocks not only have to be removed, but also have to be added, in order to arrive at the goal silhouette indicated in orange in Fig. 5b. No new learning is

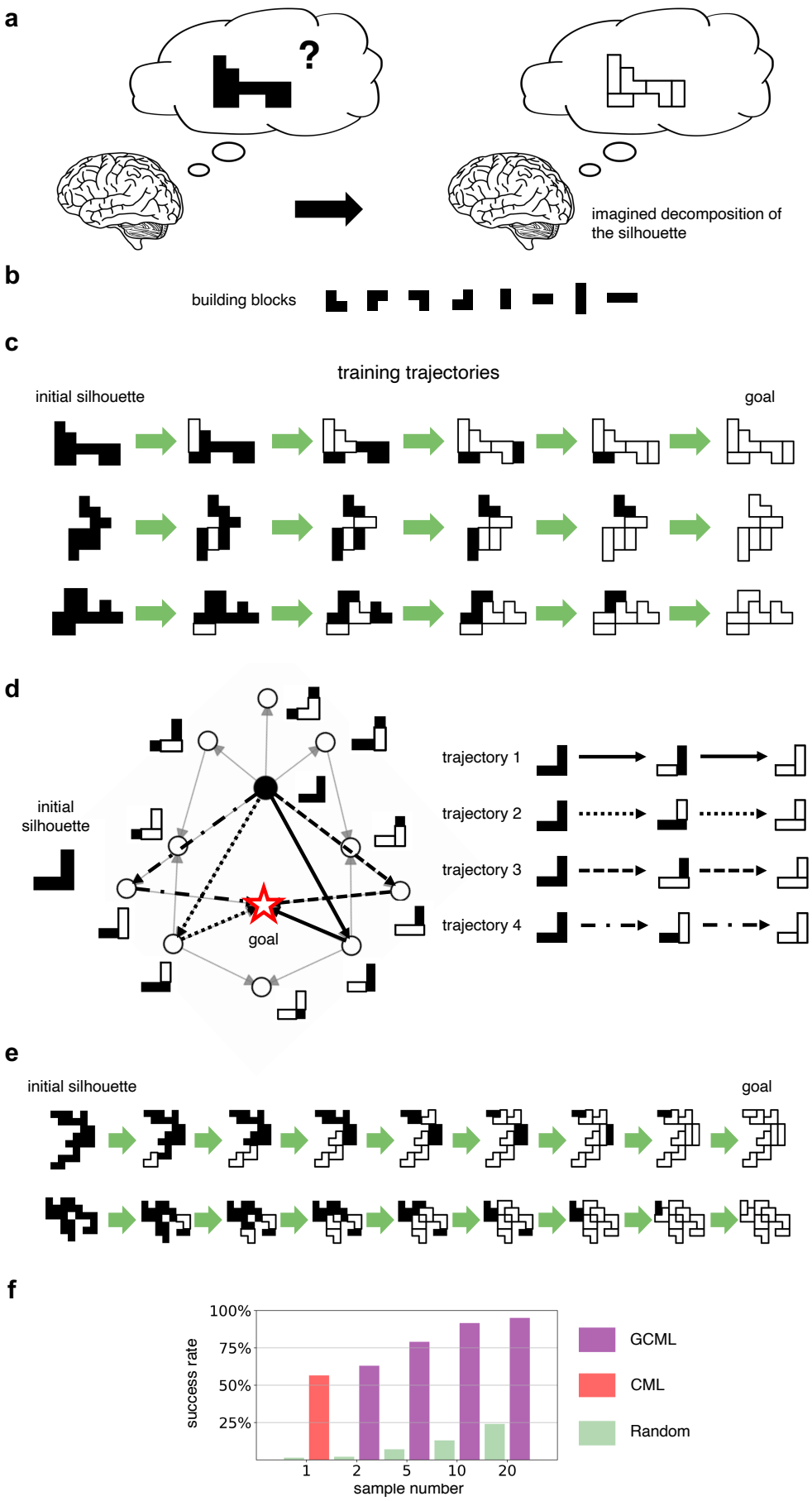
a
?
imagined decomposition of the silhouette
b
building blocks
c
training trajectories
initial silhouette
goal
d
initial silhouette
goal
trajectory 1
trajectory 2
trajectory 3
trajectory 4
e
initial silhouette
goal
f
100%
75%
50%
25%
success rate
1
2
5
10
20
sample number
GCML
CML
Random

Fig. 4 *(previous page)*: **Application of the GCML to goal-directed sampling for a compositional benchmark task.** (a, b) Illustration of the task: Find a decomposition of the silhouette into the building blocks shown in panel b, provided there exists such a decomposition. (c) Three example trajectories that were used during learning of the GCML. The five BBs used to generate each silhouette were removed in a randomly shuffled order. (d) 2D t-SNE projection of the resulting cognitive map for the embedding Q of silhouettes with affordance factors. (e) Two example trajectories generated by the GCML for decomposing silhouettes containing 8 BBs. (f) Test of the ability of the GCML to generate decompositions for decomposable silhouettes consisting of 8 building blocks, after training on decomposition of silhouettes with 5 building blocks. Hence none of the test silhouettes had been encountered by the GCML before. See Fig. S2 for additional examples.

needed for that: The action to add a building block (at a particular place of the current silhouette) is mapped to a vector that goes into the opposite direction of the one for removing this building block. Fig. 5d shows that the GCML can solve also these problems quite well, for tasks where neither the initial silhouette nor the target silhouette occurred during learning of the cognitive map.

To sum up, we have shown here that endowing the cognitive map with compositional structure permits addressing challenging problem solving tasks in compositional domains through goal-directed sampling from a cognitive map. In particular, required solution paths had to go through never encountered states. The compositional nature of the map, which reflects the relations between building blocks and their possible combinations, permits generalizing the goal-directed imagination of the GCML not only to decomposition of silhouettes that were never encountered during learning, but also to new variants of this decomposition task: partial decomposition and combining decomposition with composition of a target silhouette.

3 Discussion

We have examined three prominent facets of brain computations, stochastic neural responses, cognitive maps, and compositional computing, from the algorithmic and functional perspective. We found that together these three facets provide a core capacity of human intelligence: Being able to solve problems, in particular problems that were never encountered before. We have examined a simple architecture that integrates these three tools and can learn in a self-supervised learning with local synaptic plasticity rules. Hence this architecture is not only of interest for modeling and understanding brain function, but also provides a realistic next target for neuromorphic engineering: It shows that powerful problem solving capabilities can be installed through self-supervised on-chip learning in energy-efficient neuromorphic hardware.

On a more abstract algorithmic level this architecture provides an intriguing novel paradigm for using the inherent noise of biological and physical computing systems as a computational resource (Maass, 2014). A new feature is that this noise enables goal-directed sampling, for a goal that is provided in the form of a synaptic input, rather than just sampling from a fixed probability distribution that has been programmed into the weights of synaptic connections. This endows the neural sampling method with a brain-like flexibility in adjusting to new goals or contingencies, and to solve new problems that were never encountered before. From a more general theoretical perspective our model for goal-directed sampling is related to previous models for probabilistic inference in the brain for perception and action selection through sampling (Pouget et al., 2003; Fiser et al., 2010; Ujfalussy and Orbán, 2022; Parr et al., 2022; Ma et al., 2006; Friston, 2010). But in contrast to this preceding work it provides a neural network paradigm for sampling from a marginal distribution, relative to problem specifications that are defined through the current synaptic input to the network. A theoretical basis for that had been provided in (Habenschuss et al., 2013). It was shown there that a stationary distribution exists for this sampling from a marginal distribution even if the synaptic connections of the network are not symmetric, which is always the case if one has synaptic connections between excitatory and inhibitory neurons. It also exists if the neurons are spiking, and exhibit a brain-like diversity of firing properties.

Sampling from a marginal distribution provides an alternative to RL for generating goal-directed behavior. Models based on RL often require methods from machine learning for training, such as BP and BPTT, whose biological viability is debated. Furthermore, except for methods based on successor representations, RL approaches require retraining for each new goal that is to be reached. Apart from this versatility advantage of cognitive map based methods, there are also some recent experimental

Fig. 5: Application of the GCML for generating possible solutions to more demanding versions of the silhouette decomposition task. (a) Decomposition to a specific non-empty silhouette (containing 2 building blocks, as shown in orange) from an initial silhouette that contains 6 building blocks. Successful decompositions by the GCML are shown for two examples of start- and goal-silhouettes. (b) Examples of trajectories generated by the GCML with the same cognitive map for a more demanding variant of the silhouette decomposition task: After decomposition, building blocks have to be added to form a goal silhouette. Like the initial silhouette, it contains 6 building blocks and hence did not occur during learning. Added building blocks are shown in gray for two example imagined paths to the goal generated by the GCML. (c, d) Evaluation of the performance of the GCML for both types of extensions of the silhouette decomposition task.

data which suggest that dopamine signals support learning of various types of predictions in the brain (Jeong et al., 2022; Gershman et al., 2024; Kahnt and Schoenbaum, 2025), not only reward predictions. Hence they might also support learning of cognitive maps by minimizing prediction errors for sensory inputs, which provides the basis for the CML and GCML model. Altogether, the diversity of data on dopamine signals suggests that the brain could employ not just one but a multitude of mechanisms and strategies for learning goal-directed behavior, as suggested by (Gershman et al., 2024).

We have validated our approach in three very different application scenarios. The first one, imagination of possible paths for reaching a given goal in a 2D environment, connects our model to the original discovery of this capability of the brain: The recordings of (Pfeiffer and Foster, 2013; Widloski and Foster, 2022) from the rodent brain, which were recently complemented by recordings that clarify the essential role of the grid cell systems for the generation of imagined paths (Vollan et al., 2025). We shown in Fig. 1 that these experimental data can be reproduced with a standard model for the 2D spatial map that is provided by the grid cell system of the rodent (Hafting et al., 2005; McNaughton et al., 2006; Dong and Fiete, 2024). We have shown that an inverse model that maps differences between grid states to movement commands that reduce this difference can be learnt by a simple local rule for synaptic plasticity. By adding noise to this action selection mechanism this cognitive map becomes then a generative model that generates goal-directed movement plans that qualitatively resemble those that were decoded from the rodent brain (Pfeiffer and Foster, 2013), even rerouting around novel barriers as demonstrated in the neural recordings of (Widloski and Foster, 2022). Furthermore, the samples generated by the model generalize to never experienced trajectories, as observed in rodents (Gupta et al., 2010; Vollan et al., 2025).

Numerous more recent data suggest that some forms of cognitive maps are employed by the human brain not only for 2D spatial navigation, but also for navigation in abstract conceptual spaces, i.e.. for abstract problem solving (Behrens et al., 2018; Bottini and Doeller, 2020; Tavares et al., 2015; Xiao et al., 2025; Buzsáki and Moser, 2013; Iyer et al., 2024; FeldmanHall et al., 2025). Since it is unknown to what extent these cognitive maps are also created by grid cells, we use a more abstract generic neural network method for learning a cognitive map in a self-supervised manner through autonomous exploration, the *cognitive map learner (CML)* proposed in (Stöckl et al., 2024). We have presented in Fig. 2 a natural extension of this model– the *Generative Cognitive Map Learner (GCML)* – that stochastically samples possible trajectories to reach both known and novel goals in the cognitive map. We have shown in Fig. 3 that the GCML generates in an abstract random graph not only a single short path to a given goal, like the method of (Stöckl et al., 2024), but a menu of many different short paths to this goal. In fact, it provides heuristic online solutions to a computationally quite difficult task, the k-shortest path problem (Eppstein, 1998). Hence a downstream network can select from this menu of good solutions that which fit additional constraints, such as picking up a maximal number of rewards on the way (see Fig. 3e and f).

We show in the third application scenario that the goal-directed sampling capability of the GCML can also be applied in compositional task domains, where a virtually infinite number of states –most of them never visited before– can be defined by new combinations of a fixed set of components, like new sentences that are formed from a fixed vocabulary of words. We have addressed there the compositional domain that has recently been examined through recordings from the human brain in (Schwartenbeck et al., 2023): 2D silhouettes that are composed from a small set of building blocks ("tiles"). We have shown in Fig. 4 that the GCML can solve the same task as the human subjects in this study: Deciding whether a new 2D silhouette, that was never encountered before, can be decomposed into the given set of building blocks. This task is actually very hard from the perspective of computational complexity theory, even NP-hard, meaning that no deterministic algorithmic can perform well on this task within reasonable (polynomial) bounds on computational resources. In Fig. 5 we have shown that the GCML can also solve functionally important variations of this compositional task, such as building a given 2D silhoutte from a given repertoire of building blocks.

Taken together, we have shown that sampling from cognitive maps provides a surprisingly powerful and versatile heuristic method to solve computationally challenging problems across both spatial and non-spatial domains, using an algorithmic form of goal-directed imagination. Furthermore, this method only requires minimal computational resources can be installed through on-chip learning in neuromorphic hardware.

From a biological perspective, our approach is supported by experimental evidence showing a dual role of (hippocampal) cognitive maps, for learning states in an associative mode and supporting sequential activity in a predictive mode (Liu et al., 2023) – and provides a possible mechanistic explanation of this finding. Besides explaining previous findings, our perspective generates novel

empirical predictions that can be addressed in future studies. In particular, our model proposes that the brain learns together with each cognitive map also an inverse model that maps state differences to possible actions for reducing this state difference. Note, however, that an inverse model for a cognitive map could also be implemented in the brain in a somewhat different way, where the current and goal states are provided separately as inputs, instead of their difference. But as long as the inverse model is approximately linear, it supports local actions with "foresight", preferring actions that move into the direction of the goal. Future studies could align this proposal with empirical findings showing that goal and reward information shape trajectory representations and the geometry of cognitive maps in the hippocampus and cortex (Gauthier and Tank, 2018; Muhle-Karbe et al., 2023; Hok et al., 2007; Schuck et al., 2016; Crivelli-Decker et al., 2023; Nyberg et al., 2022; Zutshi et al., 2025; Bernardi et al., 2020).

The assumption that goal-directed behavior is driven by minimizing the discrepancy between current and goal states is shared with frameworks such as cybernetics (Rosenblueth et al., 1943; Wiener, 1961; Powers, 1973) and active inference (Parr et al., 2022). We have shown in Fig. 1f that this strategy also works in the presence of obstacles, where a direct path to the goal is not feasible. It suffices to integrate signals from object vector cells (Høydal et al., 2019) into action selection that indicate direction and distance from an obstacle. This provides a substantially more parsimonious model for planning 2D navigation in the presence of obstacles than the model of (Bakermans et al., 2025), where first the outer product of grid and object vector cells has to be created through an offline learning process. Our model is much simpler, and predicts that no new map has to be learnt when an obstacle is moved.

A key prediction of the GCML model is that noise in the selection of virtual actions or next states during imagination is directly related to the resulting diversity of action plans, and implicitly also on the difficulty of solving a given task. This could be tested experimentally. Furthermore, our model predicts that the speed of the generation of the next step of a multi-step solution plan depends only mildly on the current distance to the goal. The reason is that the next step is generated by a feedback circuit as indicated in Fig. 2e whose processing speed is independent from the distance to the goal. But the generation speed of the next step is predicted to grow with the number of action alternatives that are available, since this is likely to affect the speed of the underlying WTA computation for each action selection.

Finally, our model predicts that current goals can be decoded from brain areas that are involved in action selection, as already partially shown in (Zutshi et al., 2025). In contrast, RL models predict that the values of adjacent states are most relevant for action selection, and the identity of the current goal has no direct influence.

Altogether we have introduced the GCML as a new model for neural sampling, which supports –in contrast to preceding models – goal-directed sampling. Hence it drastically enhances the application range of neural sampling in AI. In particular, it supports the drive towards more energy-efficient AI solutions for planning and problem solving. The local learning mechanism of the GCML makes this approach suitable for self-supervised on-chip learning during exploration. Furthermore, the (virtual) action selection mechanism of the GCML is suitable for in-memory computing with memristor crossbars, yielding a minimal delay for action selection that is independent of the size of the environment and the number of options for action selection. Hence, the GCML paves the way for creating edge devices that can employ goal-directed imagination, often referred to as fantasy, in order to find solutions to difficult and novel tasks.

4 Methods

4.1 Details of our model for spatial goal-directed imagination in Section 2.1

The model shown in Fig. 1a consists of grid cells, place cells, and action basis neurons. All neurons were linear, with each neuron's activation equal to the sum of its inputs.

Our model for grid cells is based on the core idea that grid cells can perform ideal path integration based on a velocity input (Campbell et al., 2018; Sorscher et al., 2023). We regard grid-cell activity itself as the state space of our model, in which both the target state $\mathbf{s}^*$ and its estimated current state $\hat{\mathbf{s}}_t$ are expressed, see Fig. 1b. We approximate post-training grid-cell activity in their model analytically, by a deterministic function $\mathbf{x} \mapsto \mathbf{s} \in \mathbb{R}^{1000}$ that outputs the firing rates of 1000 grid cells at any location in a 2D arena. Each grid cell is initialized independently and modeled as the sum of three positive planar cosine gratings $r(\mathbf{x}) = \sum_{j=1}^{3}(1+\cos(\mathbf{k}_j \cdot \mathbf{x} + \phi_j))$, whose wave-vectors $\mathbf{k}_j$ are 60°

apart and share magnitude $|\mathbf{k}_j| = 2\pi/\lambda$. The grid scale was drawn once per cell from $\lambda \sim U(0.05, 8)$ and each phase independently from $\phi_j \sim U(0, 2\pi)$. To eliminate directional bias a global rotation $\theta_0 \sim U(0, 2\pi)$ was applied, giving $\arg(\mathbf{k}_j) = \theta_0 + j\pi/3$.

Under this setting, PCA applied to the grid-cell representations of the 10 % largest-scale (λ) grid cells revealed a nearly planar two-dimensional manifold (Fig. 1b). Because larger spatial periods generate progressively flatter interference patterns, this low-curvature embedding is well suited for goal-directed navigation.

Grid cell firing in the brain is updated by path integration of locomotion activity. A velocity signal is provided in our model by the activity of four action basis neurons, whose sum provides a continuous velocity vector. We constrained a fixed step size $\eta_a = 0.05$ by normalizing the length of the resulting velocity vector always to η_a.

To simulate continuous movement, time was discretized, with each time step representing 10 ms. The resultant movement $(\Delta x, \Delta y)$ during each time step is computed by the cumulative contributions of all four action basis neurons:

$$\Delta x = (\mathbf{a}_t)_{\text{right}} - (\mathbf{a}_t)_{\text{left}}, \quad \text{and} \quad \Delta y = (\mathbf{a}_t)_{\text{up}} - (\mathbf{a}_t)_{\text{down}}. \tag{2}$$

This provides the forward model for the cognitive map in this section.

As in (Sorscher et al., 2023), grid cell activity can predict place-cell activity. In our model the place fields are established by one-shot learning from grid cell activity. The incoming weight of each place cells is assigned in a single update using a simplified rule that captures the core idea of the LTP component of behavioural time-scale synaptic plasticity (BTSP) (Milstein et al., 2021; Wu and Maass, 2025).

During 500 s of random exploration, a plateau-potential gating signal occurs every 500 ms. At each event, one postsynaptic neuron i is randomly selected from the neuron pool of 1000 neurons, and its incoming weight vector $\mathbf{P}_i$ (row i of $\mathbf{P}$) is overwritten with the current pre-synaptic grid-cell activity pattern $\mathbf{s}_t$:

$$\mathbf{P}_i \leftarrow \mathbf{s}_t. \tag{3}$$

This centers the place field of neuron i at the location (x_t, y_t). Because the learning happens only at visited places, comprehensive exploration is essential; regions that are never visited cannot recruit corresponding place fields. We used grid cell activity, rather than place cell firing, for generating imagined trajectories, in particular for representing the current state of an imagined trajectory. Hence this imagination process was not affected by the lack of place cells for the unexplored region shown in Fig. 1c. Our motivation for leaving part of the 2D arena unexplored was to demonstrate that the unexplored region does not hinder grid cells from generating imagined trajectories that include segments in the unexplored region. But in order to be able to plot spiking activity of place cells during imagined trajectories in Fig. 1e we assumed that place cells had formed in the meantime also for the initially unexplored region.

An inverse model in the form of a weight matrix $\mathbf{W}$ learned in a self-supervised manner to infer actions from differences in grid cell states which caused these differences, using a simple Hebbian learning rule:

$$\Delta \mathbf{W}_{t+1} = \eta_w \cdot \mathbf{a}_t (\mathbf{s}_{t+1} - \mathbf{s}_t)^{\mathrm{T}}, \tag{4}$$

where η_w is the learning rate. We simulated here random exploration for 500 s, discretized by 10 ms within a 2D bounded environment ($[0, 4] \times [0, 4]$), excluding the lower-left quadrant ($x < 2$, $y < 2$) shown in Fig. 1c. The agent started within the permitted region and executed a continuous-time random walk at constant speed (0.5 unit/s) with Gaussian turn noise for . Invalid movements (entering forbidden zones or crossing boundaries) triggered re-sampling of the direction. We consistently use the matrix $\mathbf{W}$ as learned inverse model for a cognitive map throughout this paper.

We used this inverse model for (virtual) action selection during stochastic generation of an imagined trajectories to a given goal $\mathbf{s}^*$ as follows:

$$\mathbf{a}_t = \eta_a \left(\mathbf{W}(\mathbf{s}^* - \mathbf{s}_t) + \epsilon\right), \tag{5}$$

where ϵ is the noise that introduced stochasticity into the model, and $\eta_a = 0.05$ is the step size. Each dimension of ϵ was sampled from a Gaussian distribution $\epsilon_i \sim \mathcal{N}(0, \sigma_n)$, where σ_n represents the "noise level" depicted in Fig. 1d. We assumed four basis action neurons defining an action vector $\mathbf{a}_t = [(\mathbf{a}_t)_{\text{up}}, (\mathbf{a}_t)_{\text{down}}, (\mathbf{a}_t)_{\text{left}}, (\mathbf{a}_t)_{\text{right}}]$.

Spike generation, as illustrated in Fig. 1e, was modeled by interpreting the continuous activation of each place cell i as a firing probability $p_i(t)$ within a discrete time window of 10 ms. Spikes were then sampled from a Bernoulli distribution:

$$s_i(t) \sim \text{Bernoulli}\left(p_i(t)\right). \tag{6}$$

The firing probability at each time step was computed by applying the softmax function to the normalized activation $\ell_i(t)$ of all place cells:

$$p_i(t) = \frac{\exp\left(\ell_i(t)\right)}{\sum_{j=1}^{N} \exp\left(\ell_j(t)\right)}. \tag{7}$$

For generating imagined trajectories to a given goal in the presence of obstacles (see Fig. 1f) we augmented Equ. 5 and Equ. 2 with additional repulsive forces originating from each obstacle (x_o, y_o) towards the agent (x, y). The information encoded by the vector difference, $\boldsymbol{\delta} = (x_o - x, y_o - y)$, may be represented, in the brain, by object vector cells (Høydal et al., 2019). Each wall was discretized into three point-obstacles (the center and two endpoints), each exerting a repulsive force $\mathbf{f}_i$. The total repulsive force $\mathbf{F}$ was the vector sum of these:

$$\mathbf{f}_i = \frac{\boldsymbol{\delta}_i}{|\boldsymbol{\delta}_i|_2^2}, \quad \mathbf{F} = \sum_i \mathbf{f}_i. \tag{8}$$

The denominator $|\boldsymbol{\delta}_i|_2^2$ ensures the strength of each repulsive force diminishes with increasing obstacle distance. The total force $\mathbf{F}$, scaled by factor γ, modified the movement of the agent during one simulation time step (10 ms), from Equ. 2 to:

$$\Delta x' = \Delta x + \gamma \mathbf{F}_x, \quad \text{and} \quad \Delta y' = \Delta y + \gamma \mathbf{F}_y. \tag{9}$$

For all four navigation tasks in the presence of obstacles in Fig. 1f we set $\gamma = 0.2$ and noise scale $\sigma_n = 0.3$.

4.2 Mathematical description of the generative cognitive map learner (GCML) discussed in Section 2.2

The GCML uses the same cognitive maps as those that are learned in a self-supervised manner by the cognitive map learner (CML) presented in (Stöckl et al., 2024). Specifically, each observation $\mathbf{o}_t$ is mapped into a high-dimensional state space using a learned embedding matrix $\mathbf{Q} \in \mathbb{R}^{n_s \times n_o}$:

$$\mathbf{s}_t = \mathbf{Q}\mathbf{o}_t, \tag{10}$$

where n_s is the dimensionality of the state space, and n_o is the dimensionality of the observation $\mathbf{o}_t$. The target state (goal) is obtained as the embedding of a target observation:

$$\mathbf{s}^* = \mathbf{Q}\mathbf{o}^*. \tag{11}$$

Similarly, all possible actions $\mathbf{a}$ are embedded into the same high-dimensional space via a learned embedding matrix $\mathbf{V} \in \mathbb{R}^{n_s \times n_a}$, where n_a is the dimensionality of the action space. The objective of learning these embeddings is ensuring that the predicted next state $\hat{\mathbf{s}}_{t+1}$ is given by:

$$\hat{\mathbf{s}}_{t+1} = \mathbf{s}_t + \mathbf{V}\mathbf{a}_t. \tag{12}$$

Learning aims at ensuring that $\hat{\mathbf{s}}_{t+1}$ becomes a good approximation of the actual next state $\mathbf{s}_{t+1}$. The prediction error $(\mathbf{s}_{t+1} - \hat{\mathbf{s}}_{t+1})$, which the learning rules for $\mathbf{Q}$ and $\mathbf{V}$ aim to minimize (see Fig. 2c), has therefore the following form:

$$\boldsymbol{\Delta}\mathbf{V}_{t+1} = \eta_v \cdot (\mathbf{s}_{t+1} - \hat{\mathbf{s}}_{t+1})\mathbf{a}_t^{\mathrm{T}}, \tag{13}$$

$$\boldsymbol{\Delta}\mathbf{Q}_{t+1} = \eta_q \cdot (\hat{\mathbf{s}}_{t+1} - \mathbf{s}_{t+1})\mathbf{o}_t^{\mathrm{T}}, \tag{14}$$

where η_v and η_q are the learning rates. These plasticity rules, known as Delta rules, approximate gradient descent to minimize the prediction error. If this learning process is successful and $\hat{\mathbf{s}}_{t+1}$ becomes a good approximation of the actual next state $\mathbf{s}_{t+1}$, one has a basis for generating a sequence of virtual actions in a goal directed manner, without receiving observations, as indicated in Fig. 2e.

During trajectory generation, at step t, the delta vector represents the difference between the target state and the estimated current state:

$$\mathbf{\Delta}_t = \mathbf{s}^* - \hat{\mathbf{s}}_t, \tag{15}$$

where $\hat{\mathbf{s}}_t$ is the estimated current state, which equals the actual initial state $\mathbf{s}_0 = \mathbf{Q}\mathbf{o}_0$ when $t = 0$.

We introduced a learnable matrix $\mathbf{W} \in \mathbb{R}^{n_a \times n_s}$ to compute the utility of all actions from the delta vector simultaneously:

$$\mathbf{u}_t = \mathbf{W}\mathbf{\Delta}_t. \tag{16}$$

Each dimension stores the utility of a specific action, forming a utility vector $\mathbf{u}_t$ of length n_a. The estimation of action utility is referred to as Principle 2 in CML (Stöckl et al., 2024). The matrix $\mathbf{W}$ is learned following the same learning rule in Equ. 4.

In cases where not all actions are applicable, the utility values are masked by the learned affordance factor $\hat{\mathbf{g}}_t \in \mathbb{R}^{n_a}$, which aims to approximate the real affordance $\mathbf{g}_t$ defined by the constraints of the environment, a one-hot vector that indicates which actions are executable (1 for executable, 0 otherwise). This gating factor prevents the GCML from proposing infeasible actions. It is computed using the affordance gating matrix $\mathbf{G} \in \mathbb{R}^{n_a \times n_s}$ that receives the (imagined) state vector as its input. The matrix is updated during the learning process according to the following rule:

$$\mathbf{\Delta G} = \eta_g \cdot (\mathbf{g}_t - \mathbf{G}\mathbf{s}_t)\mathbf{s}_t^{\mathrm{T}}, \tag{17}$$

where η_g represents the learning rate.

The resulting vector, after applying the affordance gating, is termed the eligibility:

$$\mathbf{e}_t = \hat{\mathbf{g}}_t \odot (\mathbf{u}_t + \epsilon), \tag{18}$$

where ϵ is the noise that turns the CML into a probabilistic generative model. The injected noise enables the GCML to explore actions beyond the one with maximal eligibility. Hence, in contrast to CML's deterministic selection, GCML stochastically generates diverse trajectories.

Finally, the selected action $\mathbf{a}_t$ is the one with the highest eligibility:

$$\mathbf{a}_t = \mathrm{WTA}(\mathbf{e}_t), \tag{19}$$

where $\mathrm{WTA}(\cdot)$ denotes the winner-take-all operator, selecting the action with the highest eligibility from all possible actions.

After selecting a virtual action $\mathbf{a}_t$ at the estimated state $\hat{\mathbf{s}}_t$, the estimated next state, $t + 1$, is calculated using the bootstrapping process:

$$\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{V}\mathbf{a}_t. \tag{20}$$

By iteratively applying the trajectory generation process described earlier, a virtual goal-directed trajecvtory is generated.

By repeatedly generating trajectories in the presence of inherent noise in Equ. 18, different trajectories are generated. This results in a set of possible solutions from the initial state $\mathbf{s}_0$ to the goal $\mathbf{s}^*$.

A downstream network could select a specific one from this repertoire of imagined trajectories based on a given criterion. For example, when rewards are associated with specific states, the estimated rewards $\hat{r}_t$ of each state can be accumulated over an entire imagined trajectory, and the one with the largest sum of predicted rewards can be selected (see Fig. 3f).

4.3 Further details of the application of the GCML for goal-directed imagination for generic problem solving (Section 2.2)

The task setup was a random graph with 32 nodes, where the degree of each node was uniformly sampled between 2 and 5. Observations $\mathbf{o}_t$ corresponded to nodes, and actions $\mathbf{a}_t$ represented traversing edges in a specific direction. Hence each edge is modeled by two possible actions. Both $\mathbf{o}_t$ and $\mathbf{a}_t$

were encoded as unique one-hot vectors. The training dataset consisted of 200 randomly generated trajectories, each of length 32. During the planning process, GCML samples a set of trajectories from a given starting node to a target node.

The mappings $\mathbf{G}$ transform the estimated high-dimensional state $\mathbf{s}_t$ into corresponding affordance factors. The affordance gating vector is a binary vector indicating available actions, where ones represent feasible actions and zeros indicate restricted actions. A linear transformation is sufficient in this setting, as the vectors $\mathbf{s}_t$ are approximately orthonormal.

Exploration was regulated by adding a Gaussian noise vector $\boldsymbol{\epsilon}$—with each dimension sampled from $\mathcal{N}(0, \alpha_\epsilon)$—to a normalized unit-length utility vector. The noise amplitude α_ϵ was set to a default value of 0.1.

Initial values for model parameters were drawn from Gaussian distributions: $\mathbf{Q} \sim \mathcal{N}(0, 1)$ and $\mathbf{V}, \mathbf{W}, \mathbf{G} \sim \mathcal{N}(0, 0.1)$. Smaller initial values for $\mathbf{V}$ improved performance. Learning rates were set as $\eta_q = 0.1$ and $\eta_v = \eta_w = \eta_g = 0.01$. GCML demonstrates robustness to variations in learning rates, similar to the original CML, with stable training performance for η_g in the range $[0.001, 0.05]$.

Each node was assigned a reward drawn from a uniform distribution $\mathcal{U}(-5, 5)$.

To demonstrate that GCML provides a good approximation of the top-k shortest path problem (see Figs. reffig:fig3c-d), 40 trajectories were generated from a given starting node to a goal node under different noise levels in the GCML. The number of distinct trajectories for each possible trajectory length (i.e., the number of edges in the path from the starting node to the goal node) was computed using a breadth-first search algorithm. For each trajectory length, the proportion of distinct trajectories generated by a given noise scale GCML relative to the total number of possible distinct trajectories serves as a measure of how well GCML approximates the top-k shortest path algorithm.

4.4 Details of goal-directed sampling in compositional task domains, discussed in Section 2.3

The silhouette decomposition problem involves finding a set of building blocks (BBs, or "tiles") that completely cover a given silhouette without gaps or overlaps. This process is called the decomposition of the given silhouette. GCML generates candidate decompositions through goal-directed sampling, similar to the previously described navigation tasks. In the simplest version of the problem, the goal node $\mathbf{o}^*$ is defined as the empty silhouette, while the starting node $\mathbf{o}_0$ is the silhouette to be decomposed. An action is defined as removing a BB from a particular position in the remaining silhouette. Details of this process are provided below.

Building Blocks. The training dataset consisted of eight predefined building blocks (BBs). The first four BBs were categorized as *corners*, each with four possible orientations and a width of 1 pixel. The remaining four BBs were *bars*, including two longer types (1×3 and 3×1 pixels) and two shorter types (1×2 and 2×1 pixels). These eight BBs were derived from the human silhouette decomposition experiment of (Schwartenbeck et al., 2023). But we excluded the 1×1 BB since each presence makes each given silhouette decomposable. GCML was tasked with imagining possible decompositions of given silhouettes, each composed exactly of five BBs.

Generation of Training and Test Datasets. Each silhouette in the dataset was created by randomly selecting five BBs and placing them within a 10×10 image at random positions. In total, 18,000 training samples and 2,000 test samples were generated. Any silhouette appearing in the training dataset was excluded from the test dataset, regardless of its position. Each sample represented a decomposition trajectory, in which BBs were sequentially removed from an initial silhouette. The silhouette generation process followed these rules: all BBs had to remain within image boundaries without overlapping. An initial BB was placed randomly within the image with equal probability for each position. Subsequent BBs were added adjacent to the existing silhouette, with at least one pixel adjacency to increase task difficulty. Longer adjacencies increased the number of possible removal actions, further challenging the model. In the training dataset, the five BBs composing each silhouette were removed sequentially in random order to generate training trajectories.

Details of Actions. In the silhouette decomposition problem, each action $\mathbf{a}_t$ corresponds to removing a specific BB from a specific valid position within the 10 x 10 grid, resulting in 664 possible actions. Each action is uniquely encoded as a one-hot vector. This formulation aligns with human experiments (Schwartenbeck et al., 2023), where participants manipulate BBs using a mouse and

keyboard to cover the silhouette. Since moving the same BB to different positions requires distinct movements, each BB at a different position is treated as a separate action.

Details of Observations. In the silhouette decomposition problem, the observation $\mathbf{o}_t \in \mathbb{R}^{10\times 10}$ directly represents the silhouette image, rather than being encoded as a one-hot vector as in previous experiments.

High-Dimensional Embedding Details. The state representation $\mathbf{s}_t = \mathbf{Q}\mathbf{o}_t$ is defined in a 100-dimensional space, matching the dimensionality of the observation space. The matrix $\mathbf{Q}$ is assumed to be the identity matrix $\mathbf{I}$, ensuring a direct mapping between the high-dimensional state and the observation. Each element in $\mathbf{s}_t$ functions as a grandmother neuron, activating only when the corresponding pixel in the observation space is black.

Details of learning the cognitive map. The action embedding matrix $\mathbf{V}$ is learned according to Equ. 13. The learning rate of $\mathbf{V}$ was set to $\eta_v = 0.01$, with values between 0.0001 and 0.1 also demonstrating stable performance. The inverse model $\mathbf{W}$ is trained using Equation 4 with a learning rate $\eta_w = 1$, and the weights in $\mathbf{W}$ saturate at 1:

$$\mathbf{W}_{t+1} = \min\left(\mathbf{W}_t + \mathbf{a}_t(\mathbf{s}_t - \mathbf{s}_{t+1})^{\mathrm{T}}, 1\right). \tag{21}$$

As a result, $\mathbf{W}$ becomes a binary matrix. Both matrices $\mathbf{V}$ and $\mathbf{W}$ were initialized using Gaussian distributions $\mathcal{N}(0, 0.001)$. The method remained effective across a broad range of initial standard deviations, from 0.0001 to 1.

One can easily see that W outputs for a given state difference for each possible action (BB) the number of pixels in the overlap between the state difference and the BB, provided that the BB is contained in the state difference (otherwise it outputs 0).

Affordance Details. The affordance gating mechanism in the silhouette decomposition problem comprises two components. The first component, g_1, ensures the agent selects only BBs present in the silhouette. We assumed GCML could directly access g_1 from the environment during all imagined steps. This assumption aligns with biological experiments (Schwartenbeck et al., 2023), where subjects directly observe the screen while solving the silhouette decomposition problem. Thus, affordance information is assumed available at all steps, requiring no inference from $\hat{\mathbf{s}}_t$.

The second component, g_2, represents context-dependent affordance determined by the estimated state $\hat{\mathbf{s}}_t$. Each pixel, affordance value is set as the number of non-empty adjacent pixels (top, bottom, left, and right). This is modeled as a convolution operation with the kernel:

$$\mathbf{k} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{22}$$

using a 3×3 kernel size, padding of 1 with value 1, and stride of 1. The context-dependent affordance function g_2 transforms the delta vector $\mathbf{\Delta} = \mathbf{s}^* - \hat{\mathbf{s}}_t$ into:

$$\mathrm{g}_2(\mathbf{\Delta}) = (\mathbf{k} * (\mathbf{1} + \mathbf{\Delta})) \otimes (-\mathbf{\Delta}), \tag{23}$$

where $*$ denotes the convolution, and $\otimes$ represents the element-wise multiplication. The delta vector $\mathbf{\Delta}$ contains unremoved pixels with a value of -1 and all other pixels with a value of 0. The expression $(\mathbf{1} + \mathbf{\Delta})$ flips these values, converting -1 to 0 and 0 to 1, so the subsequent convolution counts the number of adjacent pixels with a value of 0. Finally, since only the affordance values at unsolved pixels are relevant, an element-wise multiplication between the minus delta vector $(-\mathbf{\Delta})$ and the affordance value $\mathbf{k} \cdot (\mathbf{1} + \mathbf{\Delta})$ is performed as the second component of the affordance check.

Trying to decompose a silhouette. This task was formulated as a navigation problem, where the embedding of the empty silhouette, $\mathbf{o}^* = \mathbf{0}$, represented the target node in the high-dimensional state space (cognitive map), while the embedding of the initial silhouette, $\mathbf{o}_0$, served as the starting node. In the presence of noise, GCML selected the action with the highest utility value from the set of currently feasible actions, determined by the environment's affordance factors $\mathrm{g}_1(\cdot)$ and $\mathrm{g}_2(\cdot)$. This selection process followed the directional intuition of the cognitive map:

$$\mathbf{a}_t = \mathrm{WTA}(\mathrm{g}_1(\mathbf{W}\mathrm{g}_2(\mathbf{\Delta})) + \epsilon), \tag{24}$$

where the noise term ϵ followed a Gaussian distribution $\mathcal{N}(0, 0.1)$.

After selecting an action $\mathbf{a}_t$, the next state was estimated as $\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{V}\mathbf{a}_t$, and GCML proceeded to the next iteration to attempt the removal another BB. The algorithm terminated when no feasible actions remained. If all pixels were removed, GCML successfully identified a valid decomposition; otherwise, the attempt was deemed unsuccessful.

Details of performance evaluation. The performance of GCML was evaluated based on the proportion of test cases in which it successfully solved the given silhouette in the test dataset.

As baseline comparisons, we consider a random strategy that removes as many BBs as possible from the silhouette in a random manner, as well as the CML algorithm (Stöckl et al., 2024).

Decomposition to a given non-empty goal silhouette. In the non-empty goal silhouette decomposition problem, GCML removes BBs from a given silhouette $\mathbf{o}_0$ to transform it into various non-empty target silhouettes $\mathbf{o}^* \neq \mathbf{0}$. When generating the test dataset, the initial silhouettes contained six BBs, generated following the same method described earlier. Any initial silhouette that appeared in the training dataset was removed, regardless of its position. The goal silhouettes contained two BBs, generated by randomly removing four BBs from the six BBs used to generate the initial silhouettes. GCML could plan trajectories for non-empty goal silhouettes using Equ. 24 without requiring retraining. The performance was evaluated using 1,000 test silhouette pairs.

Combining sihouette decomposition and composition. To address the silhouette rebuilding problem, GCML began with a given silhouette and iteratively modified it by adding or removing BBs to reconstruct the goal silhouette. This task was solved using the same cognitive map learned from the silhouette decomposition problem introduced earlier, leveraging mathematical symmetry.

GCML followed a two-stage approach. In the first stage, it removed as many BBs as possible from the given silhouette using a straightforward method. Here, GCML adopted a sub-goal $\mathbf{s}^\dagger = \mathbf{0}$, and the delta vector was computed as $\mathbf{\Delta} = \mathbf{s}^\dagger - \mathbf{s}_t$. In the second stage, GCML reconstructed the target silhouette by sequentially adding BBs. In this phase, the goal was the final silhouette $\mathbf{s}^* = \mathbf{Q}\mathbf{o}^*$, and the delta vector was given by $\mathbf{\Delta} = \mathbf{s}^* - \mathbf{s}_t$. Since adding BBs was the inverse operation of removing them, GCML selected actions based on:

$$\mathbf{a}'_t = \mathrm{WTA}(\mathrm{g}'_1(\mathbf{W}\mathrm{g}'_2(\mathbf{\Delta})^{\mathrm{T}}) + \epsilon), \tag{25}$$

where the affordance factor g'_1 ensured that GCML only added BBs contained within the goal $\mathbf{s}^*$. The function $\mathrm{g}'_2(\mathbf{\Delta}) = (\mathbf{k} * (\mathbf{1} - \mathbf{\Delta})) \otimes \mathbf{\Delta}$ was designed following the same principle as in Equ. 23. The negative transformation of $\mathbf{\Delta}$'s sign in Equ. 23 facilitated the addition of BBs as the inverse of their removal. Here, $\mathbf{a}'_t$ represented the action of adding a BB corresponding to $\mathbf{a}_t$, which removed a BB.

For the testing dataset, the initial and goal silhouettes were generated using the five BBs following the previously introduced generation process. To test the robustness of our model, a randomly selected noise-pixel was added to the same empty location in both the start and goal silhouettes. Any initial or goal silhouette that appears in the training dataset was removed, regardless of its position. A total of 1,000 initial and goal silhouette pairs were generated to test the performance.

Acknowledgments

YY and WM were partially supported by the National Science Foundation of the USA (EFRI BRAID project 2318152) and the Austrian Science Fund (FWF) (10.55776/COE12). GP was partially supported by funding from the European Research Council under the Grant Agreement No. 820213 (ThinkAhead), the Italian National Recovery and Resilience Plan (NRRP), M4C2, funded by the European Union – NextGenerationEU (Project IR0000011, CUP B51E22000150006, "EBRAINS-Italy"; Project PE0000013, "FAIR"; Project PE0000006, "MNESYS"), and the Ministry of University and Research, PRIN PNRR P20224FESY and PRIN 20229Z7M8N.

Author contributions

LH, YY, and WM designed the models and the tasks. LH and YY carried out numerical simulations, and analyzed the results. LH, YY, RZ, GP, and WM wrote the paper

References

Bakermans, J.J., J. Warren, J.C. Whittington, and T.E. Behrens. 2025. Constructing future behavior in the hippocampal formation through composition and replay. *Nature Neuroscience*: 1–12 .

Behrens, T.E., T.H. Muller, J.C. Whittington, S. Mark, A.B. Baram, K.L. Stachenfeld, and Z. Kurth-Nelson. 2018. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron 100*(2): 490–509 .

Berkes, P., G. Orbán, M. Lengyel, and J. Fiser. 2011. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science 331*(6013): 83–87 .

Bernardi, S., M.K. Benna, M. Rigotti, J. Munuera, S. Fusi, and C.D. Salzman. 2020. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell 183*(4): 954–967 .

Bishop, C.M. 2006. *Pattern recognition and machine learning*. Springer.

Bottini, R. and C.F. Doeller. 2020. Knowledge across reference frames: Cognitive maps and image spaces. *Trends in Cognitive Sciences 24*(8): 606–619 .

Buesing, L., J. Bill, B. Nessler, and W. Maass. 2011. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS computational biology 7*(11): e1002211 .

Bush, D., C. Barry, D. Manson, and N. Burgess. 2015. Using grid cells for navigation. *Neuron 87*(3): 507–520 .

Buzsáki, G. and E.I. Moser. 2013. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience 16*(2): 130–138 .

Campbell, M.G., S.A. Ocko, C.S. Mallory, I.I. Low, S. Ganguli, and L.M. Giocomo. 2018. Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. *Nature neuroscience 21*(8): 1096–1106 .

Crivelli-Decker, J., A. Clarke, S.A. Park, D.J. Huffman, E.D. Boorman, and C. Ranganath. 2023. Goal-oriented representations in the human hippocampus during planning and navigation. *Nature communications 14*(1): 2946 .

Davies, M., N. Srinivasa, T.H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro 38*(1): 82–99 .

Davies, M., A. Wild, G. Orchard, Y. Sandamirskaya, G.A.F. Guerra, P. Joshi, P. Plank, and S.R. Risbud. 2021. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE 109*(5): 911–934 .

Dong, L.L. and I.R. Fiete. 2024. Grid cells in cognition: mechanisms and function. *Annual Review of Neuroscience* 47 .

Eppstein, D. 1998. Finding the k shortest paths. *SIAM Journal on computing 28*(2): 652–673 .

Epstein, R.A., E.Z. Patai, J.B. Julian, and H.J. Spiers. 2017. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience 20*(11): 1504–1513 .

FeldmanHall, O., J.Y. Son, and A. Bhandari. 2025. Abstract cognitive maps for complex social systems. *Current Directions in Psychological Science*: 09637214251342742 .

Fiser, J., P. Berkes, G. Orbán, and M. Lengyel. 2010. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences 14*(3): 119–130 .

Foster, D.J. 2017. Replay comes of age. *Annual review of neuroscience 40*(1): 581–602 .

Friston, K. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience 11*(2): 127–138 .

Gardner, R.J., E. Hermansen, M. Pachitariu, Y. Burak, N.A. Baas, B.A. Dunn, M.B. Moser, and E.I. Moser. 2022. Toroidal topology of population activity in grid cells. *Nature 602*(7895): 123–128 .

Gauthier, J.L. and D.W. Tank. 2018. A dedicated population for reward coding in the hippocampus. *Neuron 99*(1): 179–193 .

Geffner, H. and B. Bonet. 2013. *A concise introduction to models and methods for automated planning.* Morgan & Claypool Publishers.

Gelly, S. and D. Silver. 2011. Monte-carlo tree search and rapid action value estimation in computer go. *Artificial Intelligence 175*(11): 1856–1875 .

George, D., R.V. Rikhye, N. Gothoskar, J.S. Guntupalli, A. Dedieu, and M. Lázaro-Gredilla. 2021. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications 12*(1): 2392 .

Gershman, S.J., J.A. Assad, S.R. Datta, S.W. Linderman, B.L. Sabatini, N. Uchida, and L. Wilbrecht. 2024. Explaining dopamine through prediction errors and beyond. *Nature neuroscience 27*(9): 1645–1655 .

Gupta, A.S., M.A. Van Der Meer, D.S. Touretzky, and A.D. Redish. 2010. Hippocampal replay is not a simple function of experience. *Neuron 65*(5): 695–705 .

Habenschuss, S., Z. Jonke, and W. Maass. 2013. Stochastic computations in cortical microcircuit models. *PLoS computational biology 9*(11): e1003311 .

Hafting, T., M. Fyhn, S. Molden, M.B. Moser, and E.I. Moser. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature 436*(7052): 801–806 .

Hills, T.T., P.M. Todd, D. Lazer, A.D. Redish, and I.D. Couzin. 2015. Exploration versus exploitation in space, mind, and society. *Trends in cognitive sciences 19*(1): 46–54 .

Hok, V., P.P. Lenck-Santini, S. Roux, E. Save, R.U. Muller, and B. Poucet. 2007. Goal-related activity in hippocampal place cells. *Journal of Neuroscience 27*(3): 472–482 .

Høydal, Ø.A., E.R. Skytøen, S.O. Andersson, M.B. Moser, and E.I. Moser. 2019. Object-vector coding in the medial entorhinal cortex. *Nature 568*(7752): 400–404 .

Iyer, A., S. Chandra, S. Sharma, and I. Fiete. 2024. Flexible mapping of abstract domains by grid cells via self-supervised extraction and projection of generalized velocity signals. *Advances in Neural Information Processing Systems* 37: 85441–85466 .

Jeong, H., A. Taylor, J.R. Floeder, M. Lohmann, S. Mihalas, B. Wu, M. Zhou, D.A. Burke, and V.M.K. Namboodiri. 2022. Mesolimbic dopamine release conveys causal associations. *Science 378*(6626): eabq6740 .

Jonke, Z., S. Habenschuss, and W. Maass. 2016. Solving constraint satisfaction problems with networks of spiking neurons. *Frontiers in neuroscience* 10: 118 .

Kahnt, T. and G. Schoenbaum. 2025. The curious case of dopaminergic prediction errors and learning associative information beyond value. *Nature Reviews Neuroscience*: 1–10 .

Kawato, M. 1999. Internal models for motor control and trajectory planning. *Current opinion in neurobiology 9*(6): 718–727 .

Kurth-Nelson, Z., T. Behrens, G. Wayne, K. Miller, L. Luettgau, R. Dolan, Y. Liu, and P. Schwartenbeck. 2023. Replay and compositional computation. *Neuron 111*(4): 454–469 .

Liu, C., R. Todorova, W. Tang, A. Oliva, and A. Fernandez-Ruiz. 2023. Associative and predictive hippocampal codes support memory-guided behaviors. *Science 382*(6668): eadi8237 .

Ma, W.J., J.M. Beck, P.E. Latham, and A. Pouget. 2006. Bayesian inference with probabilistic population codes. *Nature neuroscience 9*(11): 1432–1438 .

Maass, W. 2014. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE 102*(5): 860–880 .

Maass, W., T. Natschläger, and H. Markram. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation 14*(11): 2531–2560 .

Manns, J.R. and H. Eichenbaum. 2009. A cognitive map for object memory in the hippocampus. *Learning & memory 16*(10): 616–624 .

McNaughton, B.L., F.P. Battaglia, O. Jensen, E.I. Moser, and M.B. Moser. 2006. Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience 7*(8): 663–678 .

Milstein, A.D., Y. Li, K.C. Bittner, C. Grienberger, I. Soltesz, J.C. Magee, and S. Romani. 2021. Bidirectional synaptic plasticity rapidly modifies hippocampal representations. *Elife* 10: e73046 .

Moore, C. and J.M. Robson. 2001. Hard tiling problems with simple tiles. *Discrete & Computational Geometry* 26: 573–590 .

Moser, E.I., E. Kropff, and M.B. Moser. 2008. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci. 31*(1): 69–89 .

Muhle-Karbe, P.S., H. Sheahan, G. Pezzulo, H.J. Spiers, S. Chien, N.W. Schuck, and C. Summerfield. 2023. Goal-seeking compresses neural codes for space in the human hippocampus and orbitofrontal cortex. *Neuron 111*(23): 3885–3899 .

Newell, A., H.A. Simon, et al. 1972. *Human problem solving*, Volume 104. Prentice-hall Englewood Cliffs, NJ.

Nyberg, N., É. Duvelle, C. Barry, and H.J. Spiers. 2022. Spatial goal coding in the hippocampal formation. *Neuron 110*(3): 394–422 .

O'Keefe, J. and L. Nadel. 1978. *The hippocampus as a cognitive map*. Oxford University Press, New York.

Ólafsdóttir, H.F., D. Bush, and C. Barry. 2018. The role of hippocampal replay in memory and planning. *Current Biology 28*(1): R37–R50 .

Parr, T., G. Pezzulo, and K.J. Friston. 2022. *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press.

Pennartz, C., R. Ito, P. Verschure, F. Battaglia, and T. Robbins. 2011. The hippocampal–striatal axis in learning, prediction and goal-directed behavior. *Trends in neurosciences 34*(10): 548–559 .

Pezzulo, G., M.A. Van der Meer, C.S. Lansink, and C.M. Pennartz. 2014. Internally generated sequences in learning and executing goal-directed behavior. *Trends in cognitive sciences 18*(12): 647–657 .

Pfeiffer, B.E. and D.J. Foster. 2013. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature 497*(7447): 74–79 .

Pouget, A., P. Dayan, and R.S. Zemel. 2003. Inference and computation with population codes. *Annual review of neuroscience 26*(1): 381–410 .

Powers, W.T. 1973. *Behavior: The control of perception*. Aldine Chicago.

Raju, R.V., J.S. Guntupalli, G. Zhou, C. Wendelken, M. Lázaro-Gredilla, and D. George. 2024. Space is a latent sequence: A theory of the hippocampus. *Science Advances 10*(31): eadm8470 .

Redish, A.D. 2016. Vicarious trial and error. *Nature Reviews Neuroscience 17*(3): 147–159 .

Rivard, B., Y. Li, P.P. Lenck-Santini, B. Poucet, and R.U. Muller. 2004. Representation of objects in space by two classes of hippocampal pyramidal cells. *Journal of General Physiology 124*(1): 9–25 .

Rosenblueth, A., N. Wiener, and J. Bigelow. 1943. Behavior, purpose and teleology. *Philosophy of science 10*(1): 18–24 .

Russell, S.J. and P. Norvig. 2016. *Artificial intelligence: a modern approach*. pearson.

Schuck, N.W., M.B. Cai, R.C. Wilson, and Y. Niv. 2016. Human orbitofrontal cortex represents a cognitive map of state space. *Neuron 91*(6): 1402–1412 .

Schwartenbeck, P., A. Baram, Y. Liu, S. Mark, T. Muller, R. Dolan, M. Botvinick, Z. Kurth-Nelson, and T. Behrens. 2023. Generative replay underlies compositional inference in the hippocampal-prefrontal circuit. *Cell 186*(22): 4885–4897 .

Shin, J.D. and S.P. Jadhav. 2016. Multiple modes of hippocampal–prefrontal interactions in memory-guided behavior. *Current opinion in neurobiology* 40: 161–169 .

Sorscher, B., G.C. Mel, S.A. Ocko, L.M. Giocomo, and S. Ganguli. 2023. A unified theory for the computational and mechanistic origins of grid cells. *Neuron 111*(1): 121–137 .

Stöckl, C., Y. Yang, and W. Maass. 2024. Local prediction-learning in high-dimensional spaces enables neural networks to plan. *Nature Communications 15*(1): 2344 .

Stoianov, I., D. Maisto, and G. Pezzulo. 2022. The hippocampal formation as a hierarchical generative model supporting generative replay and continual learning. *Progress in Neurobiology* 217: 102329 .

Sutton, R.S. 1992. Adapting bias by gradient descent: An incremental version of delta-bar-delta. In *AAAI*, pp. 171–176. Citeseer.

Sutton, R.S. and A.G. Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Tang, W., J.D. Shin, and S.P. Jadhav. 2021. Multiple time-scales of decision-making in the hippocampus and prefrontal cortex. *Elife* 10: e66227 .

Tavares, R.M., A. Mendelsohn, Y. Grossman, C.H. Williams, M. Shapiro, Y. Trope, and D. Schiller. 2015. A map for social navigation in the human brain. *Neuron 87*(1): 231–243 .

Tsoar, A., R. Nathan, Y. Bartan, A. Vyssotski, G. Dell'Omo, and N. Ulanovsky. 2011. Large-scale navigational map in a mammal. *Proceedings of the National Academy of Sciences 108*(37): E718–E724 .

Ujfalussy, B.B. and G. Orbán. 2022. Sampling motion trajectories during hippocampal theta sequences. *Elife* 11: e74058 .

Ulsaker-Janke, I., T. Waaga, T. Waaga, E.I. Moser, and M.B. Moser. 2023. Grid cells in rats deprived of geometric experience during development. *Proceedings of the National Academy of Sciences 120*(41): e2310820120 .

Van de Maele, T., B. Dhoedt, T. Verbelen, and G. Pezzulo. 2024. A hierarchical active inference model of spatial alternation tasks and the hippocampal-prefrontal circuit. *Nature Communications 15*(1): 9892 .

Van Der Meer, M., Z. Kurth-Nelson, and A.D. Redish. 2012. Information processing in decision-making systems. *The Neuroscientist 18*(4): 342–359 .

Vollan, A.Z., R.J. Gardner, M.B. Moser, and E.I. Moser. 2025. Left–right-alternating theta sweeps in entorhinal–hippocampal maps of space. *Nature 639*(8056): 995–1005 .

Widloski, J. and D.J. Foster. 2022. Flexible rerouting of hippocampal replay sequences around changing barriers in the absence of global place field remapping. *Neuron 110*(9): 1547–1558 .

Wiener, N. 1961. *Cybernetics or Control and Communication in the Animal and the Machine*, Volume 25. MIT press.

Wittkuhn, L., S. Chien, S. Hall-McMaster, and N.W. Schuck. 2021. Replay in minds and machines. *Neuroscience & Biobehavioral Reviews* 129: 367–388 .

Wolpert, D.M. and M. Kawato. 1998. Multiple paired forward and inverse models for motor control. *Neural networks 11*(7-8): 1317–1329 .

Wu, Y. and W. Maass. 2025. A simple model for behavioral time scale synaptic plasticity (btsp) provides content addressable memory with binary synapses and one-shot learning. *Nature communications 16*(1): 342 .

Xiao, Z., X. Wang, J. Zhang, J. Ou, L. He, Y. Qu, X. Hu, T. Behrens, and Y. Liu. 2025. Human hippocampal ripples predict the alignment of experience to a grid-like schema. *bioRxiv*: 2025–01 .

Zutshi, I., A. Apostolelli, W. Yang, Z.S. Zheng, T. Dohi, E. Balzani, A.H. Williams, C. Savin, and G. Buzsáki. 2025. Hippocampal neuronal activity is aligned with action plans. *Nature*: 1–9 .

Supplementary Information for

Neural sampling from cognitive maps enables goal-directed imagination and planning

A Supplementary information for the implementation of CMLs and GCMLs by neural networks

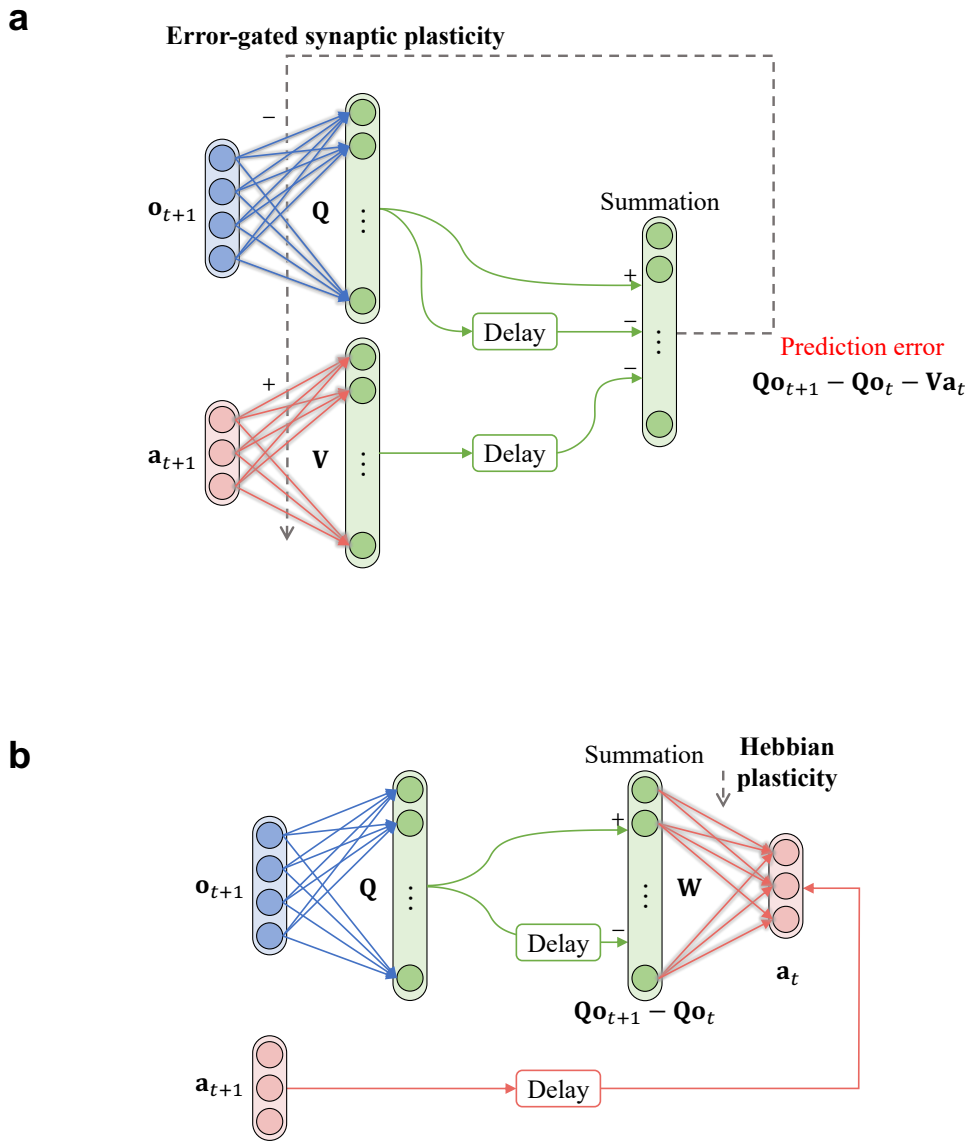

Fig. S1: Neural network implementation of CML and GCML learning (copied from Stöckl et al. (2024)). a Network structure for cognitive map learning. The prediction error, computed by the population of linear units on the right, gates synaptic plasticity between neuron populations representing observations, actions, and internal states of the cognitive map. **b** Network structure for learning the weight matrix **W** (inverse model) using Hebbian plasticity. Some signals pass through inhibitory interneurons (not shown), indicated by a negative sign at the corresponding synaptic connections. These signals are assumed to be delayed by one time step.

The architecture and learning process of the GCML is the same as for the CML. The architecture consists of several populations of neurons. Peripheral neurons encode observations (blue in Fig. S1), and other neurons encode actions, using one-hot coding (red in Fig. S1).

Delay modules introduce one-time-step delays through inhibitory relays. Such delay modules are readily available in neuromorphic hardware platforms, including Spinnaker and Intel's Loihi chip (Davies et al., 2021), as well as for in-memory computing systems.

As illustrated in Fig. S1, the model learns the parameters $\mathbf{Q}$, $\mathbf{V}$ through minimizing the prediction error $\mathbf{Q}\mathbf{o}_{t+1} - \mathbf{Q}\mathbf{o}_T - \mathbf{V}\mathbf{a}_t$ (Fig. S1(a)), while $\mathbf{W}$ is updated via Hebbian plasticity (Fig. S1(b)).

B Details of the GCML planning algorithm

Algorithm 1 provides all details of the GCML algorithm for the abstract graph task, and Algorithm 2 for the silhouette decomposition task. All notations are consistent with those used in the Methods section.

Algorithm 1 Generating Trajectories in the Abstract Graph Task

Require: Initial observation $\mathbf{o}_0$, target observation $\mathbf{o}^*$, number of trajectories n_θ, trajectory length n_{traj}, trained GCML

Compute target state: $\mathbf{s}^* \leftarrow \mathbf{Q}\mathbf{o}^*$
for $\theta = 0$ to $n_\theta - 1$ **do**
  Initialize current state: $\hat{\mathbf{s}}_t \leftarrow \mathbf{Q}\mathbf{o}_0$
  for $t = 0$ to $n_{traj} - 1$ **do**
    if $\hat{\mathbf{s}}_t \approx \mathbf{s}^*$ **then**
      break
    end if
    if $t = 0$ **then**
      Set affordance factor $\hat{\mathbf{g}}$ using real affordance from $\mathbf{o}_0$
    else
      Estimate affordance vector: $\hat{\mathbf{g}} \leftarrow \mathbf{G}\hat{\mathbf{s}}_t$
    end if
    Compute delta vector: $\mathbf{\Delta} \leftarrow \mathbf{s}^* - \hat{\mathbf{s}}_t$
    Compute utility with noise: $\mathbf{u} \leftarrow \mathbf{W}\mathbf{\Delta} + \epsilon$
    Compute affordance factor: $\hat{\mathbf{g}}_t = \mathbf{G}\hat{\mathbf{s}}_t$
    Compute eligibility: $\mathbf{e} \leftarrow \hat{\mathbf{g}}_t \otimes \mathbf{u}$
    Select next action: $\mathbf{a}_t \leftarrow \mathrm{WTA}(\mathbf{e})$
    Predict next state: $\hat{\mathbf{s}}_{t+1} \leftarrow \hat{\mathbf{s}}_t + \mathbf{V}\mathbf{a}_t$
  end for
end for
return a trajectory from the given start to the given goal.

C Supplementary information for Section 2.3

C.1 Cognitive Maps for Additional Example Silhouettes

Fig.S2 shows cognitive maps similar to Fig.4d, but for additional example silhouettes. Panels (a) and (b) present the cognitive maps of two silhouettes containing two BBs. These results demonstrate that GCML successfully maintains meaningful cognitive maps across different silhouettes, providing a foundation for generalizing to more complex tasks with more BBs.

Algorithm 2 Generating a Set of Possible Decompositions in the Silhouette Task

Require: Initial silhouette $\mathbf{o}_0$, target observation $\mathbf{o}^* = \mathbf{0}$, maximum trajectory length n_{traj}, trained cognitive map

Initialize state: $\hat{\mathbf{s}}_0 \leftarrow \mathbf{Q}\mathbf{o}_0$
for $t = 0$ to $n_{traj} - 1$ **do**
  if $\hat{\mathbf{s}}_t \approx \mathbf{s}^*$ **then**
    Found a valid decomposition for $\mathbf{o}_0$
    break
  end if
  Compute delta vector: $\mathbf{\Delta} \leftarrow \mathbf{Q}\mathbf{o}^* - \hat{\mathbf{s}}_t$
  Apply context-dependent affordance factor g_2 (see Equ. 23):
    $\mathrm{g}_2(\mathbf{\Delta}) = (\mathbf{k} * (\mathbf{1} + \mathbf{\Delta})) \otimes (-\mathbf{\Delta})$
  if no valid action remains **then**
    No valid decomposition found.
    break
  end if
  Compute eligibility with noise: $\mathbf{e}_t \leftarrow \mathrm{g}_1(\mathbf{W}\mathrm{g}_2(\mathbf{\Delta})) + \epsilon$ (see Equ. 24)
  Select action: $\mathbf{a}_t \leftarrow \mathrm{WTA}(\mathbf{e}_t)$
  Compute next state: $\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \mathbf{V}\mathbf{a}_t$
end for
return a generated decomposition plan

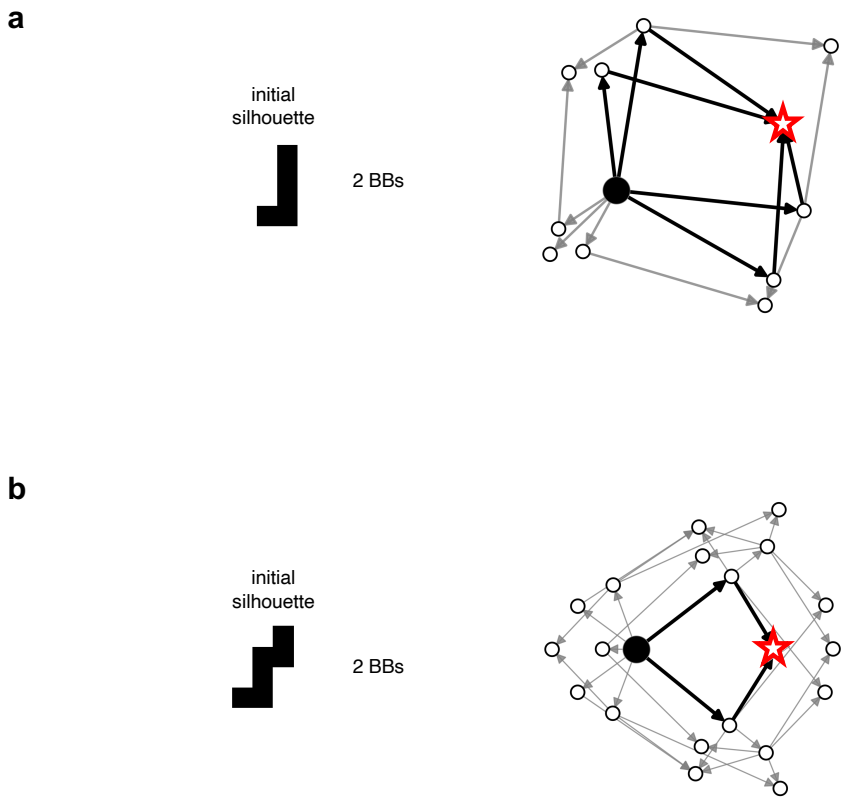

Fig. S2: **Cognitive map of GCML for the tiling problem.** Panels (a) and (b) show the cognitive map of a silhouette containing two BBs. The left panels display the initial silhouette, and the right panels present their corresponding cognitive maps. The black node represents the high-dimensional state $\mathbf{s}_0$ of the initial silhouette, and the red star denotes the high-dimensional state $\mathbf{s}^*$ of the empty goal silhouette. Each arrow corresponds to the embedding of an action (i.e., removing a BB from a silhouette). All nodes are calculated following Eq. 23. All 2-D projections are generated via t-SNE (PCA initialized). The solid black arrows indicate trajectories that successfully decompose the initial silhouettes.